DISTRIBUTED AND PLUG-AND-PLAY CONTROL FOR CONSTRAINED SYSTEMS

Matricola: 397092

Dipartimento di Ingegneria Industriale e dell'Informazione
Identification and Control of Dynamic Systems Laboratory

Università degli Studi di Pavia

A Ph.D. dissertation by

Stefano Riverso

Prof Giancarlo Ferrari Trecate, advisor

Pavia, 2010-2013

To Hope...

Money does not represent such a value as men have placed upon it.
All my money has been invested into experiments
with which I have made new discoveries
enabling mankind to have a little easier life.
— Nikola Tesla

If you knew how much work went into it,
you would not call it genius.
— Michelangelo Buonarroti

Acknowledgements

This thesis is the result of my research carried out at the Università degli Studi di Pavia.
First of all, I would like to thank Professor Giancarlo Ferrari Trecate. He has been more than a tutor and I truly believe he has been my "scientific father". He has been my guide in the field of decentralized and distributed control. Gianni taught me not only technical aspects of control theory, he taught me how to become a researcher. I will be always in debt for his incredible support.
My thanks go also to Dr. Marcello Farina. I had the opportunity to learn a lot from him. In particular Marcello taught me basic mathematical tools needed for the research described in Part II of this thesis. I also thank Professor Riccardo Scattolini for illuminating and helpful discussions.
A special thank goes to Davide that, five years ago, made me to love automatic control.
I would like to thank Professor James B. Rawlings that gave me the opportunity to visit the University of Wisconsin and to improve my skills in MPC. A special thank goes to Mrs. Mary that has been incredibly kind with me during my stay in Madison.
I thank all people of the ICDS lab: I wish them all the best. I also thank all the master students that I worked with and helped me develop crazy ideas: Giuseppe, Daria, Alberto, Michele, Simone and, in particular, Fabio that is starting now his PhD.
I would also thank Stefania and Simone for their special support during my PhD: their friendship made me feel better during my PhD and gave me the energy needed for facing great challenges.
I thank my family. I don't need to spend a lot of words: they are and they will be my landmark. Their support was more than important, it has been

fundamental. They inspired me, encouraged me and helped me follow my way. I also thank Mario, always kind with his older brother.
Finally, I thank Giusi. She is the best part of my life and I want that she will be always on my side to help me to be a better man. Thanks for everything and for that "crazy little thing called love".

Pavia, 21 November 2013

S. R.

Acronyms

AGC	Automatic Generation Control
CeMPC	Centralized Model Predictive Control
DeMPC	Decentralized Model Predictive Control
DSE	Distributed State Estimator
DiMPC	Distributed Model Predictive Control
LP	Linear Programming
LQR	Linear Quadratic Regulator
LSE	Local State Estimator
LSS	Large-Scale System
LTI	Linear Time Invariant
MPC	Model Predictive Control
mRPI	minimal Robust Positively Invariant
MRPI	Maximal Robust Positively Invariant
PnP	Plug-and-Play
PnPMPC	Plug-and-Play Model Predictive Control
PNS	Power Network System
PnP-DeMPC	Plug-and-Play Decentralized Model Predictive Control
PnP-DiMPC	Plug-and-Play Distributed Model Predictive Control
pRPI	practical Robust Positive Invariance or practical Robust Positively Invariant
RCI	Robust Control Invariance or Robust Control Invariant

RPI	Robust Positive Invariance or Robust Positively Invariant
QP	Quadratic Programming

List of Figures

List of Tables

Contents

List of Algorithms

Chapter 1

Introduction

Contents

1.1 Control architectures for large-scale systems

Modern engineering offers several examples of man-made systems characterized by a large number of states and inputs or deployed over a wide area. Although there is no rigorous definition of Large-Scale Systems (LSSs), they are often thought as the result of many subsystems interacting through the coupling of physical variables or the transmission of information over a communication network. It is therefore common to represent an LSS through a coupling graph, i.e. a directed graph where nodes are subsystems and edges represent coupling relations. In Figure 1.1, we show an example of LSS decomposed into subsystems: an arrow with label $x_{[i]}$ from subsystem i to subsystem j indicates that the dynamics of subsystem j depends on the variable $x_{[i]}$. In this case we say that subsystems i and j are arranged in a parent-child relation.

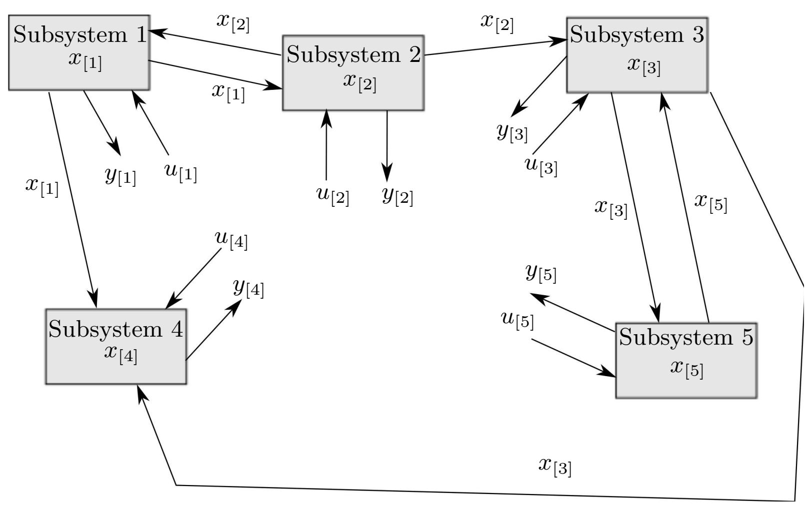


Figure 1.1: LSS decomposed into subsystems. States, inputs and outputs of each subsystem are $x_{[i]}$, $u_{[i]}$ and $y_{[i]}$, respectively. For simplicity, subsystems have no inputs, outputs or states in common. Furthermore, coupling arises through state variables only. Coupling relations are represented by continuous arrows connecting subsystems.

This kind of decomposition arises naturally in, e.g., models for building temperature regulation [MBH+12], [OPJ+12], models for traffic control [Dag97], [PK02], [BDD06], models of power network systems [Saa02], microgrids [EDI12], [BK13], [BZ13], and cyber-physical systems [SP11], [Kon13]. The decomposition of an LSS into subsystems is dictated by several criteria such as the physical locations of system components and the need of obtaining a coupling graph that is sparse and where existing couplings are "small". These issues have been thoroughly studied in the past, leading to algorithms and guidelines for obtaining graph representations of LSSs [GM86], [Lun92], [Zv94], [GR06]. Complexity of LSS models poses several challenges as the application of simulation, analysis and control design algorithms conceived for small and medium-scale systems can become prohibitive [BL88], [Lun92].
In the following, we highlight issues arising in centralized control and illus-

trate alternative control architectures.

1.1.1 Centralized control

In centralized control schemes, all control variables are computed by a single regulator. Therefore all subsystems transmit their outputs to the central controller that computes control inputs which are sent to actuators collocated with subsystems. The flow of information and the resulting star-like topology of the communication network are represented in Figure 1.2.

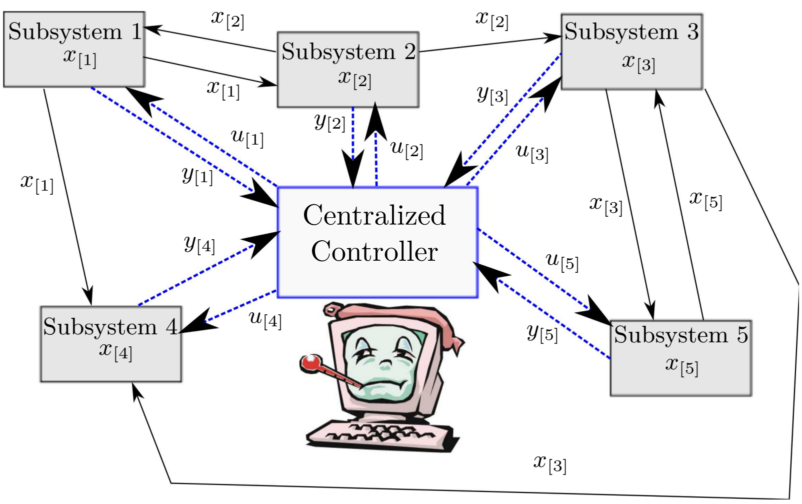


Figure 1.2: Centralized control scheme. The meaning of variables and continuos arrows is the same as in Figure 1.1. Dashed arrows indicate that the outputs of each subsystem are sent to a centralized controller that computes and transmits inputs to each subsystem.

A centralized control scheme for an LSS can suffer from the following problems.

- *Computational burden.* A centralized regulator needs a considerable amount of computing power and memory in order to compute control inputs within a sampling interval. Moreover, these issues are more and more relevant as the sampling rate increases.

- *Communication network.* Centralized control requires a star-like topology of the communication network that could impact on the cost of the control system and could also introduce delays in the transmissions when subsystems are distributed over a wide geographical area.

- *Reliability.* A failure in a single subsystem or in a link could compromise the proper functioning of the overall controlled LSS.

However, centralized control has been by far the most studied architecture in the past and several design methods exist for guaranteeing stability and performance for the closed-loop system, see e.g. [AM89], [SP96], [Kha02].

Alternatives to centralized control are offered by decentralized and distributed control schemes.

1.1.2 Decentralized control

In decentralized control each subsystem is equipped with a local controller that receives the outputs from the corresponding subsystem and computes the control inputs (see Figure 1.3).

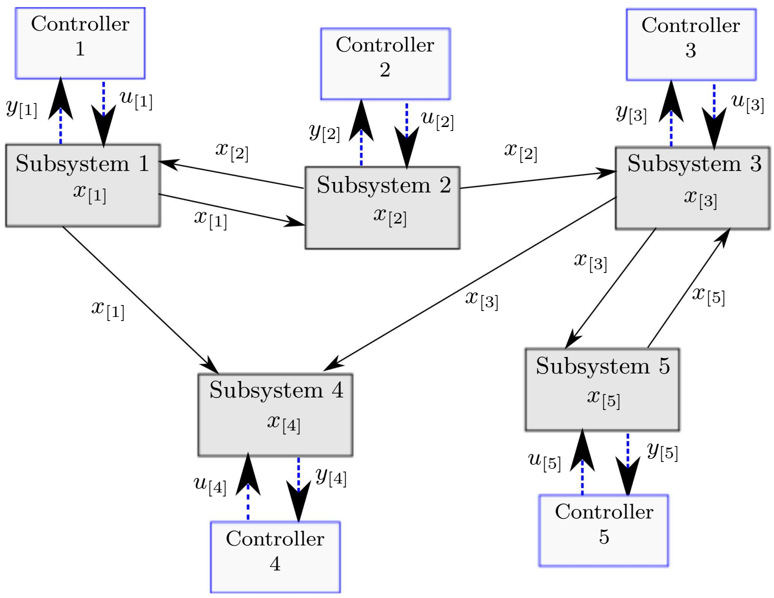


Figure 1.3: Decentralized control scheme. The meaning of variables and continuous arrows is the same as in Figure 1.1. Dashed arrows indicate that the outputs of each subsystem are sent to a local controller that computes and transmits inputs to corresponding subsystem.

As a result of decentralization, we have the following advantages with respect to centralized control.

- *Communication network.* Each subsystem is equipped with a local controller that can use local outputs only for computing the control inputs of the corresponding subsystem. In this case, the flow of information is reduced to a peer-to-peer communication between the local subsystem and its controller.
- *Computational burden.* Since each subsystem is equipped with a local controller, the control inputs for each subsystem can be computed in parallel, using local hardware. Therefore computational resources for a single controller are usually much lower than resources needed for implementing a centralized controller.

However, the critical issue of decentralized control is how to guarantee stability and suitable level of performance for the closed-loop LSS. As an example, assume that the LSS is linear, stabilizable and detectable. It is well known that these assumption guarantee the existence of a stabilizing centralized output-feedback controller. However, in a decentralized architecture, local controllers providing closed-loop stability might not exist. This issue has motivated a large stream of research on the problem of designing local controllers for achieving closed-loop stability and desired performance levels, especially when system variables are not affected by constraints, [Gv73], [SVAS78], [BL88], [Lun92], [ZIF01], [Zv10].

1.1.3 Distributed control

A compromise between centralized and decentralized control is offered by distributed architectures (see Figure 1.4).

In distributed control, as in decentralized control, each subsystem is equipped with a local controller, but controllers can transmit and receive quantities from other subsystems and controllers. Therefore, if these pieces of information are properly used, the goal of stabilizing the closed-loop LSS and guaranteeing prescribed level of performance can be easier to achieve [Lun92], [LA06], [Zv10].

1.1.4 Decentralized design of local controllers

In previous sections, we discussed different control schemes and we highlighted that issues brought about by centralized control architectures can be overcome by decentralized and distributed control. Classification of

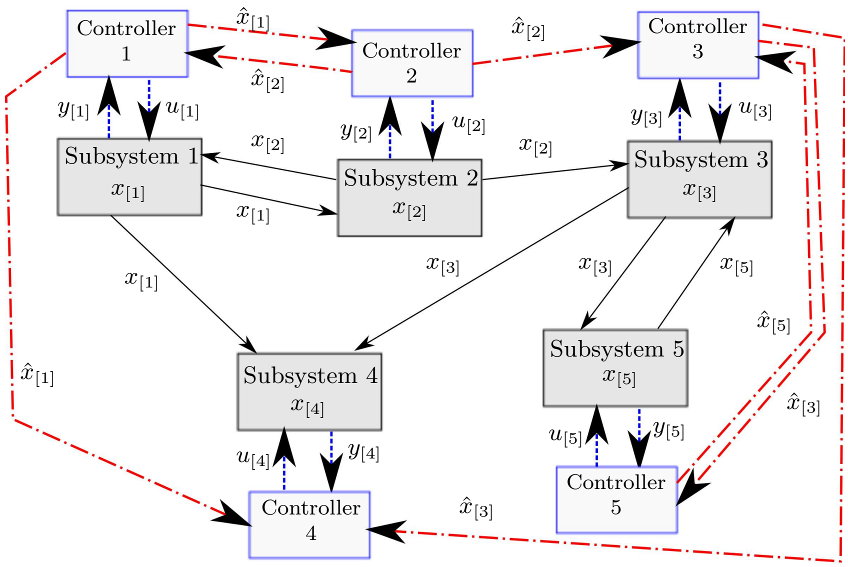


Figure 1.4: Distributed control scheme. The meaning of variables, continuous and dashed arrows is the same as in Figure 1.3. Dot-dashed arrows represent how quantities $\hat{x}_{[i]}$, $i \in 1:5$ are exchanged between subsystems.

control schemes into centralized, decentralized and distributed is based on features of online operations. From the applicative point of view, it is also important to study how difficult is the offline design of the whole controller. In this respect, synthesis algorithms can be classified into centralized and decentralized [BL88], [Lun92].

Centralized design means that the synthesis of a local controller is based on the knowledge of a model for the whole system. An example is provided by the procedure in [WD73] for the design of a stabilizing decentralized controller for linear systems. Centralized methods are not scalable, meaning that the complexity for designing a local controller increases with the size of the overall system. Furthermore, a global model of an LSS might not even be available, as in the case of nation-wide power networks.

A design procedure is decentralized if the whole model of the system is never used in any step of the synthesis process [BL88] and [Lun92]. This definition is subtle because it does not exclude that pieces of information from all subsystems are required for designing a local controller. Examples are decentralized control schemes that rely on vector Lyapunov functions for assessing the stability of the closed-loop system and require to analyze the stability of a M-th order system where M is the number of subsystems.

In this case the complexity of designing a controller for a given subsystem scales with M (see [Bai66], [Gv73], [Lun92] and recent results in [RKF10], [RKF11]).
The concept of decentralized design can be strengthened by asking the design of a local controller requires information from the corresponding subsystem only. In this case we achieve complete decentralization [Lun92]. However, these design methods can guarantee stability only if the LSS has very special properties, e.g. the coupling graph is a directed tree (see Section 6 of [BL88]).
In conclusion, decentralized design mitigates scalability problems of centralized synthesis but, in general, does not eliminate them completely. In the next section we discuss a particular case of decentralized design that is scalable but less restrictive than complete decentralization.

1.1.5 Plug-and-play design of local controllers

We consider decentralized synthesis with the additional constraint that the design of a local controller can use information at most from parents of the corresponding subsystem.
This approach has several advantages. First, the communication flow at the design stage has the same topology of the coupling graph, that is usually sparse. Second, after each parent subsystem has sent required quantities to its children, the design of local controllers can be done in parallel. Indeed, computations required for the design of a local controller can be performed using local computational resources only. Third, the complexity of synthesizing a local controller for a subsystem scales only with the number of its parents rather than the total number of subsystems. Fourth, if a subsystem joins an existing network (plug-in operation) at most subsystems that are influenced by it can retune their controllers. In other words, besides the synthesis of a local controller for the new subsystem, changes in the existing control scheme can embrace controllers of child subsystems only (see Figure 1.5). In a similar way, if a subsystem leaves the network (unplugging operation) at most its children can retune their controllers.
We refer to this kind of decentralized synthesis as Plug-and-Play (PnP) design if, in addition, when a subsystem joins/leaves an existing network of subsystems there is a procedure for automatically assessing if the operation does not spoil stability and constraint satisfaction for the overall LSS. A different definition of PnP design is given in [Sto09, BTS13].
PnP design procedures are very attractive for LSS where the number of subsystems can vary over time. As an example, in power network systems

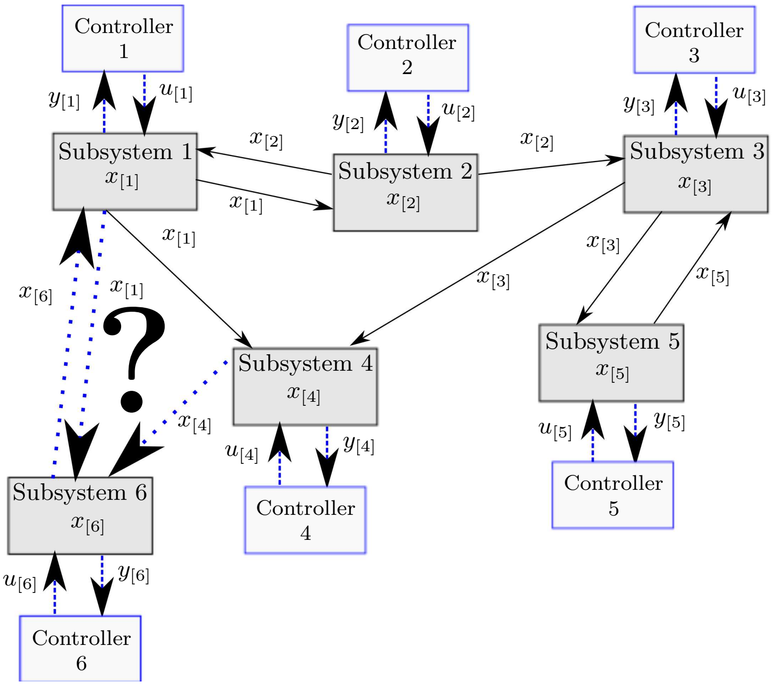


Figure 1.5: Parent-based decentralized control scheme. When subsystem 6 is added to the network, besides the design of controller 6, at most controller 1 can be retuned. Similarly, if subsystem 6 leaves the network, at most the retuning of controller 1 is allowed.

and microgrids, PnP synthesis provides a scalable procedure for the addition and removal of new generation units. Furthermore, PnP controllers can considerably facilitate the revamping of control systems. For instance, the replacement of an old actuator with a newer one corresponds to an unplugging operation followed by the plug-in of a new subsystem. In this case, PnP design allows one to automatically assess feasibility of the whole process.

We highlight that, in this thesis, plugging in and unplugging are considered as offline operations, i.e. they do not lead to switching between different dynamics in real time.

1.2 Decentralized and distributed schemes for constrained systems: model predictive control and state estimation

Nowadays, Model Predictive Control (MPC) is *de facto* the standard to the control of constrained systems. One of the reasons for the success of MPC algorithms consists in the intuitive way of addressing the control problem: using a model of the system, MPC predicts its behavior in the future and computes the control inputs accordingly. This allows one to achieve different aims, such as tracking of set-points, regulation to an equilibrium or disturbance rejection, while guaranteeing the fulfillment of constraints on states variables and control inputs. The main ingredients of a state-feedback MPC scheme for discrete-time system are:

- a model of the system;
- a performance index encoding the desired behavior of the system in the future;
- an optimization algorithm that minimizes over a time horizon the performance index and computes the control inputs based on the model of the system and constraints on system variables;
- the receding horizon strategy: at each sampling time the optimization problem is solved using the current state of the system. Then, the computed control action for the current time instant is applied while the rest of the calculated control inputs is discarded.

For extensive details, we refer the interested reader to [BM99], [ML99], [Raw00], [MRRS00], [Mac02] and [RM09].
MPC schemes have been developed under the assumptions that a model of the system is available. For large-scale systems, MPC algorithms suffer from several problems: indeed when the number of states and constraints becomes large, the optimization problem is characterized by several variables and constraints and this can prevent from solving the problem in a sampling interval. Clearly, this issue adds up to the ones described in Section 1.1.1 for centralized controllers.

There are two main approaches in order to overcome the difficulties of solving online an MPC optimization problem for LSS:

- use distributed optimization algorithms;

- design decentralized and distributed controllers.

Distributed optimization is based on the idea of calculating the control inputs on a distributed computational architecture. We defer the interested reader to [Wis71], [BT97], [NS08], [GR10], [GDK+13] and the references therein for MPC approaches based on distributed optimization.

In decentralized MPC (DeMPC) each subsystem is equipped with a local MPC that receives the state of the corresponding subsystem and computes the control inputs. We highlight that most of the proposed DeMPC schemes are based on the assumption of small coupling between subsystems [MS06], [KBB06], [RMS07], [HLJ10], [ABB11].
A compromise between centralized MPC (CeMPC) and DeMPC is offered by distributed MPC (DiMPC) where information is transmitted in real time among regulators. DiMPC schemes have recently attracted the attention of many researchers and can be classified into cooperative and non-cooperative methods [Sca09], [CSML13]. Cooperative DiMPC schemes with stability guarantees have been proposed in [RS08] and [SVR+10] for linear systems and aim at approximating CeMPC controllers. However, they require all-to-all communication between regulators and each MPC controller requires knowledge on how subsystem operations impact on the whole system. Furthermore, existing solutions account for input constraints only. Other cooperative DiMPC schemes have been proposed for nonlinear systems [SWR11], for decoupled subsystems with coupled constraints [MRA12], based on dynamic games [SGML08], [Gio11] and for tracking problems [FLAC13]. As for non-cooperative schemes, each MPC controller minimizes a performance index local to each subsystem [CJKT02], [Dun07], [TR10], [FS12]. A recent approach [CJKT02] considers linear discrete-time systems without constraints and proposes DiMPC schemes guaranteeing stability of the origin under a specific controllability assumption. Remarkably, these methods assume transmission of information from parent to child subsystems only. This communication flow is also assumed in [Dun07] where a DiMPC regulator for nonlinear continuous-time systems subject to input constraints has been proposed. A robust DiMPC scheme for subsystems coupled only through constraints has been recently studied in [TR10]. To date, the most feature-rich non-cooperative DiMPC schemes for discrete-time linear systems appeared in [FS12]. Relying on the existence of a decentralized state-feedback static and stabilizing controller for the unconstrained system, the controllers in [FS12] account for constraints on state and inputs of individual subsystems as well as coupling constraints among states of different subsystems. Moreover, communica-

tion is required only among neighboring subsystems or subsystems involved in state-coupling constraints. Stability of the closed-loop system can then be guaranteed under suitable assumptions. Another interesting approach has been proposed in [LMC09] where a DiMPC scheme based on the availability of a Lyapunov-based controller has been discussed.

DeMPC and DiMPC schemes, as well as CeMPC schemes, are often based on state feedback, i.e. they assume that state measurements are always available. However, in most applications only outputs can be measured and therefore one need observers or state estimators for reconstructing the state of the system. As an example, for centralized state estimation, usually centralized Kalman filters or moving horizon estimation can be adopted (for more details, we refer the reader to Chapter 4 in [RM09]).
As for controllers, observers can be classified according to the flow of signals needed for their functioning and the amount of information needed for their design. If state estimates are computed by a single unit, the observer is centralized while if subsystems are equipped with local observers estimating local states, the architecture is decentralized (no communication between observers) or distributed. Local observers can be synthesized using the model of the whole system (centralized design) or not (decentralized design). PnP design will then correspond to decentralized or distributed observers that (i) are synthesized with a decentralized procedure where local observers use information from parent subsystems only, (ii) there is an automatic procedure for designing a local observer and (iii) when a subsystem enters the network there is an automatic test for checking whether the design of the corresponding local observer would spoil stability and performance of the whole estimator.
Decentralized and distributed observers can be further classified according to the goal of a local observer, that can be can be either to reconstruct the state of the overall plant [AR06], [CCSZ08], [KT08] and [FFTS10a], or a subset of it [Mut98], [VD03], [KM08], [SSS09b], [SSS09a], [FFTS10b] and [FS11a]. In particular, estimators are termed partition-based if a local estimator must reconstruct only the state of the corresponding subsystem and different subsystems have non-overlapping states. Observers can also differ for the topology of the communication network connecting them, ranging from all-to-all communication [VD03] to transmission of information only from each subsystem to its children [KM08], [SSS09b], [SSS09a], [FFTS10b], [FS11a]. Furthermore, local estimators can assume unconstrained models [Mut98], [VD03], [KM08], [SSS09b], [SSS09a] or cope with constraints on system variables such as disturbances, states [FFTS10b]

or estimation errors [FS11a].
Besides centralized estimators, distributed architectures are the most common. The reason is that estimates of local states can be greatly improved using estimates of parents states. Note that the adoption of decentralized or distributed observers is also necessary for avoiding to spoil advantages brought about by decentralized and distributed controllers. Similarly, scalability of controller design is less useful if a centralized method is adopted for observer synthesis.

We highlight that neither controllers not state estimators mentioned above are based on a PnP design. Indeed most of the approaches require to design controllers and state estimators in a centralized fashion. Only some approaches, e.g. [FS12], allow for decentralized design, even if the synthesis of controllers and state estimators requires some collective quantities.

1.3 Conservativity of decentralized and distributed architectures

In general, it is more challenging to stabilize an LSS with a decentralized/distributed controller than a centralized controller. Indeed, as recalled in Section 1.1.2, even if an unconstrained LSS is stabilizable and detectable, it could be not possible to guarantee overall stability of the closed-loop LSS using decentralized or distributed controllers. More precisely, a decentralized output-feedback dynamical controller always exists only if the LSS is stabilizable and detectable and there are no unstable decentralized fixed modes, i.e. uncontrollable and unobservable modes that cannot be modified using a decentralized controller [WD73]. If the controller has a distributed structure, similar results can be found in [LA08]. Therefore, compared with the centralized case, extra assumptions are needed.
Most of the existing decentralized and distributed controllers are based on the idea of "small coupling" among subsystems, that is also required by several design methods presented in this thesis. In particular, in non-cooperative architectures, coupling between subsystems is treated as a disturbance. Furthermore, as discussed in [BL88] and [Lun92], unless the LSS has some special features, decentralized design can be performed only synthesizing local controllers that are robust with respect to the coupling with parent subsystems. Clearly, methods based on small coupling suffer from some degree of conservativity. In general, however, it is difficult to make any conclusive statement concerning the applicability of small-coupling-based controllers for the following reasons.

- *Adopted decomposition of the LSS.* The degree of coupling depends on how subsystems are defined. In several application, subsystems are not identified a priori and there are several algorithms for computing decompositions where subsystems are weakly coupled [Lun92] and [Zv94]. These methods have been successfully used in many applications. Since the disturbances that must be counteracted by local controllers arise from the coupling between subsystems, the use of these tools can greatly help reduce the conservativity of decentralized and distributed architectures.

- *Specific features of the LSS under control.* The degree of conservativity also depends on specific applications. Mechanical systems, power networks and thermal processes are often composed by subsystems that are weakly coupled *by design*. Moreover, in some cases, one can design a first control layer that aims at reducing coupling between subsystems [SP96].

1.4 Thesis overview and contributions

The first part of the thesis focuses on centralized and decentralized design of distributed controllers and observers for constrained systems.

Chapter 2 In this chapter we consider a Linear Time Invariant (LTI) LSS and propose an innovative distributed control scheme capable of guaranteeing asymptotic stability and satisfaction of constraints on system inputs and states. Our method hinges on the availability of a decentralized stabilizing regulator for the unconstrained system and provides a two-layer controller for each subsystem. Upper controllers receive planned state trajectories from parent subsystems and exploit the notion of tubes [LCRM04] for achieving robustness of stability with respect to coupling. Lower controllers generate planned trajectories using MPC independently of the other subsystems. The main advantage of our scheme is that a non-trivial region of attraction of the origin of the closed-loop LSS exists independently of the coupling strength among subsystems.
Chapter 2 is based on the following papers.

- [RFT12b] S. Riverso and G. Ferrari-Trecate, “Tube-based distributed control of linear constrained systems,” *Automatica*, vol. 48, no. 11, pp. 2860–2865, 2012.

- [RFT12c] S. Riverso and G. Ferrari-Trecate, "Tube-based distributed control of linear constrained systems," Tech. Rep., 2012, [Online]. Available: http://sisdin.unipv.it/lab/personale/pers_hp/ferrari/publication_details/RFT12.php.

Chapter 3 In this chapter we propose a Distributed State Estimator (DSE) based on the notion of practical robust positively invariance [RKF10], [RKF11]. As in [FS11a], the proposed DSE is composed by local Luenberger estimators and can guarantee boundedness of the state estimation error. Moreover we include coupling attenuation terms and allow a decentralized synthesis of the DSE. In particular, the only centralized operations are executed on an M-th order system, where M is the number of subsystems in the network. We also propose procedures in order to reduce as much as possible the number of centralized operations needed for the addition or removal of a subsystem.
Chapter 3 is based on the following published and submitted papers.

- [RRFT13a] S. Riverso, D. Rubini, and G. Ferrari-Trecate, "Distributed bounded-error state estimation based on practical robust positive invariance," in *Systems & Control Letters*, Submitted, 2013.

- [RRFT13b] S. Riverso, D. Rubini, and G. Ferrari-Trecate, "Distributed bounded-error state estimation for partitioned systems based on practical robust positive invariance," in *Proceedings of the 12th European Control Conference*, 2013, pp. 2633–2638.

- [RRFT13c] S. Riverso, D. Rubini, and G. Ferrari-Trecate, "Distributed bounded-error state estimation for partitioned systems based on practical robust positive invariance," Tech. Rep., 2013, [Online]. Available: arXiv:1311.4306.

The second part of the thesis is devoted to methods for PnP design of DeMPC and DiMPC regulators and observers.

Chapter 4 In this chapter, we introduce basic mathematical tools that will be used in the next chapters in order to prove stability of an LSS equipped with PnPMPC controllers. We show how to design distributed static and dynamical PnP controllers for unconstrained LTI LSS, guaranteeing stability of the closed-loop system.
Chapter 4 is partially based on the following paper.

- [RFT13] S. Riverso and G. Ferrari-Trecate, "Plug-and-Play distributed model predictive control with coupling attenuation," *Optimal Control Applications and Methods*, Submitted, 2013.

Chapter 5 In this chapter, we consider an LTI LSS and propose a decentralized control scheme capable of guaranteeing asymptotic stability and satisfaction of constraints on system inputs and states. Using tube-based MPC [MSR05] and mathematical tools introduced in Chapter 4, we exploit robustness to respect parent subsystems and propose a PnP design procedure. We give conditions to guarantee stability and constraint satisfaction of the closed-loop LSS and show how to automatize the design of local controllers by solving suitable nonlinear optimization problems.
Chapter 5 is based on the following publications.

- [RFFT13b] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-Play Decentralized Model Predictive Control for Linear Systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2608–2614, 2013.

- [RFFT12a] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-Play Decentralized Model Predictive Control," in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012, pp. 4193–4198.

- [RFFT12b] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-Play Decentralized Model Predictive Control," Tech. Rep., 2012, [Online]. Available: arXiv:1302.0226.

Chapter 6 In this chapter, exploiting recent results on robust positive invariance for LTI systems and using efficient procedures for tube-based MPC [RM05], we propose a decentralized PnPMPC where the design of a local controller hinges on the solution of a suitable Linear Programming (LP) problem. We highlight computational advantages brought about by our method by considering the control of a large array of masses connected by springs and dampers.
Chapter 6 is based on the following papers.

- [RFFT13c] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-Play Model Predictive Control based on robust control invariant sets," *Automatica*, Submitted, 2013.

- [RFFT13a] S. Riverso, M. Farina, and G. Ferrari-Trecate, “Design of plug-and-play model predictive control: an approach based on linear programming,” in *Proceedings of the 52nd IEEE Conference on Decision and Control*, 2013, pp. 6530-6535.

- [RFFT12c] S. Riverso, M. Farina, and G. Ferrari-Trecate, “Plug-and-Play Model Predictive Control based on robust control invariant sets,” Tech. Rep., 2012, [Online]. Available: arXiv:1210.6927.

Chapter 7 In this chapter, using procedures similar to those described in Chapters 5 and 6, we enhance PnP-DeMPC regulators in order to achieve different aims: we show how to design robust PnP-DeMPC controllers for LTI LSSs and how to design local controllers for subsystems described by dynamics with matched nonlinearities. Moreover, we propose a PnP-DiMPC design procedure that accounts for the online information flow.
Part of Chapter 7 is based on the paper [RFT13] mentioned above.

Chapter 8 In this chapter, differently from Chapter 3, we propose a PnP design of Local State Estimators (LSEs): using tools similar to those proposed in Chapters 4 and 5, we show how to guarantee nominal asymptotic stability of the estimator and boundedness on the state-estimation error.
Chapter 8 is based on the following papers.

- [RFSFT13a] S. Riverso, M. Farina, R. Scattolini, and G. Ferrari-Trecate, “Plug-and-play distributed state estimation for linear systems,” in *Proceedings of the 52nd IEEE Conference on Decision and Control*, 2013, pp. 4889-4894.

- [RFSFT13b] S. Riverso, M. Farina, R. Scattolini, and G. Ferrari-Trecate, “Plug-and-play distributed state estimation for linear systems,” Tech. Rep., 2012, [Online]. Available: arXiv:1309.2002.

Chapter 9 In this chapter we present a PnP procedure for designing output-feedback controllers for LTI LSSs. More in detail, we propose an algorithm for coupling DSEs with PnP capabilities (as in Chapter 8) with PnP-DiMPC controllers (as in Chapter 6 and 7). Moreover we show how to guarantee asymptotic stability of the origin and constraints satisfaction at all time instants for the closed-loop system.

Chapter 10 This chapter is devoted to conclusions and future research directions: we summarize the results obtained in this thesis and discuss generalizations of the proposed PnP design methods for achieving different aims, such as tracking of set-points, and developing PnP fault detection schemes. Moreover, we discuss applications where PnP schemes can play a crucial role.

Appendix A In this appendix we provide basic definitions and notations used in this thesis.

Appendix B In most of the numerical examples in Part I and II of the thesis, we apply the proposed control and state estimation schemes to Power Network Systems (PNSs). In Appendix B, we describe PNSs models that have been also proposed as a benchmark exercise [RFT12a] within the HYCON2 (Highly-complex and networked control systems) network of excellence [Hyc10].

Appendix C For easing modeling, analysis and control design of LSSs, we developed the *PnPMPC-toolbox* for MatLab [RBFT12]. In this appendix we highlight the main features of the toolbox. Note that all simulations in this thesis, as well as modeling of LSSs and PnP design of controllers and state estimators, have been developed using functions of the *PnPMPC-toolbox*.

The research leading to the results of this thesis has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 257462 HYCON2 Network of excellence.

1.5 System definition

In this section, we introduce the class of systems that will be used in the thesis. We will consider discrete-time LTI LSS given by

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{D}\mathbf{d} \tag{1.1a}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \boldsymbol{\varrho} \tag{1.1b}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^p$, $\mathbf{d} \in \mathbb{R}^r$ and $\boldsymbol{\varrho} \in \mathbb{R}^p$ are the state, the input, the output, the model disturbance and the output disturbance,

respectively, at time t and $\mathbf{x}^+$ stands for $\mathbf{x}$ at time $t+1$. Let $\mathcal{M} = 1 : M$ be the set of subsystem indexes. We assume the state is composed by M state vectors $x_{[i]} \in \mathbb{R}^{n_i}$, $i \in \mathcal{M}$ such that $\mathbf{x} = (x_{[1]}, \dots, x_{[M]})$, and $n = \sum_{i\in\mathcal{M}} n_i$. Similarly, the input, the output, the model disturbance and the output disturbance are composed by vectors $u_{[i]} \in \mathbb{R}^{m_i}$, $y_{[i]} \in \mathbb{R}^{p_i}$, $d_{[i]} \in \mathbb{R}^{r_i}$, $\varrho_{[i]} \in \mathbb{R}^{p_i}$, $i \in \mathcal{M}$ such that $\mathbf{u} = (u_{[1]}, \dots, u_{[M]})$, $m = \sum_{i\in\mathcal{M}} m_i$, $\mathbf{y} = (y_{[1]}, \dots, y_{[M]})$, $p = \sum_{i\in\mathcal{M}} p_i$, $\mathbf{d} = (d_{[1]}, \dots, d_{[M]})$, $r = \sum_{i\in\mathcal{M}} r_i$ and $\boldsymbol{\varrho} = (\varrho_{[1]}, \dots, \varrho_{[M]})$.
We assume (1.1) can be equivalently described by subsystems $\Sigma_{[i]}$, $i \in \mathcal{M}$, given by

$$\Sigma_{[i]} : \quad x_{[i]}^+ = A_{ii}x_{[i]} + B_i u_{[i]} + \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]} + D_i d_{[i]} \tag{1.2a}$$

$$y_{[i]} = C_i x_{[i]} + \varrho_{[i]} \tag{1.2b}$$

where $A_{ij} \in \mathbb{R}^{n_i\times n_j}$, $i, j \in \mathcal{M}$, $B_i \in \mathbb{R}^{n_i\times m_i}$, $D_i \in \mathbb{R}^{n_i\times r_i}$, $C_i \in \mathbb{R}^{p_i\times n_i}$ and $\mathcal{N}_i$ is the set of parents of subsystem i defined as $\mathcal{N}_i = \{j \in \mathcal{M} : A_{ij} \neq 0, i \neq j\}$. Moreover, since $y_{[i]}$ depends on the local state $x_{[i]}$ only, subsystems $\Sigma_{[i]}$ are output-decoupled and then $\mathbf{C} = \text{diag}(C_1, \dots, C_M)$. Similarly, subsystems $\Sigma_{[i]}$ are input- and disturbance- decoupled, i.e. $\mathbf{B} = \text{diag}(B_1, \dots, B_M)$ and $\mathbf{D} = \text{diag}(D_1, \dots, D_M)$. We also define $\mathbf{A_D} = \text{diag}(A_{11}, \dots, A_{MM})$ and $\mathbf{A_C} = \mathbf{A} - \mathbf{A_D}$, i.e. $\mathbf{A_D}$ collects the state transition matrices of every subsystem and $\mathbf{A_C}$ collects coupling terms between subsystems. We highlight that subsystems are coupled through state variables only. In the literature, this type of coupling is sometimes referred to as "dynamic coupling".

Assumption 1.1. *The pair (A_{ii}, B_i) is stabilizable, $\forall i \in \mathcal{M}$.*

Assumption 1.2. *The pair (A_{ii}, C_i) is detectable, $\forall i \in \mathcal{M}$.*

Equivalently, we can also describe the state dynamics (1.2a) of subsystem $\Sigma_{[i]}$ as

$$\Sigma_{[i]} : \quad x_{[i]}^+ = A_{ii}x_{[i]} + B_i u_{[i]} + w_{[i]} + D_i d_{[i]} \tag{1.3a}$$

$$w_{[i]} = \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]}. \tag{1.3b}$$

We equip each subsystem with the following constraints:

- state constraints, $x_{[i]} \in \mathbb{X}_i \subseteq \mathbb{R}^{n_i}$;
- input constraints, $u_{[i]} \in \mathbb{U}_i \subseteq \mathbb{R}^{m_i}$;

- output constraints, $y_{[i]} \in \mathbb{Y}_i \subseteq \mathbb{R}^{p_i}$;
- model disturbance constraints, $d_{[i]} \in \mathbb{D}_i \subseteq \mathbb{R}^{r_i}$;
- output disturbance constraints, $\varrho_{[i]} \in \mathbb{O}_i \subseteq \mathbb{R}^{p_i}$.

The collective constraints for system (1.1) are given by

- state constraints, $\mathbf{x} \in \mathbb{X} = \prod_{i\in\mathcal{M}} \mathbb{X}_i$;
- input constraints, $\mathbf{u} \in \mathbb{U} = \prod_{i\in\mathcal{M}} \mathbb{U}_i$;
- output constraints, $\mathbf{y} \in \mathbb{Y} = \prod_{i\in\mathcal{M}} \mathbb{Y}_i$;
- model disturbance constraints, $\mathbf{d} \in \mathbb{D} = \prod_{i\in\mathcal{M}} \mathbb{D}_i$;
- output disturbance constraints, $\boldsymbol{\varrho} \in \mathbb{O} = \prod_{i\in\mathcal{M}} \mathbb{O}_i$.

Note that constraints coupling variables of different subsystems are absent and they will be introduced only when needed.

For classic definition of stability and asymptotic stability of (1.1) when $\mathbf{d} = \mathbf{0}_r$, $\forall i \in \mathcal{M}$, we defer the reader to Appendix B of [RM09]. Next, we recall the definition of robust asymptotic stability.

Definition 1.1. A set $\mathbb{Z} \subset \mathbb{X}$ is robustly attractive for system $x^+ = f(x, w)$, $w \in \mathbb{W}$, $x \in \mathbb{X}$ if $\exists\ \mathbb{A} \supset \mathbb{Z}$, $\mathbb{A} \subset \mathbb{X}$ such that, for all admissible disturbances $w(t) \in \mathbb{W}$

$$x(0) \in \mathbb{A} \Rightarrow \lim_{t\to\infty} \operatorname{dist}(x(t), \mathbb{Z}) = 0.$$

The set $\mathbb{A}$ above is termed region of attraction of $\mathbb{Z}$.

Part I

Centralized and decentralized design of distributed controllers and observers

Chapter 2

Tube-based distributed MPC

Contents

2.1 Introduction

In this chapter, we propose DiMPC architecture for the linear systems in Section 1.5, possibly equipped with coupling state constraints. We first propose a standard procedure for turning a state-feedback stabilizing decentralized controller (that can be computed using the procedures reviewed in [Lun92] and [Zv10]) into a stabilizing distributed controller. Our method is based on the notion of tubes proposed in [LCRM04] for developing computationally affordable, robust MPC schemes and used in [TR10] and [FS12] for designing DiMPC regulators. Here we extend the approach of [FS12] in order to propose local controllers that have a hierarchical structure. The upper controller $\mathcal{UC}_{[i]}$ (see Figure 2.1) for subsystem i exploits transmitted information from controllers of parent subsystems and it is coupled with a lower level controller, independent of parent subsystems, that allows input and state constraints of the whole system to be fulfilled.

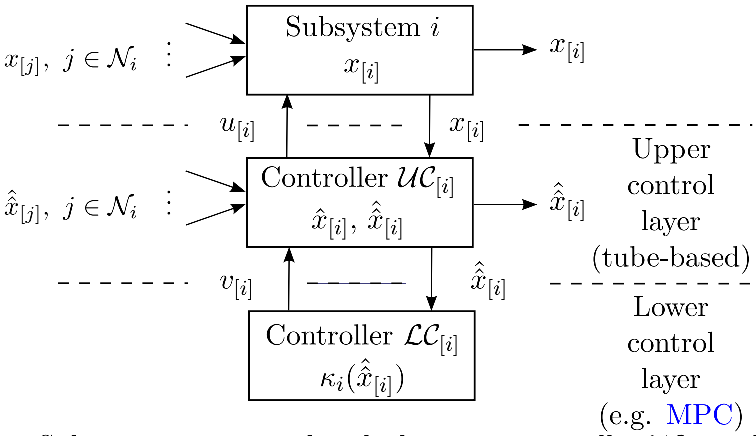


Figure 2.1: Subsystem i equipped with the upper controller $\mathcal{UC}_{[i]}$ and lower controller $\mathcal{LC}_{[i]}$.

While we adopt the same settings of [FS12] the main advantages of our method are:

1. simpler initialization of the controller;
2. reduced amount of transmitted variables between subsystems in each sampling interval;
3. existence of a non-trivial region of attraction of the origin independently of the coupling strength among subsystems;
4. the possibility of using standard explicit MPC techniques [Bor03] to compute local MPC regulators.

There are also several differences between our DiMPC scheme and the one proposed in [TR10]. The most important one is that our method applies to physically coupled subsystems.

The chapter is structured as follows. The design of upper and lower controllers is introduced in Section 2.2 with a focus on the properties guaranteeing asymptotic stability of the origin and constraint satisfaction. In

Section 2.3 we discuss how to compute all quantities local controllers depend upon. In Section 2.4 the distributed control scheme is applied to an example system and Section 2.5 is devoted to concluding remarks.

2.2 Distributed control of linear systems based on a decentralized architecture

We consider a large-scale discrete-time LTI system

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2.1}$$

composed of M subsystems, in accordance with the notation introduced in Section 1.5. In this chapter we assume that each subsystem is equipped with state and input constraints. More in detail, we equip subsystems $i \in \mathcal{M}$ with the constraints $x_{[i]} \in \mathbb{X}_i$, $u_{[i]} \in \mathbb{U}_i$ where $\mathbb{X}_i$ and $\mathbb{U}_i$ are *PC*-sets, and define the sets $\mathcal{X} = \prod_{i\in\mathcal{M}} \mathbb{X}_i$, $\mathbb{U} = \prod_{i\in\mathcal{M}} \mathbb{U}_i$. We also allow for collective state constraints given by $\mathbb{C}_x = \{\mathbf{x} \in \mathbb{R}^n : H(\mathbf{x}) \leq 0\}$ for a suitable function $H(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^c$ and assume the origin is in the interior of $\mathbb{C}_x$. Then, setting $\mathbb{X} = \mathcal{X} \bigcap \mathbb{C}_x$, we consider the collective constrained system (2.1) with[1]

$$\mathbf{x} \in \mathbb{X}, \ \mathbf{u} \in \mathbb{U}. \tag{2.2}$$

2.2.1 The upper control layer

In this section we discuss how it is possible to build a stabilizing distributed controller for system (2.1) based on the availability of a state-feedback, non-dynamical, stabilizing and decentralized controller, i.e. a matrix $\mathbf{K} = \text{diag}(K_1, \ldots, K_M)$, $K_i \in \mathbb{R}^{m_i \times n_i}$, $i \in \mathcal{M}$ such that $\mathbf{A} + \mathbf{B}\mathbf{K}$ is Schur. To this purpose we first clarify the exchange of information among subsystems. At time t each subsystem receives a *planned state* $\hat{\hat{x}}_{[i]} \in \mathbb{R}^{n_i}$ from its parent subsystems. Then, the controller associated to subsystem i uses the measured states $x_{[i]}$ and $\hat{\hat{x}}_{[j]}$, $j \in \mathcal{N}_i$ for computing the control input $u_{[i]}$. The dynamics of subsystem i can be written as

$$x_{[i]}^+ = A_{ii} x_{[i]} + B_i u_{[i]} + \sum_{j\in\mathcal{N}_i} A_{ij} \hat{\hat{x}}_{[j]} + w_{[i]} \tag{2.3}$$

$$w_{[i]} = \sum_{j\in\mathcal{N}_i} A_{ij} (x_{[j]} - \hat{\hat{x}}_{[j]}). \tag{2.4}$$

[1] With a little abuse of notation we overload the definition of $\mathbb{X}$ given in Section 1.5.

As in [FS12], in the spirit of tube-based control, we treat $w_{[i]}$ as a disturbance and define the nominal model

$$\hat{x}^+_{[i]} = A_{ii}\hat{x}_{[i]} + B_i\hat{u}_{[i]} + \hat{w}_{[i]} \tag{2.5}$$

$$\hat{w}_{[i]} = \sum_{j\in\mathcal{N}_i} A_{ij}\hat{\hat{x}}_{[j]} \tag{2.6}$$

where $\hat{u}_{[i]} \in \mathbb{R}^{m_i}$ is the input and $\hat{x}_{[i]} \in \mathbb{R}^{n_i}$ is the state. Furthermore, we assume

$$u_{[i]} = \hat{u}_{[i]} + K_i(x_{[i]} - \hat{x}_{[i]}). \tag{2.7}$$

Note that the planned states $\hat{\hat{x}}_{[j]}$ act as coupling terms in (2.5). Then, differently from [FS12], we exploit once more tube-based control and treat $\hat{w}_{[i]}$ in (2.5) as a disturbance hence defining the system

$$\hat{\hat{x}}^+_{[i]} = A_{ii}\hat{\hat{x}}_{[i]} + B_i v_{[i]} \tag{2.8}$$

where $v_{[i]} \in \mathbb{R}^{m_i}$ is the input and $\hat{\hat{x}}_{[i]} \in \mathbb{R}^{n_i}$ is the state.
Moreover, for $\hat{K}_i \in \mathbb{R}^{m_i\times n_i}$ we set in (2.5)

$$\hat{u}_{[i]} = v_{[i]} + \hat{K}_i(\hat{x}_{[i]} - \hat{\hat{x}}_{[i]}) \tag{2.9}$$

From (2.5)-(2.9), we obtain the upper controller $\mathcal{UC}_{[i]}$

$$\mathcal{UC}_{[i]} : \begin{cases} \hat{x}^+_{[i]} = (A_{ii} + B_i\hat{K}_i)\hat{x}_{[i]} + \sum_{j\in\mathcal{N}_i} A_{ij}\hat{\hat{x}}_{[j]} - B_i\hat{K}_i\hat{\hat{x}}_{[i]} + B_i v_{[i]} \\ \hat{\hat{x}}^+_{[i]} = A_{ii}\hat{\hat{x}}_{[i]} + B_i v_{[i]} \\ u_{[i]} = v_{[i]} + \hat{K}_i(\hat{x}_{[i]} - \hat{\hat{x}}_{[i]}) + K_i(x_{[i]} - \hat{x}_{[i]}) \end{cases} \tag{2.10}$$

that it is driven by the exogenous input $v_{[i]}$. Note that the only pieces of information from parent subsystems used by $\mathcal{UC}_{[i]}$ are the planned states $\hat{\hat{x}}_{[j]}$ (see Figure 2.1). This reveals the distributed nature of controllers $\mathcal{UC}_{[i]}$. We also highlight that (2.8) defines the dynamics of the planned states.
Next, we clarify properties of matrices K_i and $\hat{K}_i$, $i \in \mathcal{M}$, that are required for the stability of system (2.3)-(2.9). Defining the errors

$$z_{[i]} = x_{[i]} - \hat{x}_{[i]} \tag{2.11}$$

$$\hat{z}_{[i]} = \hat{x}_{[i]} - \hat{\hat{x}}_{[i]} \tag{2.12}$$

from (2.3)-(2.9) one obtains

$$z_{[i]}^+ = (A_{ii} + B_i K_i) z_{[i]} + \sum_{j \in \mathcal{N}_i} A_{ij} z_{[j]} + \sum_{j \in \mathcal{N}_i} A_{ij} \hat{z}_{[j]} \tag{2.13}$$

$$\hat{z}_{[i]}^+ = (A_{ii} + B_i \hat{K}_i) \hat{z}_{[i]} + \hat{w}_{[i]}. \tag{2.14}$$

Using the collective errors $\mathbf{z} = (z_{[1]}, \ldots, z_{[M]}) \in \mathbb{R}^n$ and $\hat{\mathbf{z}} = (\hat{z}_{[1]}, \ldots, \hat{z}_{[M]}) \in \mathbb{R}^n$, from (2.6), (2.8), (2.13) and (2.14) one has

$$\begin{bmatrix} \mathbf{z} \\ \hat{\mathbf{z}} \\ \hat{\hat{\mathbf{x}}} \end{bmatrix}^+ = \begin{bmatrix} \mathbf{A} + \mathbf{BK} & \mathbf{A_C} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_D} + \mathbf{B}\hat{\mathbf{K}} & \mathbf{A_C} \\ \mathbf{0} & \mathbf{0} & \mathbf{A_D} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \hat{\mathbf{z}} \\ \hat{\hat{\mathbf{x}}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{B} \end{bmatrix} \mathbf{v}. \tag{2.15}$$

This system has a cascade structure in the sense that $\mathbf{v}$ acts only on $\hat{\hat{\mathbf{x}}}$, $\hat{\hat{\mathbf{x}}}$ acts only on $\hat{\mathbf{z}}$ and $\hat{\mathbf{z}}$ acts only on $\mathbf{z}$. Noting that $\mathbf{A_D} + \mathbf{B}\hat{\mathbf{K}} = \text{diag}(A_{11} + B_1 \hat{K}_1, \ldots, A_{MM} + B_M \hat{K}_M)$, the following assumption must be fulfilled for stability.

Assumption 2.1. *The matrices* $\mathbf{A} + \mathbf{BK}$ *and* $A_{ii} + B_i \hat{K}_i$, $i \in \mathcal{M}$ *are Schur.*

It is easy to show that if subsystems are decoupled, i.e. $A_{ij} = \mathbf{0}_{n_i \times n_j}$, $i \neq j$, under Assumption 2.1, the system given by (2.3) and (2.7) is a follower of system (2.5) and the system given by (2.5) and (2.9) is a follower of system (2.8), in the sense that $\mathbf{x}(t) - \hat{\mathbf{x}}(t) \to \mathbf{0}_n$ and $\hat{\mathbf{x}}(t) - \hat{\hat{\mathbf{x}}}(t) \to \mathbf{0}_n$ as $t \to +\infty$. Therefore, the planned states $\hat{\hat{x}}_{[i]}$ can be interpreted as references that states $x_{[i]}$ try to follow.

2.2.2 The lower control layer

In this section we show how controllers $\mathcal{UC}_{[i]}$ enable the design of a lower control layer capable to stabilize system (2.1) while fulfilling state and input constraints.

Our next goal is to design state-feedback non-dynamical lower controllers $v_{[i]} = \kappa_i(\hat{\hat{x}}_{[i]})$ such that the origin of the closed-loop system (2.3)-(2.9) is asymptotically stable and constraints (2.2) are fulfilled at all time instants. For constraint satisfaction, as in tube-based control, we will compute tightened constraints $\hat{\hat{\mathbb{X}}} \subseteq \mathbb{X}$ and $\mathbb{V} \subseteq \mathbb{U}$ such that,

$$\begin{aligned} &\hat{\hat{x}}_{[i]}(k) \in \hat{\hat{\mathbb{X}}}_i, \;\; v_{[i]}(k) \in \mathbb{V}_i, \;\; \forall i \in \mathcal{M}, \;\; \forall k \in 0 : t \\ &\quad \Rightarrow \mathbf{x}(k) \in \mathbb{X}, \;\; \mathbf{u}(k) \in \mathbb{U}, \;\; \forall k \in 0 : t \end{aligned}$$

Then we will require $\hat{\hat{x}}_{[i]}(0) \in \hat{\hat{\mathbb{X}}}_i$ and that lower controllers fulfill the following assumption.

Assumption 2.2. *Lower controllers*

$$\mathcal{LC}_{[i]} : v_{[i]} = \kappa_i(\hat{\hat{x}}_{[i]}) \tag{2.16}$$

guarantee $\hat{\hat{x}}^+_{[i]} \in \hat{\hat{\mathbb{X}}}_i,\ v_{[i]} \in \mathbb{V}_i, \forall i \in \mathcal{M}$.

We start characterizing constraints on $\mathbf{x}$ and $\mathbf{u}$ induced by arbitrary constraints $\hat{\hat{\mathbb{X}}}_i$ and $\mathbb{V}_i$ on $\hat{\hat{x}}_{[i]}$ and $v_{[i]}$, respectively.

Assumption 2.3. *Sets* $\hat{\hat{\mathbb{X}}}_i$ *and* $\mathbb{V}_i$, $i \in \mathcal{M}$, *are PC-sets.*

From the results in [KG98], under Assumptions 2.1 and 2.3 there exist nonempty Robust Positively Invariant (RPI) sets $\hat{\mathbb{Z}}_i \subseteq \mathbb{R}^{n_i}$, $i \in \mathcal{M}$ for (2.14) and $\hat{w}_i \in \hat{\mathbb{W}}_i = \bigoplus_{j\in\mathcal{N}_i} A_{ij}\hat{\hat{\mathbb{X}}}_j$. Moreover, there is a nonempty RPI $\mathbb{Z} \subseteq \mathbb{R}^n$ for

$$\mathbf{z}^+ = (\mathbf{A} + \mathbf{BK})\mathbf{z} + \mathbf{w} \tag{2.17}$$

and $\mathbf{w} \in \mathbb{W} = \mathbf{A_C}\hat{\mathbb{Z}}$, $\hat{\mathbb{Z}} = \prod_{i\in\mathcal{M}} \hat{\mathbb{Z}}_i$. We also define the following sets

$$\bar{\mathbb{X}} = \hat{\mathbb{X}} \oplus \mathbb{Z},\ \ \hat{\mathbb{X}} = \prod_{i\in\mathcal{M}} \hat{\mathbb{X}}_i,\ \ \hat{\mathbb{X}}_i = \hat{\hat{\mathbb{X}}}_i \oplus \hat{\mathbb{Z}}_i \tag{2.18}$$

$$\bar{\mathbb{U}} = \mathbb{V} \oplus \hat{\mathbf{K}}\hat{\mathbb{Z}} \oplus \mathbf{K}\mathbb{Z},\ \ \mathbb{V} = \prod_{i\in\mathcal{M}} \mathbb{V}_i \tag{2.19}$$

and the collective vectors $\hat{\mathbf{x}} = (\hat{x}_{[1]}, \dots, \hat{x}_{[M]})$. The next theorem establishes key properties of lower controllers.

Theorem 2.1. *Let Assumptions 2.1, 2.2 and 2.3 hold and assume that for all* $i \in \mathcal{M}$ *controllers* $\mathcal{LC}_{[i]}$ *make the origin of (2.8) asymptotically stable with region of attraction* $\hat{\hat{\mathbb{X}}}^a_i \subseteq \hat{\hat{\mathbb{X}}}_i$. *Then,*

(a) *the origin of the closed-loop system (2.3)-(2.9) is asymptotically stable;*

(b) *if the following conditions simultaneously hold*

$$\hat{\hat{x}}_{[i]}(0) \in \hat{\hat{\mathbb{X}}}^a_i,\ \ i \in \mathcal{M} \tag{2.20}$$

$$\mathbf{x}(0) - \hat{\mathbf{x}}(0) \in \mathbb{Z} \tag{2.21}$$

$$\hat{x}_{[i]}(0) - \hat{\hat{x}}_{[i]}(0) \in \hat{\mathbb{Z}}_i,\ \ i \in \mathcal{M} \tag{2.22}$$

then $(\mathbf{x}(t), \hat{\mathbf{x}}(t), \hat{\hat{\mathbf{x}}}(t)) \to \mathbf{0}_{3n}$ *as* $t \to \infty$ *and constraints*

$$\mathbf{x}(t) \in \bar{\mathbb{X}}, \ \mathbf{u}(t) \in \bar{\mathbb{U}} \tag{2.23}$$

are fulfilled $\forall t \geq 0$.

Proof. The proof is given in Appendix 2.6.1. □

In view of Theorem 2.1, for guaranteeing (2.2) we still have to solve the problem of computing sets $\hat{\hat{\mathbb{X}}}_i$ and $\mathbb{V}_i$, $i \in \mathcal{M}$ verifying

$$\bar{\mathbb{X}} \subseteq \mathbb{X}, \ \bar{\mathbb{U}} \subseteq \mathbb{U} \tag{2.24}$$

Note that choosing $\hat{\hat{\mathbb{X}}}_i = \{\mathbf{0}_{n_i}\}$ and $\mathbb{V}_i = \{\mathbf{0}_{m_i}\}$, $i \in \mathcal{M}$ one has $\bar{\mathbb{X}} = \{\mathbf{0}_n\}$ and $\bar{\mathbb{U}} = \{\mathbf{0}_m\}$ and hence (2.24) holds. However, in this case (2.21) and (2.22) imply $\mathbf{x}(0) = \hat{\mathbf{x}}(0) = \hat{\hat{\mathbf{x}}}(0) = \mathbf{0}_n$. It is therefore of interest to study when (2.24) can be fulfilled using sets $\hat{\hat{\mathbb{X}}}_i$ and $\mathbb{V}_i$, $i \in \mathcal{M}$ that contain the origin in their non-empty interior. The following result shows that such sets $\hat{\hat{\mathbb{X}}}_i$ and $\mathbb{V}_i$ can be always found, irrespectively of the coupling strength among subsystems. Intuitively, this is possible because planned states, that are governed by lower controllers, modify the disturbances $\hat{w}_{[i]}$ and $w_{[i]}$ defined in (2.4) and (2.6), respectively.

Proposition 2.1. *If Assumption 2.1 holds, for given sets* $\mathbb{X}$ *and* $\mathbb{U}$ *containing the origin in their interior, there are sets* $\hat{\hat{\mathbb{X}}}_i$ *and* $\mathbb{V}_i$, $i \in \mathcal{M}$ *verifying Assumption 2.3 and such that (2.24) holds.*

Proof. The proof is given in Appendix 2.6.2. □

2.3 Practical design of the DiMPC scheme

In this section we discuss the following issues in the design of controllers $\mathcal{UC}_{[i]}$ and $\mathcal{LC}_{[i]}$: how to compute sets $\hat{\hat{\mathbb{X}}}_i$ and $\mathbb{V}_i$, $i \in \mathcal{M}$ for guaranteeing (2.24), how to design lower controllers (2.16) verifying the assumptions of Theorem 2.1 and how to compute $\hat{\mathbf{x}}(0)$ and $\hat{\hat{\mathbf{x}}}(0)$ verifying (2.21) and (2.22) for a given $\mathbf{x}(0)$.

For the computation of sets $\hat{\hat{\mathbb{X}}}_i$ and $\mathbb{V}_i$, $i \in \mathcal{M}$, we propose the procedure described in Algorithm 2.1 that has to be executed only once and offline. Finite termination of Algorithm 2.1 can be proved using arguments similar to the ones adopted in the proof of Proposition 2.1. Note that if the algorithm stops then, from the definition of $\bar{\mathbb{X}}$ and $\bar{\mathbb{U}}$ in (2.18) and (2.19) one has

Algorithm 2.1 Computation of tightened constraints

Input: sets $\hat{\hat{\mathbb{X}}}_i$, $\mathbb{V}_i$, $i \in \mathcal{M}$ verifying Assumption 2.3, sets $\mathbb{X}$, $\mathbb{U}$ and $\alpha < 1$.
Output: updated sets $\hat{\hat{\mathbb{X}}}_i$, $\mathbb{V}_i$ and sets $\hat{\mathbb{Z}}_i$, $\mathbb{Z}$.

(I) Each subsystem $i \in \mathcal{M}$ computes $\hat{\mathbb{W}}_i = \bigoplus_{j \in \mathcal{N}_i} A_{ij} \hat{\hat{\mathbb{X}}}_j$ and $\hat{\mathbb{Z}}_i$ that is the mRPI for (2.14)

(II) Compute the following sets in a centralized fashion

  (i) $\mathbb{W} = \mathbf{A_C} \hat{\mathbb{Z}}$, $\hat{\mathbb{Z}} = \prod_{i \in \mathcal{M}} \hat{\mathbb{Z}}_i$

  (ii) $\tilde{\mathbb{Z}}$, the MRPI set for (2.17) such that $\tilde{\mathbb{Z}} \subseteq \mathbb{X} \ominus \hat{\hat{\mathbb{X}}} \ominus \hat{\mathbb{Z}}$ and $\mathbb{V} \oplus \hat{\mathbf{K}} \hat{\mathbb{Z}} \oplus \mathbf{K} \tilde{\mathbb{Z}} \subseteq \mathbb{U}$

(III) **If** $\tilde{\mathbb{Z}} \neq \emptyset$ then set $\mathbb{Z} = \tilde{\mathbb{Z}}$ and stop;
otherwise set

$$\hat{\hat{\mathbb{X}}}_i \leftarrow \alpha \hat{\hat{\mathbb{X}}}_i, \; \mathbb{V}_i \leftarrow \alpha \mathbb{V}_i$$

and **go to** (I)

---

that (2.24) is verified because of the inclusions in Step (IIii). Outer approximations of minimal RPI (mRPI) sets in the Step (I) and (II) can be computed with a given precision using the methods developed in [RKKM05]. Algorithms for computing $\tilde{\mathbb{Z}}$ as a Maximal RPI (MRPI) in Step (IIii) have been proposed in [KGB04] when all sets are polytopes.

Algorithm 2.1 suffers from two main limitations. First, the shape of sets $\hat{\hat{\mathbb{X}}}_i$ and $\mathbb{V}_i$ provided as inputs might impact on the size of the output sets that have the same shape. Second, the computational bottleneck is the step (II) that involves the collective dynamics (2.17). Step (II) implies that our design method is centralized: possible improvements towards decentralization are discussed in Section 2.5.

As for the synthesis of lower controllers (2.16), all assumptions concerning lower controllers can be fulfilled if $\kappa_i(\hat{x}_{[i]}(t))$ is the result of an MPC regulator, hereafter termed MPC-i, for system (2.8). For a review of MPC schemes with the desired properties we defer the reader to Section 2.5.3.1 of [RM09].

Remark 2.1. The main source of conservatism of our DiMPC method is that sets $\bar{\mathbb{X}}$ and $\bar{\mathbb{U}}$, computed from the results of Algorithm 2.1, could be much smaller than $\mathbb{X}$ and $\mathbb{U}$, respectively. This is partially due to the fact that (i) tube-based control is a robust control technique that we used twice for counteracting the effect of coupling terms $\mathbf{w}$ and $\hat{\mathbf{w}}$; (ii) MPC-i regulators do not use information from parent subsystems and the fulfillment of collective state constraints is achieved by shrinking the sets $\hat{\hat{\mathbb{X}}}_i$ and $\mathbb{V}_i$ (and hence $\bar{\mathbb{X}}$ and $\bar{\mathbb{U}}$) in Algorithm 2.1. However, general statements are hard to make because $\bar{\mathbb{X}}$ and $\bar{\mathbb{U}}$ also depend upon the shape of $\mathbb{X}$ and $\mathbb{U}$ and the coupling terms A_{ij}, $i \neq j$. This suggests that a precise assessment of the degree of conservatism should be conducted on a case by case basis.

Next we address the problem of choosing the initial states $\hat{\mathbf{x}}(0)$ and $\hat{\hat{\mathbf{x}}}(0)$, given $\mathbf{x}(0)$ so as to verify (2.20), (2.21) and (2.22). Many MPC-i schemes guarantee that $\hat{\hat{\mathbb{X}}}_i^a$ is the set of states in $\hat{\hat{\mathbb{X}}}_i$ for which the MPC-i optimization problem is feasible [RM09] and hence (2.20) can be replaced by

$$\text{MPC-}i \text{ problems are feasible for } \hat{\hat{x}}_{[i]}(0), \forall i \in \mathcal{M}. \tag{2.25}$$

Conditions (2.21), (2.22) and (2.25) give rise to a bilevel programming problem [VC94] in the unknowns $\hat{\mathbf{x}}(0)$ and $\hat{\hat{\mathbf{x}}}(0)$ that is usually hard to solve. However, when $\hat{\hat{\mathbb{X}}}_i$ and $\mathbb{V}_i$ are polytopes, under suitable assumptions MPC schemes for linear constrained systems produce polytopic regions of attraction $\hat{\hat{\mathbb{X}}}_i^a$ that can be computed offline in closed form by means of explicit MPC algorithms (see [Bor03] and [KGB04]). Furthermore, methods in [RKKM05] and [KGB04] allow Algorithm 2.1 to produce sets $\hat{\mathbb{Z}}_i$, $i \in \mathcal{M}$ and $\mathbb{Z}$ that are polytopes. Therefore, if explicit MPC-i regulators are used, (2.21), (2.22) and (2.25) amount to a feasibility problem that can be solved through LP. Another method for computing $\hat{\mathbf{x}}(0)$ and $\hat{\hat{\mathbf{x}}}(0)$ hinges on the observation that a stabilizing MPC-i controller usually relies on the use of a terminal constraint set $\hat{\hat{\mathbb{X}}}_{f,i} \subseteq \hat{\hat{\mathbb{X}}}_i$ that is know in closed form and verifies $\hat{\hat{\mathbb{X}}}_{f,i} \subseteq \hat{\hat{\mathbb{X}}}_i^a$. Hence one can replace (2.20) with the more restrictive condition $\hat{\hat{x}}_{[i]}(0) \in \hat{\hat{\mathbb{X}}}_{f,i}$ and, similarly to the previous case, solve an LP problem when sets $\hat{\hat{\mathbb{X}}}_{f,i}$, $i \in \mathcal{M}$ are polytopes.

Remark 2.2. In terms of features, the DiMPC scheme proposed in [FS12], is the closest among existing ones to our control strategy. However, substantial differences arise. First, in order to initialize our controller only the computation of states $\hat{x}(0)$ and $\hat{\hat{x}}(0)$ is required while, in [FS12] the user must supply initial assumed states over a whole control horizon. Note also that there is no systematic method for choosing initial planned trajectories in [FS12], while in our scheme the choice of initial states for the controller can be done using the procedures described as above. Second, Theorem 2.1 and finite termination of Algorithm 2.1 show that tightened constraints for lower MPC-i controllers guaranteeing satisfaction of constraints (2.2) can be always computed. Differently, in [FS12] the existence of suitable tightened constraints can be guaranteed only if coupling among subsystems is sufficiently weak. However, the DiMPC scheme in [FS12], that uses tube-based control only once, might provide a region of attraction of the origin that is larger than the one produced by our controller, and therefore the choice of the most suitable control algorithm has to be conducted on the basis of the specific application at hand.

2.4 Example

In this section, we apply the proposed DiMPC scheme to the system proposed in [FS11b] and illustrated in Figure 2.2.

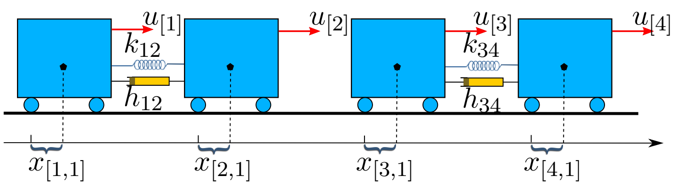


Figure 2.2: Example system.

The system is composed by four trucks, with trucks 1-2 and 3-4 coupled by a spring and a damper. Parameters values, that are the same used in [FS11b], are: $m_1 = 3$, $m_2 = 2$, $m_3 = 3$, $m_4 = 6$, $k_{12} = 0.5$, $k_{34} = 1$, $h_{12} = 0.2$, $h_{34} = 0.3$. Each truck $i \in \mathcal{M} = \{1, 2, 3, 4\}$, is a subsystem with state variables $x_{[i]} = (x_{[i,1]}, x_{[i,2]})$ and input $u_{[i]}$, where $x_{[i,1]}$ is the displacement of truck i with respect to a given equilibrium position, $x_{[i,2]}$ is the velocity of the truck i and $100u_{[i]}$ is a force applied to truck i. Subsystems are equipped with the state constraints $|x_{[i,1]}| \leq 4.5$, $|x_{[i,2]}| \leq 2$, $i \in \mathcal{M}$ and with the input constraints $|u_{[i]}| \leq 1$, $i \in \{1, 2, 3\}$ and $|u_{[4]}| \leq 2$. Moreover, the collective constraints $|x_{[i,1]} - x_{[i+1,1]}| \leq 6$, for $i = 1, 2, 3$ are enforced.
The only parent of subsystem 1 is subsystem 2 (and vice-versa) because they are dynamically coupled. Similarly subsystem 3 and 4 are arranged in a parent-child relation. The model has been discretized with sampling interval $T_s = 0.1$ sec. Modeling and discretization have been performed using the *PnPMPC-toolbox* for MatLab that offers facilities for handling

the interconnections of constrained subsystems [RBFT12] (see Appendix C).
In order to apply Algorithm 2.1, we define the decentralized controllers

$$\mathbf{K} = -\text{diag}([0.535, 0.253], [0.355, 0.168], [0.530, 0.252], [1.070, 0.507])$$

and

$$\hat{\mathbf{K}} = -\text{diag}([0.103, 0.112], [0.067, 0.074], [0.098, 0.111], [0.206, 0.225])$$

that guarantee fulfillment of Assumption 2.1.
For the following sets

$$\hat{\hat{\mathbb{X}}}_i = \left\{ |\hat{\hat{x}}_{[i,1]}| \leq 2.6,\ |\hat{\hat{x}}_{[i,2]}| \leq 1.5 \right\},\ i \in \mathcal{M}$$

$$\mathbb{V}_i = \left\{ |v_{[i]}| \leq 0.9 \right\},\ i = 1, 2, 3 \qquad \mathbb{V}_4 = \left\{ |v_{[4]}| \leq 1.9 \right\}$$

Algorithm 2.1 terminates in one iteration.
The lower controllers $\mathcal{LC}_{[i]}$, $i \in \mathcal{M}$, are synthesized using explicit MPC for system (2.8) based on the quadratic cost function

$$V_i^N(\hat{\hat{x}}_{[i]}(t), v_{[i]}(t : t + N_i - 1)) = \\ \sum_{k=t}^{t+N_i-1} (||\hat{\hat{x}}_{[i]}(k)||_{Q_i} + ||v_{[i]}(k)||_{R_i}) + ||\hat{\hat{x}}_{[i]}(t + N_i)||_{S_i}$$

where $N_i = 10$, $i \in \mathcal{M}$,

$$Q_1 = \begin{bmatrix} 0.120 & -0.014 \\ -0.014 & 0.144 \end{bmatrix},\ Q_2 = \begin{bmatrix} 0.127 & -0.010 \\ -0.010 & 0.145 \end{bmatrix},$$

$$Q_3 = \begin{bmatrix} 0.120 & -0.014 \\ -0.014 & 0.144 \end{bmatrix},\ Q_4 = \begin{bmatrix} 0.091 & -0.029 \\ -0.029 & 0.138 \end{bmatrix},$$

$$R_1 = 0.184,\ R_2 = 0.294,\ R_3 = 0.186,\ R_4 = 0.075$$

and $S_i = \begin{bmatrix} 8.901 & 0.700 \\ 0.700 & 0.624 \end{bmatrix}$, $i \in \mathcal{M}$. The matrices Q_i, R_i and S_i, $i \in \mathcal{M}$ have been computed so as to guarantee stability of the origin of (2.8) and (2.16). Moreover they guarantee stability for the DiMPC scheme proposed in [FS12]. Constraints for the MPC-i problem are the dynamics (2.8), state constraints $\hat{\hat{x}}_{[i]} \in \hat{\hat{\mathbb{X}}}_i$, input constraints $v_{[i]} \in \mathbb{V}_i$ and suitable polytopic terminal constraints $\hat{\hat{\mathbb{X}}}_{f,i}$ for guaranteeing closed-loop stability of the origin of (2.8), see [RM09]. Explicit MPC-i regulators have been computed using the MPT Toolbox [KGB04] and then sets $\hat{\hat{\mathbb{X}}}_i^a$ are known. Initial states $\hat{\mathbf{x}}(0)$ and $\hat{\hat{\mathbf{x}}}(0)$ have been computed from $\mathbf{x}(0)$ solving an LP problem, as explained in Section 2.3. Figures 2.3 and 2.4 show state and control trajectories obtained using a CeMPC scheme, the DiMPC method proposed in [FS12] and our distributed control scheme. In all cases, the initial state is $\mathbf{x}(0) = (1.8, 0, -0.5, 0, 1, 0, -1, 0)$. For CeMPC, we have used the quadratic cost function

$$V^N(\mathbf{x}(t), \mathbf{u}(t : t+N+1)) = \sum_{k=t}^{t+N-1} (||\mathbf{x}(k)||_{\mathbf{Q}} + ||\mathbf{u}(k)||_{\mathbf{R}}) + ||\mathbf{x}(t+N)||_{\mathbf{S}}$$

where $N = 10$ and $\mathbf{Q} = \text{diag}(Q_1, \ldots, Q_M)$, $\mathbf{R} = \text{diag}(R_1, \ldots, R_M)$ and $\mathbf{S} = \text{diag}(S_1, \ldots, S_M)$. These matrices guarantee stability of the origin of the collective system. CeMPC includes constraints on all states $x_{[i,j]}, i \in \mathcal{M}, j \in \{1, 2\}$, on inputs $u_{[i]}$ and suitable terminal constraints for stability of the origin.
Figures 2.3 and 2.4 show that the performance of our distributed control scheme is comparable with the performance of CeMPC and the distributed control proposed in [FS12].
Table 2.1 highlights the computational advantages brought about by our method: since we can use standard explicit MPC for synthesizing local controllers, the average time for computing the inputs in each sampling interval is considerably reduced compared to the other methods[2]. Using the DiMPC scheme proposed in [FS12], local optimization problems have a number of constraints comparable to the ones of CeMPC. Moreover, since the local controllers of [FS12] depend of planned trajectories sent by the parent subsystems, explicit MPC methods cannot be applied out of the box. For these reasons, in our example computational savings brought about by the DiMPC in [FS12] are limited.

2.5 Final comments

In this chapter we proposed a novel DiMPC scheme for linear constrained systems. We showed that the availability of a decentralized static state-feedback controller allows one to systematically design distributed controllers for each subsystem that stabilize the origin of the closed-loop system

[2] All simulations have been done using a MacOS 10.7.5, with processor Intel Core i5, 1.7 GHz, MatLab r2013a, solver CPLEX [IBM11], YALMIP [Lö04] and MPT [KGB04].

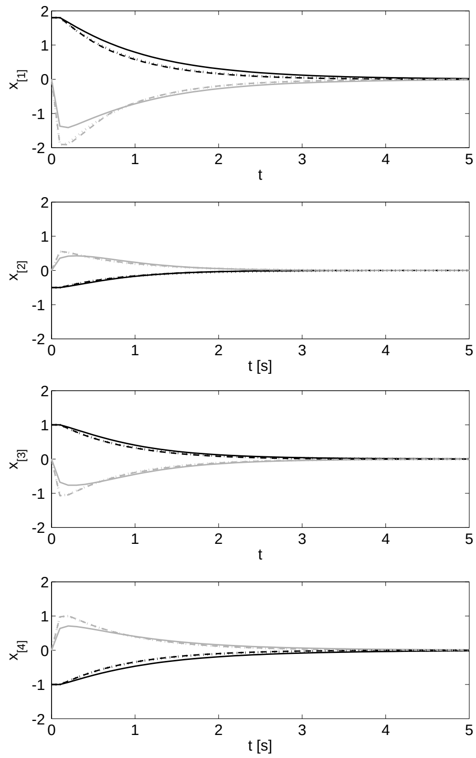


Figure 2.3: State trajectories: positions (black) and velocities (grey) of the trucks, using centralized control (dotted lines), distributed control proposed in [FS12] (dashed lines) and our distributed control (solid lines).

	# Opt	Min time	Mean time	Max time	Speed-up
CeMPC	50	0.0343	0.0382	0.0472	1
DiMPC of [FS12]	200	0.0250	0.0289	0.0457	1.32
Our DiMPC	0	0.0016	0.0017	0.0018	22.47

Table 2.1: Performance of CeMPC, DiMPC proposed in [FS12] and our DiMPC. # *Opt* is the number of solved optimization problem. *Min Time*, *Mean Time* and *Max Time* represent respectively the minimum, the mean and the maximum time expressed in seconds to solve the optimization problem. *Speed-up* is defined as the ratio between *Mean Time* of MPC respect to each analyzed control scheme.

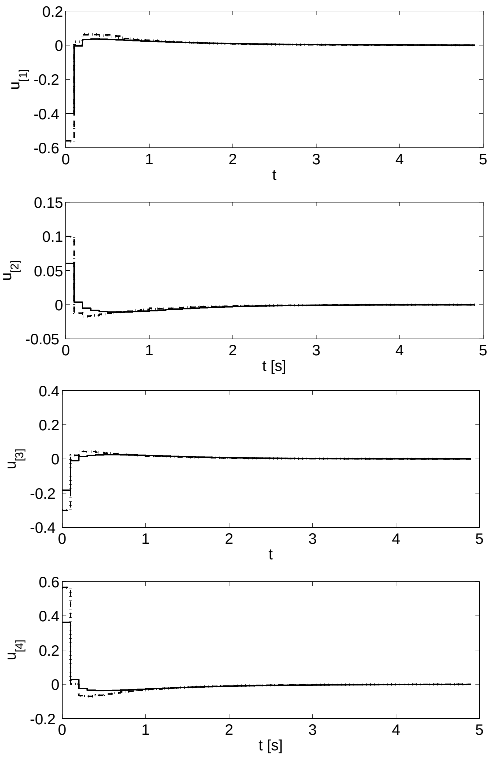


Figure 2.4: Control variables, using centralized control (dotted lines), distributed control proposed in [FS12] (dashed lines) and our distributed control (solid lines).

and exchange only planned states at each sampling time. Furthermore, local controllers have a two-layer structure where MPC is exploited at the lower layer to guarantee constraint satisfaction. In future research, we will study how to account for model uncertainties in the scheme and how to avoid offline centralized computations in Algorithm 2.1 using methods in [RKF10] and [RKF11]. If the state of each subsystem is not available, the proposed DiMPC can be directly used together with the DSE proposed in Chapter 3 although further research is needed for assessing the stability properties of the closed-loop system. In Part II of the thesis, based on tube MPC, we propose DeMPC and DiMPC schemes with PnP capabilities.

2.6 Appendix

2.6.1 Proof of Theorem 2.1

We introduce the following preliminary result.

Proposition 2.2. *Under Assumptions 2.1 and 2.3, if* (2.21) *and* (2.22) *hold, then one has*

$$\begin{aligned}&\hat{\hat{x}}_{[i]}(k) \in \hat{\hat{\mathbb{X}}}_i,\ v_{[i]}(k) \in \mathbb{V}, \forall i \in \mathcal{M},\ \forall k \in 0:t \Rightarrow \\ &\mathbf{x}(k) \in \bar{\mathbb{X}},\ \mathbf{u}(k) \in \bar{\mathbb{U}},\ \forall k \in 0:t\end{aligned} \tag{2.26}$$

Proof. For given sets $\hat{\hat{\mathbb{X}}}_i$, $i \in \mathcal{M}$, using the definition of RPI sets $\hat{\mathbb{Z}}_i$ and $\mathbb{Z}$ one has

$$\begin{aligned}&\hat{z}_{[i]}(0) \in \hat{\mathbb{Z}}_i,\ \hat{\hat{x}}_{[j]}(k) \in \hat{\hat{\mathbb{X}}}_j, \forall j \in \mathcal{M},\ \forall k \in 0:t \Rightarrow \\ &\quad \hat{z}_{[i]}(k) \in \hat{\mathbb{Z}}_i,\ \forall k \in 0:t+1\end{aligned} \tag{2.27}$$

$$\begin{aligned}&\mathbf{z}(0) \in \mathbb{Z},\ \hat{z}_{[j]}(k) \in \hat{\mathbb{Z}}_j, \forall j \in \mathcal{M},\ \forall k \in 0:t \Rightarrow \\ &\quad \mathbf{z}(k) \in \mathbb{Z},\ \forall k \in 0:t+1\end{aligned} \tag{2.28}$$

Defining the proposition P as "$\mathbf{z}(0) \in \mathbb{Z},\ \hat{z}_{[j]}(0) \in \hat{\mathbb{Z}}_j,\ \hat{\hat{x}}_{[j]}(k) \in \hat{\hat{\mathbb{X}}}_j, \forall j \in \mathcal{M},\ \forall k \in 0:t$", from (2.27) and (2.28), it follows that

$$P \Rightarrow \mathbf{z}(k) \in \mathbb{Z},\ \hat{z}_{[i]}(k) \in \hat{\mathbb{Z}}_i,\ \forall k \in 0:t+1 \tag{2.29}$$

Since in P we assume $\hat{\hat{x}}_{[j]}(k) \in \hat{\hat{\mathbb{X}}}_j$, from (2.12) and (2.18), inclusions $\hat{z}_{[i]}(k) \in \hat{\mathbb{Z}}_i$, $k \in 0:t+1$ can be replaced by

$$\hat{x}_{[i]}(k) \in \hat{\mathbb{X}}_i,\ k \in 0:t+1 \tag{2.30}$$

Moreover, from (2.11), (2.18) and (2.30), inclusions $\mathbf{z}(k) \in \mathbb{Z}$, $k \in 0:t+1$ can be replaced by $\mathbf{x} \in \mathbb{X}$, $k \in 0:t+1$. Therefore (2.29) becomes

$$P \Rightarrow \mathbf{x}(k) \in \bar{\mathbb{X}},\ \hat{x}_{[i]}(k) \in \hat{\mathbb{X}}_i,\ \forall k \in 0:t+1 \tag{2.31}$$

From the expression of $u_{[i]}$ in (2.10) and the definition of $\bar{\mathbb{U}}$ in (2.19) one has

$$\begin{aligned}&\mathbf{z}(k) \in \mathbb{Z},\ \hat{z}_{[j]}(k) \in \hat{\mathbb{Z}}_j,\ v_{[j]}(k) \in \mathbb{V}_j,\ \forall j \in \mathcal{M}, \\ &\quad \forall k \in 0:t \Rightarrow \mathbf{u}(k) \in \bar{\mathbb{U}},\ \forall k \in 0:t\end{aligned}$$

that, together with (2.29), gives

$$P \text{ and } v_{[j]}(k) \in \mathbb{V}_j,\ \forall j \in \mathcal{M}, \forall k \in 0:t \Rightarrow \\ \mathbf{u}(k) \in \bar{\mathbb{U}},\ \forall k \in 0:t \tag{2.32}$$

Noting that when (2.21) and (2.22) hold P becomes "$\hat{\hat{x}}_{[j]}(k) \in \hat{\hat{\mathbb{X}}}_j, \forall j \in \mathcal{M},\ \forall k \in 0:t$", formula (2.26) follows from (2.31) and (2.32). □

Proof of Theorem 2.1.

Proof. Asymptotic stability of the origin of (2.3)-(2.9) equipped with the controller (2.16) follows from the assumed stabilizing properties of $\kappa_i(\hat{\hat{x}})$, $i \in \mathcal{M}$, Assumption 2.1 and the cascade structure of (2.15). Lower controllers (2.16) guarantee that $\hat{\hat{\mathbf{x}}}(t) \to \mathbf{0}_n$ as $t \to +\infty$ and, from (2.15), it is easy to show that $\hat{\mathbf{x}}(t) \to \mathbf{0}_n$ and $\mathbf{x}(t) \to \mathbf{0}_n$, irrespectively of $\hat{\mathbf{x}}(0) \in \mathbb{R}^n$ and $\mathbf{x}(0) \in \mathbb{R}^n$. In particular, convergence to zero of $(\mathbf{x}(t), \hat{\mathbf{x}}(t), \hat{\hat{\mathbf{x}}}(t))$ is guaranteed for initial states verifying (2.21) and (2.22) and satisfaction of constraints (2.23) follows from Proposition 2.2. □

2.6.2 Proof of Proposition 2.1

Proof. Let $\hat{\hat{\mathbb{X}}}_i$, $\mathbb{V}_i$, $i \in \mathcal{M}$ be sets verifying Assumption 2.3 and define $\hat{\hat{\mathbb{X}}} = \prod_{i\in\mathcal{M}} \hat{\hat{\mathbb{X}}}_i$. For matrices $\mathbf{A} + \mathbf{B}\mathbf{K}$ and $\mathbf{A_D} + \mathbf{B}\hat{\mathbf{K}}$ that are Schur, consider the map $\mathcal{R}(\hat{\hat{\mathbb{X}}}) = (\hat{\mathbb{Z}}, \mathbb{Z})$ where $\hat{\mathbb{Z}}_i$ and $\mathbb{Z}$ are the mRPI sets for (2.14) and (2.17), respectively and $\hat{\mathbb{Z}} = \prod_{i\in\mathcal{M}} \hat{\mathbb{Z}}_i$. One has that $\mathcal{R}(\{\mathbf{0}_n\}) = (\{\mathbf{0}_n\}, \{\mathbf{0}_n\})$ and $\mathcal{R}$ is continuous at $\{0\}$ in the sense that, denoting with B_n the unit ball in $\mathbb{R}^n$, one has $\forall \varepsilon \geq 0\ \exists \delta > 0\ :\ \hat{\hat{\mathbb{X}}} \subseteq \delta B_n \Rightarrow \mathcal{R}(\hat{\hat{\mathbb{X}}}) \subseteq \varepsilon B_{2n}$.
Also the map $\mathcal{S}(\hat{\hat{\mathbb{X}}}) = \bar{\mathbb{X}}$ defined by (2.18), where $\hat{\mathbb{Z}}_i$ are the mRPI sets for (2.14) and $\mathbb{Z}$ is the mRPI set for (2.17), is continuous at $\{\mathbf{0}_n\}$ and $\mathcal{S}(\{\mathbf{0}_n\}) = \{\mathbf{0}_n\}$. Then, for a given $\varepsilon > 0$ and sets $\hat{\hat{\mathbb{X}}}_i$, $i \in \mathcal{M}$ containing the origin in their interior, there exists $\eta > 0$ such that

$$\mathcal{S}(\eta\hat{\hat{\mathbb{X}}}) \subseteq \varepsilon B_n. \tag{2.33}$$

Since the origin is in the interior of $\mathbb{X}$, there exists $\varepsilon > 0$ such that $\varepsilon B_n \subseteq \mathbb{X}$, and (2.33) shows that sets $\eta\hat{\hat{\mathbb{X}}}_i$, $i \in \mathcal{M}$ yield a set $\bar{\mathbb{X}}$ verifying $\bar{\mathbb{X}} \subseteq \mathbb{X}$.
For proving the inclusion $\bar{\mathbb{U}} \subseteq \mathbb{U}$, we use a similar argument. The map $\mathcal{U}(\hat{\hat{\mathbb{X}}}, \mathbb{V}) = \bar{\mathbb{U}}$ defined by (2.19), where $\hat{\mathbb{Z}}_i$ are the mRPI sets for (2.14) and $\mathbb{Z}$ is the mRPI set for (2.17), is continuous at $(\{\mathbf{0}_n\}, \{\mathbf{0}_m\})$ and verifies

$\mathcal{U}(\{\mathbf{0}_n\},\{\mathbf{0}_m\}) = \{\mathbf{0}_m\}$. Therefore, for a given $\varepsilon > 0$ and sets $\mathbb{V}_i$, $\hat{\hat{\mathbb{X}}}_i$, $i \in \mathcal{M}$ containing the origin in their interior there exists $\eta > 0$ such that $\mathcal{U}(\eta\hat{\hat{\mathbb{X}}}, \eta\mathbb{V}) \subseteq \varepsilon B_m$. The proof is concluded by noting that $\exists \varepsilon > 0 : \varepsilon B_m \subseteq \mathbb{U}$, since the origin is in the interior of $\mathbb{U}$. $\square$

Chapter 3

Decentralized design of distributed bounded-error state estimation for partitioned systems

Contents

3.1 Introduction

In this chapter we propose a novel partition-based state estimator for linear discrete-time subsystems affected by bounded disturbances. Similarly to the methods proposed in [FFTS10b] and [FS11a], our scheme is distributed in the sense that computation of local state estimates can be performed in parallel but only after each estimator has received suitable pieces of information from parent subsystems. Moreover, as in [FS11a], state estimators account for constraints on subsystem disturbances and guarantee the fulfillment of *a priori* specified constraints on local estimation errors. Differently from the scheme in [FFTS10b], that is based on moving horizon estimation, and similarly to [FS11a], local estimators have a Luenberger structure and therefore do not require the online solution to optimization problems. Furthermore, most operations needed for the design of a local estimator can be performed using computational resources collocated with the corresponding subsystem and the only centralized step requires the analysis of a system whose order is equal to the number of subsystems.

In order to guarantee convergence of state estimates in absence of disturbances and fulfillment of prescribed constraints on the estimation error, we rely on the notion of practical RPI (pRPI) developed in [RKF11] that is applied to the error dynamics. We also highlight that most of the appealing computational features of our method directly follow from results reported in [RKF11] for the case of polytopic constraints. Since pRPI implies worst-case robustness against the propagation of errors between subsystems, our design method involves some degree of conservatism and can not be always applied. Therefore, in the attempt of maximizing chances of successful design, we provide guidelines on the choice of local estimator parameters. We also show that when subsystems are added or removed, the state estimation scheme can be updated with limited efforts. More in detail, we prove that, in order to preserve convergence and fulfillment of constraints on estimation errors, (i) plug in of a subsystem requires the decentralized design of local estimators for the subsystem and its children only, besides the re-execution of the centralized step; (ii) plug out of a subsystem does not require any update. Compared to the distributed state estimator proposed in [FS11a], our scheme has several distinctive features. First, the use of the notion of practical robust positive invariance instead of the more standard concept of robust positive invariance, allows us to achieve, in some cases, a less conservative design procedure (see [RKF10] for a discussion on the degree of conservativeness of various invariance concepts). Second, our local estimators can take advantage of the knowledge of parents' outputs and this can be fundamental for successful estimator design, and demonstrated in Section 3.6 through an example. Third, the method in [FS11a] requires to analyze in a centralized fashion the stability of a system whose order is equal to the sum of the orders of all subsystems. However, as described in Section 3.6.4, design decentralization can introduce more conservativity.

The chapter is structured as follows. Local state estimators are described in Section 3.2. In Section 3.3 we introduce practical robust decentralized invariance and show how it can be applied for guaranteeing convergence of estimators and constraint satisfaction. In Section 3.4 we detail the design of local estimators. Section 3.5 describes how to retune the estimator when subsystems are added or removed from the network. In Section 3.6 we illustrate the use of the distributed state estimator for reconstructing the states of a PNS and compare our method with the state estimation scheme in [FS11a]. Section 3.7 is devoted to conclusions.

3.2 Distributed state estimator

We consider a large-scale discrete-time LTI system

$$\begin{aligned} \mathbf{x}^+ &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{D}\mathbf{d} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{3.1}$$

composed of M subsystems, as described in Section 1.5. In this chapter we will focus our attention on the problem of bounded-error state estimation, therefore we consider constraints on model disturbances on each subsystem, i.e.

$$d_{[i]}(t) \in \mathbb{D}_i, \ \forall t \geq 0. \tag{3.2}$$

Assumption 3.1. *The sets $\mathbb{D}_i$, $i \in \mathcal{M}$ are C-sets.*

Next, we propose a DSE for (3.1). We define for $i \in \mathcal{M}$ the local state estimator

$$\tilde{\Sigma}_{[i]}: \quad \tilde{x}^+_{[i]} = A_{ii}\tilde{x}_{[i]} + B_i u_{[i]} - L_{ii}(y_{[i]} - C_i\tilde{x}_{[i]}) + \sum_{j\in\mathcal{N}_i} A_{ij}\tilde{x}_{[j]} - \sum_{j\in\mathcal{N}_i}\tilde{\delta}_{ij}L_{ij}(y_{[j]} - C_j\tilde{x}_{[j]}) \tag{3.3}$$

where $\tilde{x}_{[i]} \in \mathbb{R}^{n_i}$ is the state estimate, $L_{ij} \in \mathbb{R}^{n_i\times p_j}$ are gain matrices and $\tilde{\delta}_{ij} \in \{0,1\}$. Hereafter we assume $\tilde{\delta}_{ij} = 0$ and $L_{ij} = \mathbf{0}_{n_i\times p_j}$ if $j \notin \mathcal{N}_i$. This implies that $\tilde{\Sigma}_{[i]}$ depends only on local variables ($\tilde{x}_{[i]}$, $u_{[i]}$ and $y_{[i]}$) and parents' variables ($\tilde{x}_{[j]}$ and $y_{[j]}$, $j \in \mathcal{N}_i$). Binary parameters $\tilde{\delta}_{ij}$, $j \in \mathcal{N}_i$ can be chosen to take advantage of the knowledge of parents' outputs ($\tilde{\delta}_{ij} = 1$) or to reduce the amount of information received from parents ($\tilde{\delta}_{ij} = 0$).
Defining the state estimation error as

$$e_{[i]} = x_{[i]} - \tilde{x}_{[i]}, \tag{3.4}$$

from (1.2), (3.3) and (3.4), we obtain the local error dynamics

$$e^+_{[i]} = \bar{A}_{ii}e_{[i]} + \sum_{j\in\mathcal{N}_i}\bar{A}_{ij}e_{[j]} + D_i d_{[i]} \tag{3.5}$$

where $\bar{A}_{ii} = A_{ii} + L_{ii}C_i$ and $\bar{A}_{ij} = A_{ij} + \tilde{\delta}_{ij}L_{ij}C_j$, $i \neq j$. Our main goal is to solve the following problem.

Problem 3.1. Design local state estimators $\tilde{\Sigma}_{[i]}$, $i \in \mathcal{M}$ that

(a) are nominally convergent, i.e. when $\mathbb{D} = \{\mathbf{0}_r\}$ it holds

$$||e_{[i]}(t)|| \to \mathbf{0}_{n_i} \text{ as } t \to \infty \tag{3.6}$$

(b) guarantee

$$e_{[i]}(t) \in \mathbb{E}_i, \ \forall t \geq 0 \tag{3.7}$$

where $\mathbb{E}_i \subseteq \mathbb{R}^{n_i}$ are prescribed sets containing the origin in their interior.

Defining the collective variable $\mathbf{e} = (e_{[1]}, \dots, e_{[M]}) \in \mathbb{R}^n$, from (3.5) one obtains the collective dynamics of the estimation error

$$\mathbf{e}^+ = \bar{\mathbf{A}}\mathbf{e} + \mathbf{D}\mathbf{d} \tag{3.8}$$

where the matrix $\bar{\mathbf{A}}$ is composed by blocks $\bar{A}_{ij}$, $i,j \in \mathcal{M}$.
We equip system (3.8) with constraints $\mathbf{e} \in \mathbb{E} = \prod_{i\in\mathcal{M}}\mathbb{E}_i$ and $\mathbf{d} \in \mathbb{D}$. In Section 3.3 we address Problem 3.1 under the following assumptions

Assumption 3.2. *The matrices $\bar{A}_{ii}$, $i \in \mathcal{M}$ are Schur.*

Assumption 3.3. *The sets $\mathbb{E}_i$, $i \in \mathcal{M}$ are PC-sets.*

We highlight that if $\mathbf{L}$ is such that $\bar{\mathbf{A}}$ is Schur, then property (3.6) holds. If, in addition, Assumptions 3.3 and 3.1 hold, then there exists a RPI set $\Omega \subset \mathbb{E}$ for the constrained system (3.8) (see [KG98]) and $\mathbf{e}(0) \in \Omega$ guarantees property (3.7). Remarkably, when sets $\mathbb{E}_i$ and $\mathbb{D}_i$ are polytopes, an RPI set Ω can be found solving a LP problem (see [RKKM05], [RB10]). However the LP problem includes the collective model (3.1) in the constraints and computations become prohibitive for large n.
In absence of coupling between subsystems (i.e. $A_{ij} = 0$, $i \neq j$) the estimator dynamics (3.3) and error dynamics (3.5) are decoupled as well. Therefore, under Assumptions 3.2, 3.3 and 3.1, properties (3.6) and (3.7) can be guaranteed computing RPI sets $\Omega_i \subseteq \mathbb{E}_i$ for each local error dynamics and requiring $e_{[i]}(0) \in \Omega_i$. Furthermore, if $\mathbb{E}_i$ and $\mathbb{D}_i$ are polytopes, the computation of sets Ω_i, $i \in \mathcal{M}$ amounts to the solution of M LP problems that can be solved in parallel using computational resources collocated with subsystems. In order to propose a partially decentralized design procedure in presence of coupling between subsystems one has to take into account how coupling propagates errors between subsystems. As we will show in the next section, the notion of pRPI, proposed in [RKF11] allows one to study precisely this issue and offers a computationally feasible, yet conservative, procedure for solving Problem 3.1.

3.3 Practical robust positive invariance for state estimation

In this section, we show how the main results of [RKF11], applied to the error dynamics (3.5) equipped with constraints (3.2) and (3.7), allow one to guarantee properties (a) and (b) of Problem 3.1.
Given a collection of sets $\mathbb{S} = \{\mathbb{S}_i,\ i \in \mathcal{M}\}$, $\mathbb{S}_i \subset \mathbb{R}^{n_i}$ and a set $\Theta \subset \mathbb{R}^M_{0+}$, we define a parameterized family of sets $\mathcal{S}(\mathbb{S}, \Theta) = \{(\theta_1\mathbb{S}_1, \ldots, \theta_M\mathbb{S}_M) : \theta \in \Theta\}$, where $\theta = (\theta_1, \ldots, \theta_M)$. Intuitively, scalars θ_i can be interpreted as scaling factors.

Definition 3.1. The family of sets $\mathcal{S}(\mathbb{S}, \Theta)$ is pRPI for the constrained local error dynamics given by (3.5), (3.2) and (3.7), if, for all $i \in \mathcal{M}$ and all $(\theta_1\mathbb{S}_1, \dots, \theta_M\mathbb{S}_M) \in \mathcal{S}(\mathbb{S}, \Theta)$, one has

$$\theta_i\mathbb{S}_i \subseteq \mathbb{E}_i \tag{3.9a}$$

$$\bar{A}_{ii}\theta_i\mathbb{S}_i \oplus \bigoplus_{j\in\mathcal{N}_i} \bar{A}_{ij}\theta_j\mathbb{S}_j \oplus D_i\mathbb{D}_i \subseteq \theta_i^+\mathbb{S}_i \tag{3.9b}$$

$$(\theta_1^+\mathbb{S}_1, \dots, \theta_M^+\mathbb{S}_M) \in \mathcal{S}(\mathbb{S}, \Theta) \tag{3.9c}$$

Assumption 3.4. *The sets $\mathbb{S}_i$, $i \in \mathcal{M}$ are PC-sets.*

The main issue we will address in the sequel is the following: given $\mathbb{S}$, is there a nonempty set $\Theta \subset \mathbb{R}^M_{0+}$ such that the family $\mathcal{S}(\mathbb{S}, \Theta)$ is pRPI? In order to provide an answer, in [RKF11] it is proposed to first derive the dynamics of the scaling factors θ_i. More precisely, for all $i, j \in \mathcal{M}$ we set

$$\mu_{ij} = \begin{cases} \min_{\mu\geq 0}\{\mu : \bar{A}_{ij}\mathbb{S}_j \subseteq \mu\mathbb{S}_i\} & \text{if } i = j \text{ or } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

$$\alpha_i = \min_{\beta\geq 0}\{\beta : D_i\mathbb{D}_i \subseteq \beta\mathbb{S}_i\}. \tag{3.11}$$

and define the collective dynamics of the scaling factors

$$\theta^+ = T\theta + \alpha \tag{3.12}$$

where the entries of $T \in \mathbb{R}^{M\times M}$ are $T_{ij} = \mu_{ij}$ and $\alpha = (\alpha_1, \dots, \alpha_M)$. It is easy to show that (3.12) guarantees

$$e_{[i]} \in \theta_i S_i \Rightarrow e_{[i]}^+ \in \theta_i^+ S_i.$$

For fulfilling (3.9a), let us define

$$\Theta_0 = \{\theta \in \mathbb{R}^M_{0+} : \forall i \in \mathcal{M},\ \theta_i\mathbb{S}_i \subseteq \mathbb{E}_i\}. \tag{3.13}$$

The key assumption used in [RKF11] for providing a set Θ that makes $\mathcal{S}(\mathbb{S}, \Theta)$ a pRPI family is the following one.

Assumption 3.5. *(i) T is Schur.*

(ii) The unique equilibrium point $\bar{\theta}$ of system (3.12) is such that $\bar{\theta} \in \Theta_0$.

(iii) The set Θ is an invariant set for system (3.12) and constraint set Θ_0, i.e. $\forall\theta \in \Theta \subseteq \Theta_0$, $\theta^+ \in \Theta$.

Lemma 3.1 ([RKF11])**.** *Let Assumptions 3.1-3.5 hold. Then,*

(i) *there is a non-trivial convex and compact positively invariant set* Θ *for system* (3.12) *equipped with constraints* $\theta \in \Theta_0$*;*

(ii) $\mathcal{S}(\mathbb{S}, \Theta)$ *is pRPI for* (3.5) *with constraints* (3.2) *and* (3.7)*.*

Lemma 3.1 guarantees that

$$\begin{aligned} \theta(0) \in & \Theta \text{ and } e_{[i]}(0) \in \theta_i(0)\mathbb{S}_i, \ \forall i \in \mathcal{M} \Rightarrow \\ & e_{[i]}(t) \in \theta_i(t)\mathbb{S}_i, \ \forall i \in \mathcal{M}, \ \forall t \geq 0. \end{aligned} \tag{3.14}$$

Furthermore, as shown in [RKF11], $\text{dist}(e_{[i]}(t), \bar{\theta}_i \mathbb{S}_i) \to 0$ as $t \to \infty$. In the nominal case, i.e. $\mathbb{D} = \{\mathbf{0}_r\}$, one has $\alpha = \mathbf{0}_M$ in (3.12). Then $\bar{\theta} = \mathbf{0}_M$ and property (3.6) is guaranteed. Also (3.7) holds since, from (3.14) and (3.9a) one has $e_{[i]}(t) \in \theta_i(t)\mathbb{S}_i \subseteq \mathbb{E}_i$. Therefore, Problem 3.1 is solved if we can design local state estimators fulfilling the assumptions of Lemma 3.1. A design procedure to achieve this goal is proposed in Section 3.4.

Remark 3.1. Note that, according to (3.14), the initialization of the local estimators requires to find a suitable initial state $\theta(0) \in \Theta$ for system (3.12) and this is a centralized operation. In order to allow each estimator to locally compute its initial state, one can build offline an inner box approximation $\bar{\Theta} = \prod_{i \in \mathcal{M}} [0, \bar{\theta}_i]$ contained in Θ and choose $\tilde{x}_{[i]}(0)$ such that $x_{[i]}(0) - \tilde{x}_{[i]}(0) \in [0, \bar{\theta}_i]$.

3.4 Design of local estimators

In this section, we propose a method to design the distributed state estimator presented in Sections 3.2 and 3.3. The key issue is how to compute suitable gains L_{ij} and binary variables $\tilde{\delta}_{ij}$ such that Assumption 3.5 holds. From now on we consider polytopic sets $\mathbb{E}_i$, $\mathbb{D}_i$ and $\mathbb{S}_i$, $i \in \mathcal{M}$ verifying Assumptions 3.1, 3.3 and 3.4. Without loss of generality we can write

$$\begin{aligned} \mathbb{E}_i &= \{e_{[i]} \in \mathbb{R}^{n_i} : h_{i,\tau}^T e_{[i]} \leq 1, \forall \tau \in 1 : \bar{\tau}_i\} \\ &= \{e_{[i]} \in \mathbb{R}^{n_i} : \mathcal{H}_i e_{[i]} \leq \mathbf{1}_{\bar{\tau}_i}\} \end{aligned} \tag{3.15a}$$

$$\begin{aligned} \mathbb{D}_i &= \{d_{[i]} \in \mathbb{R}^{r_i} : f_{i,\upsilon}^T d_{[i]} \leq 1, \forall \upsilon \in 1 : \bar{\upsilon}_i\} \\ &= \{d_{[i]} \in \mathbb{R}^{r_i} : \mathcal{F}_i d_{[i]} \leq \mathbf{1}_{\bar{\upsilon}_i}\} \end{aligned} \tag{3.15b}$$

$$\begin{aligned} \mathbb{S}_i &= \{s_{[i]} \in \mathbb{R}^{n_i} : g_{i,\psi}^T s_{[i]} \leq 1, \forall \psi \in 1 : \bar{\psi}_i\} \\ &= \{s_{[i]} \in \mathbb{R}^{n_i} : \mathcal{G}_i s_{[i]} \leq \mathbf{1}_{\bar{\psi}_i}\} \end{aligned} \tag{3.15c}$$

where $\mathcal{H}_i = (h_{i,1}^T, \dots, h_{i,\bar{\tau}_i}^T) \in \mathbb{R}^{\bar{\tau}_i \times n_i}$, $\mathcal{F}_i = (f_{i,1}^T, \dots, f_{i,\bar{v}_i}^T) \in \mathbb{R}^{\bar{v}_i \times r_i}$ and $\mathcal{G}_i = (g_{i,1}^T, \dots, g_{i,\bar{\psi}_i}^T) \in \mathbb{R}^{\bar{\psi}_i \times n_i}$. The design procedure is summarized in Algorithm 3.1 that is composed by three parts.

Algorithm 3.1 Computation a pRPI family of sets $\mathcal{S}(\mathbb{S}, \Theta)$

Input: polytopic sets $\mathbb{E}_i$, $\mathbb{D}_i$, $i \in \mathcal{M}$ verifying Assumptions 3.1 and 3.3.
Output: A pRPI family of sets $\mathcal{S}(\mathbb{S}, \Theta)$.

(A) *Decentralized steps.* For all $i \in \mathcal{M}$,

- (I) compute the matrix L_{ii} such that $\bar{A}_{ii}$ is Schur and has as many zero eigenvalues as possible;
- (II) compute a λ_i-contractive set $\mathbb{S}_i$ for

$$e_{[i]}^+ = \bar{A}_{ii} e_{[i]}$$

  verifying $\mathbb{S}_i \subseteq \mathbb{E}_i$ and set $\mu_{ii} = \lambda_i$;
- (III) compute α_i as in (3.11).

(B) *Distributed steps.* For all $i \in \mathcal{M}$,

- (I) if $\tilde{\delta}_{ij} = 1$, compute the matrix L_{ij}, $\forall j \in \mathcal{N}_i$ solving

$$\min_{L_{ij}} ||\mathcal{G}_i \bar{A}_{ij} \mathcal{G}_j^\flat||_p \tag{3.16}$$

  where p is a generic norm.
- (II) compute μ_{ij} as in (3.10).

(C) *Centralized steps*

- (I) if matrix T is not Schur **stop**;
- (II) compute set Θ_0 as in (3.13) and the equilibrium point $\bar{\theta}$ of system (3.12). If $\bar{\theta} \notin \Theta_0$ **stop**;
- (III) compute the maximal invariant set Θ_∞ of system (3.12) equipped with constraint Θ_0;
- (IV) compute an inner box approximation $\bar{\Theta}$ of Θ_∞.

Operations in Part (A) can be executed in parallel using computational resources associated with subsystems, i.e. in a decentralized fashion. Steps in Part (B) have a distributed nature, meaning that computations are decentralized but they can be performed only after each subsystem has received suitable pieces of information from its parents. Finally, design steps in Part (C) require centralized computations involving only the M-th order system (3.12). Next, we comment each step of Algorithm 3.1 in details.

3.4.1 Part (A)

Step (AI) is the easiest one and it can be performed only if pairs (A_{ii}, C_i), $i \in \mathcal{M}$ are detectable. The requirement of placing eigenvalues of $\bar{A}_{ii}$ in zero is motivated by Step (AII).
The computation of sets $\mathbb{S}_i$ as in Step (AII) has been suggested in [RKF11] and it is based on the argument that sets $(1-\lambda_i)$ can be used for compensating coupling terms in the error dynamics. Remarkably, using the efficient procedures proposed in [RB10], the computation of a set $\mathbb{S}_i$ amounts to solving the optimization problem

$$\mathcal{P}_i(\mathbb{S}_i^0, k_i): \min_{\gamma_i, \beta_i, \{\mathbb{S}_i^s\}_{s=1}^{k_i}} \gamma_i \tag{3.17a}$$

$$\gamma_i \in [0, 1), \quad \mathbb{S}_i^{k_i} \subseteq \gamma_i \mathbb{S}_i^0 \tag{3.17b}$$

$$\beta_i \in \mathbb{R}_+, \quad \bigoplus_{s=0}^{k_i-1} \mathbb{S}_i^s \subseteq \beta_i \mathbb{E}_i \tag{3.17c}$$

$$\mathbb{S}_i^s = \bar{A}_{ii}^s \mathbb{S}_i^0, \ \forall s = 1, \dots, k_i \tag{3.17d}$$

where $k_i \in \mathbb{N}$ and the set $\mathbb{S}_i^0 \subset \mathbb{R}^{n_i}$ are provided as inputs. In particular, (3.17) is an LP problem and the set $\mathbb{S}_i$ can be obtained as $\mathbb{S}_i = \beta_i^{-1} \bigoplus_{s=0}^{k-1} \mathbb{S}_i^s$. Furthermore, the contractivity parameter is $\lambda_i = \frac{\xi_i + \gamma_i^* - 1}{\delta_i}$, where γ_i^* is a solution to (3.17) and $\xi_i = \min_\xi \{\xi \in \mathbb{R}_+ : \bigoplus_{s=0}^{k_i-1} \mathbb{S}_i^s \subseteq \xi \mathbb{S}_i^0, \ \xi \geq 1\}$. Note that also ξ_i can be computed solving an LP problem. As shown in [RB10], since the matrix $\bar{A}_{ii}$ is Schur, then, given a PC-polytopic set $\mathbb{S}_i^0$, there exists a sufficiently large k_i such that problem (3.17) is feasible. Moreover, if all eigenvalues of $\bar{A}_{ii}$ are zero, feasibility of (3.17) can be guaranteed setting $k_i = n_i$. Indeed since $\bar{A}_{ii}^{n_i} = \mathbf{0}_{n_i \times n_i}$ we have $\mathbb{S}_i^{n_i} = \{\mathbf{0}_{n_i}\}$ and hence, irrespectively of $\mathbb{S}_i^0$, constraints (3.17b) hold with $\alpha_i = 0$. Moreover, since from (3.17d) sets $\{\mathbb{S}_i^s\}_{s=1}^{k_i-1}$ are polytopes containing the origin, then there exists β_i such that constraints (3.17c) hold. We highlight that the scalar μ_{ii} computed as in (3.10) is equal to the contractivity parameter λ_i.

Step (AIII) focuses on the computation of scalars α_i. From (3.11) and (3.15b), using procedures proposed in [KG98], we have $\alpha_i = \max_{\psi\in 1:\bar{\psi}_i}\{z_i\}$ where

$$\begin{aligned} z_i = &\max_{w_{[i]}} \; g_{i,\psi} D_i d_{[i]} \\ &\mathcal{F}_i d_{[i]} \leq \mathbf{1}_{\bar{v}_i} \end{aligned} \tag{3.18}$$

Therefore, Step (AIII) requires the solution to the ψ_i LP problems (3.18).

Remark 3.2. When pairs (A_{ii}, C_i) are observable, Step (AI) correspond to the synthesis of dead beat observers. One of the main limitation of dead beat observers is that they are not optimal when the subsystem is affected by stochastic disturbances. Generalizations of Steps (AI) and (AII) to this case will be considered in future research.

3.4.2 Part (B)

For the computation of matrices L_{ij} and parameters μ_{ij}, each subsystem $\Sigma_{[i]}$ needs to receive the matrix C_j and the set $\mathbb{S}_j$ from parents $j \in \mathcal{N}_i$ such that $\tilde{\delta}_{ij} = 1$.

In Step (BI), if $\tilde{\delta}_{ij} = 1$, the computation of matrices L_{ij}, $j \in \mathcal{N}_i$ is required. Since the choice of L_{ij} affects the coupling term $\bar{A}_{ij}$ and hence the Schurness of matrix T, we propose to reduce the magnitude of coupling by minimizing the magnitude of $\bar{A}_{ij}$ in (3.16), where $\mathcal{G}_i$ and $\mathcal{G}_j^\flat$ allow us to take into account the size of sets $\mathbb{S}_i$ and $\mathbb{S}_j$, respectively. More precisely, it can be shown that the term $||\mathcal{G}_i \bar{A}_{ij} \mathcal{G}_j^\flat||_p$ is a measure of how much the coupling term $\bar{A}_{ij} s_{[j]}$, $j \in \mathcal{N}_i$ affects the fulfillment of the constraint $s_{[i]} \in \mathbb{S}_i$. As an example, we highlight that the minimization of $\|\mathcal{G}_i \bar{A}_{ij} \mathcal{G}_j^\flat\|_1$ in (3.16) amounts to an LP problem and the minimization of $\|\mathcal{G}_i \bar{A}_{ij} \mathcal{G}_j^\flat\|_F$ can be recast into a Quadratic Programming (QP) problem. So far the parameters $\tilde{\delta}_{ij}$ have been considered fixed. However, if in Step (BI) one obtains $L_{ij} = \mathbf{0}_{n_i \times p_j}$ for some $j \in \mathcal{N}_i$, it is impossible to reduce the magnitude of the coupling term $\bar{A}_{ij}$ and, from (3.3), the knowledge of $y_{[j]}$ is useless. This suggests to revise the choice of $\tilde{\delta}_{ij}$ and set $\tilde{\delta}_{ij} = 0$. In step (BII), since $\mathbb{S}_i$ are polytopes, using procedures proposed in [KG98] we can compute scalars μ_{ij} as

$$\mu_{ij} = \max_{\psi\in 1:\bar{\psi}_i} \{\max_{s_{[j]}} g_{i,\psi} \bar{A}_{ij} s_{[j]} : \mathcal{G}_j s_{[j]} \leq \mathbf{1}_{\bar{\psi}_j}\}.$$

that requires the solution of $\bar{\psi}_i$ LP problems.

3.4.3 Part (C)

In Step (CI) we check the Schurness of matrix T. If the test fails, Assumption 3.5-(i) cannot be fulfilled and the only possibility is to restart the algorithm after increasing the number of variables $\tilde{\delta}_{ij}$ that are equal to one. In Step (CII), since the sets $\mathbb{S}_i$ and $\mathbb{E}_i$ are polytopes, using results from [KG98] the computation of the set Θ_0 can be done as follows

$$\begin{aligned} \Theta_0 &= \prod_{i\in\mathcal{M}} [0, \tilde{\theta}_i] \\ \tilde{\theta}_i &= (\max_{\tau\in 1:\bar{\tau}_i} \{\max_{s_{[i]}} h_{i,\tau} s_{[i]} : \mathcal{G}_i s_{[i]} \leq \mathbf{1}_{\bar{\psi}_i}\})^{-1}. \end{aligned} \tag{3.19}$$

Moreover, in Step (CII) we compute the equilibrium point $\bar{\theta}$ of system (3.12). If $\bar{\theta} \notin \Theta_0$ we can not guarantee property (3.7) and therefore the algorithm stops. Note that if $\mathbb{D}_i = \{\mathbf{0}_{r_i}\}$, $\forall i \in \mathcal{M}$, the equilibrium point $\bar{\theta}$ is the origin and hence $\bar{\theta} \in \Theta_0$ by construction.
According to Assumption 3.5-(iii) , the set Θ of all feasible contractions θ is computed as an RPI set for system (3.12) and constraints $\theta \in \Theta_0$. In particular, since T is Schur and Θ_0 is a polytope, using results from [GT91] we can compute the MRPI set Θ_∞ by solving a suitable LP problem.
As discussed in Remark 3.1, a decentralized initialization of state estimators is possible computing an hyper rectangle $\bar{\Theta}$ contained in Θ_∞. This is done in step (CIV). More precisely, using results from [BFT04], we can set $\bar{\Theta} = \prod_{i\in\mathcal{M}}[0, \bar{\theta}_i]$ where

$$\bar{\theta}_i = \max_{\tilde{\theta}\in\Theta_\infty} \gamma^T \tilde{\theta}, \tag{3.20}$$

$$\gamma = (\gamma_1, \ldots, \gamma_M)$$

$$\gamma_i = (\max_{\theta} \theta_i : \theta \in \Theta_\infty)^{-1}. \tag{3.21}$$

As described in [BFT04], the vector γ is used for maximizing the volume of $\bar{\Theta}$. From (3.20) and (3.21) the computation of the hyper-rectangle $\bar{\Theta}$ requires the solution of $M+1$ LP optimization problems.

3.5 Large-scale systems with variable number of subsystems

In this section, we discuss the retuning of the DSE when a subsystem is added or removed. We highlight that plug in and plug out of subsystems are here considered as offline operations. In particular, we will show how to preserve properties (3.6) and (3.7) without performing all computations required by Algorithm 3.1. As a starting point, we consider system (3.1) equipped with a DSE designed using Algorithm 3.1.

3.5.1 Plug-in operation

Assume the new subsystem $\Sigma_{[M+1]}$ is plugged in and set $\bar{\mathcal{M}} = \mathcal{M} \cup \{M+1\}$. Since the overall system has changed, in principle one has to design the DSE from scratch running Algorithm 3.1. Note however that Part (A) of Algorithm 3.1 is decentralized and therefore it has to be executed for the new subsystem only. Part (B) of Algorithm 3.1 involves only the new subsystem, its parents and its children $\mathcal{C}_{M+1} = \{j \in \mathcal{M} : A_{M+1,j} \neq 0, j \neq M+1\}$. In fact, subsystem $\Sigma_{[M+1]}$ needs sets $\mathbb{S}_j$ from its parents for computing parameters $\mu_{M+1,j}$, $j \in \mathcal{N}_{M+1}$. Moreover since children of $\Sigma_{[M+1]}$ have a new parent, they need to know $\mathbb{S}_{M+1}$ in order to update parameters $\mu_{k,M+1}$, $k \in \mathcal{C}_{M+1}$.
If Step (CI) or Step (CII) fail, we declare that system $\Sigma_{[M+1]}$ can not be added, because the family of sets $\mathcal{S}(\mathbb{S}, \Theta)$ is not a pRPI. In Algorithm 3.2 we summarize the computations for updating the DSE that are triggered by the addition of $\Sigma_{[M+1]}$.

3.5.2 Unplugging operation

Assume subsystem $\Sigma_{[q]}$, $q \in \mathcal{M}$ is removed. We will show that no update of the DSE is required in order to guarantee (3.6) and (3.7). In the following, vectors, matrices and sets with a hat are quantities of the DSE after subsystem q has been removed. As an example, the matrix

$$\hat{T} = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1,q-1} & \mu_{1,q+1} & \cdots & \mu_{1,M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{q-1,1} & \cdots & \mu_{q-1,q-1} & \mu_{q-1,q+1} & \cdots & \mu_{q-1,M} \\ \mu_{q+1,1} & \cdots & \mu_{q+1,q-1} & \mu_{q+1,q+1} & \cdots & \mu_{q+1,M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{M,1} & \cdots & \mu_{M,q-1} & \mu_{M,q+1} & \cdots & \mu_{M,M} \end{bmatrix} \in \mathbb{R}^{M-1 \times M-1}$$

is obtained from matrix T, by eliminating the q-th row and column. Next, we show Assumptions 3.5-(i), 3.5-(ii) and 3.5-(iii) are still verified after the removal of $\Sigma_{[q]}$.
Let $\mathcal{G} = (V, \mathcal{E})$ be the coupling graph of (3.1), i.e. a directed graph where vertices in $V = 1 : M$ are associated to subsystems and $(i,j) \in \mathcal{E} \Leftrightarrow i \in \mathcal{N}_j$.

Algorithm 3.2 Update of pRPI $\mathcal{S}(\mathbb{S}, \Theta)$ after a plug-in operation

Input: new subsystem $\Sigma_{[M+1]}$ with sets $\mathbb{E}_{M+1}$ and $\mathbb{D}_{M+1}$.
Output: an updated pRPI family of sets $\mathcal{S}(\mathbb{S}, \Theta)$.

(A) *Decentralized steps*
For $i = M + 1$ execute Steps (AI)-(AIII) of Algorithm 3.1;

(B) *Distributed steps*

- For subsystem $\Sigma_{[M+1]}$, if $\tilde{\delta}_{M+1,j} = 1$, compute the matrix $L_{M+1,j}$, $\forall j \in \mathcal{N}_{M+1}$ solving $\min_{L_{M+1,j}} ||\mathcal{G}_{M+1}\bar{A}_{M+1,j}\mathcal{G}_j||_p$, p is a generic norm, and then compute $\mu_{M+1,j}$;
- For subsystems $\Sigma_{[k]}$, if $\tilde{\delta}_{k,M+1} = 1$, compute the matrix $L_{k,M+1}$, $\forall k \in \mathcal{C}_{M+1}$ solving $\min_{L_{k,M+1}} ||\mathcal{G}_k\bar{A}_{k,M+1}\mathcal{G}_{M+1}||_p$, p is a generic norm, and then compute $\mu_{k,M+1}$;

(C) *Centralized steps*
Execute Steps (CI)-(CIV) of Algorithm 3.1.

In the sequel we assume $\mathcal{G}$ is strongly connected (see Definition 3.3 in Appendix 3.8.1). Indeed, if this is not true, then (3.1) can be represented as a directed acyclic graph $\mathbb{G}$ whose nodes are strongly connected subgraphs. In this case, a DSE can be designed for each system corresponding to a subgraph starting from the roots of $\mathbb{G}$. The next proposition concerns the Assumption 3.5-(i).

Proposition 3.1. *If the matrix $T \in \mathbb{R}^{M\times M}$ in (3.12) is Schur, then also the matrix $\hat{T}$ is Schur.*

Proof. The proof of Proposition 3.1 can be found in Appendix 3.8.1. □

The next result guarantees Assumption 3.5-(ii) still holds after the removal of subsystem q.

Proposition 3.2. *For $q \in \mathcal{M}$, let $\hat{\theta} = (\theta_1, \dots, \theta_{q-1}, \theta_{q+1}, \dots, \theta_{M-1})$, $\hat{\alpha} = (\alpha_1, \dots, \alpha_{q-1}, \alpha_{q+1}, \dots, \alpha_{M-1})$ and*

$$\hat{\Theta}_0 = \{\xi \in \mathbb{R}^{M-1} : (\xi_1, \dots, \xi_{q-1}, 0, \xi_q, \dots, \xi_{M-1}) \in \Theta_0\}$$

If Assumption 3.5-(ii) holds, the unique equilibrium $\hat{\bar{\theta}}$ *of system*

$$\hat{\theta}^{+} = \hat{T}\hat{\theta} + \hat{\alpha} \tag{3.22}$$

is such that $\hat{\bar{\theta}} \in \hat{\Theta}_0$*.*

Proof. The proof of Proposition 3.2 can be found in Appendix 3.8.2. □

Finally, the following proposition concerns Assumption 3.5-(iii).

Proposition 3.3. *For* $q \in \mathcal{M}$*, the set*

$$\hat{\Theta} = \{\hat{\theta} \in \mathbb{R}^{M-1} : (\hat{\theta}_1, \ldots, \hat{\theta}_{q-1}, 0, \hat{\theta}_q, \ldots, \hat{\theta}_{M-1}) \in \Theta_\infty\} \tag{3.23}$$

is an RPI set for system (3.22).

Proof. The proof of Proposition 3.3 can be found in Appendix 3.8.3. □

From Proposition 3.3 we have that the projection of set Θ on the coordinates $\hat{\theta}$ is still an RPI set for (3.1) after the removal of subsystem q, but we also note that the set $\hat{\Theta}$ is not the MRPI, i.e. with a new execution of Step (CIII) of Algorithm 3.1 we could obtain $\hat{\Theta} \subseteq \hat{\Theta}_\infty$. We also note that the projection $\hat{\bar{\Theta}}$ of $\bar{\Theta}$ on the coordinates $\hat{\theta}$ is a box verifying $\hat{\bar{\Theta}} \subseteq \hat{\Theta}$. However, with a new execution of Step (CIV) of Algorithm 3.1 we could obtain a bigger inner box approximation.

3.6 Examples

In this section, we apply the proposed distributed state estimator to the PNS proposed in Scenario 1 in Section B.1.1 of Appendix B. We rewrite the dynamics (B.1) of each area as

$$\Sigma^C_{[i]} : \quad \dot{x}_{[i]} = A_{ii}x_{[i]} + \bar{B}_i\bar{u}_{[i]} + \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]} + d_{[i]} \tag{3.24}$$

where $x_{[i]} = (\Delta\theta_i,\ \Delta\omega_i,\ \Delta P_{m_i},\ \Delta P_{v_i})$ is the state, $\bar{u}_{[i]} = (\Delta P_{ref_i}, \Delta P_{L_i})$ is composed by the control input of each area. In (3.24), $d_{[i]} \in \mathbb{R}^{n_i}$ is the disturbance term for the i-th area and it is bounded in the polytopic set $\mathbb{D}_i \subset \mathbb{R}^{n_i}$. For the simulations, we use the load power steps given in Section B.1.1 of Appendix B and the control inputs computed using MPC controllers as in Section B.3 of Appendix B. In Example 1 and 2, for each area, we consider the following bounds on the state estimation error

$$\mathbb{E}_i = \{e_{[i]} \in \mathbb{R}^{n_i} : ||e_{[i,1]}||_\infty \leq 0.005,\ ||e_{[i,k]}||_\infty \leq 0.01,\ k \in 2:4\}. \tag{3.25}$$

We highlight that constraints (3.25) correspond in tolerating state estimation errors less then 10% of the maximum value assumed by the state variables. In Example 3, we consider constraints on the error equal to $2\mathbb{E}_i$, $\forall i \in \mathcal{M}$. All simulations have been performed using the *PnPMPC-toolbox* for MatLab [RBFT12] (see also Appendix C).

3.6.1 Example 1

As first example, we consider $\tilde{\delta}_{ij} = 1$, $\forall i \in \mathcal{M}$, $\forall j \in \mathcal{N}_i$, $\mathbb{D}_i = \{\mathbf{0}_{r_i}\}$, $\forall i \in \mathcal{M}$ (i.e. no disturbances act on the system) and assume to measure only the angular speed deviation $\Delta\omega_{[i]}$ of each area. Therefore, outputs of subsystem i are given by

$$y_{[i]} = C_i x_{[i]}, \qquad C_i = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}. \tag{3.26}$$

In this case, Algorithm 3.1 stops in Step (CI) because the computed sets $\mathbb{S}_i$ are such that T is not Schur. We highlight that from the results of Step (BI), one obtains the same results if parameters $\tilde{\delta}_{ij}$ are all set equal to zero. Indeed, for matrices C_i in (3.26), it is impossible to reduce the magnitude of the coupling terms $\bar{A}_{ij} = A_{ij} + L_{ij}C_j$ by solving the optimization problems (3.16).

3.6.2 Example 2

We consider $\mathbb{D}_i = \{\mathbf{0}_{r_i}\}$, $\forall i \in \mathcal{M}$, i.e. no disturbances act on the system, and we assume to measure both $\Delta\theta_{[i]}$ and $\Delta\omega_{[i]}$ of each area. Therefore the outputs are given by

$$y_{[i]} = C_i x_{[i]}, \qquad C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{3.27}$$

First we consider $\tilde{\delta}_{ij} = 0$, $\forall i \in \mathcal{M}$, $\forall j \in \mathcal{N}_i$. In this case, as in the first example, since we cannot take advantage of the knowledge of parents' outputs, Algorithm 3.1 stops before its conclusion. Indeed, it is impossible to find sets $\mathbb{S}_i$ such that T is Schur. This example shows that if we also consider more output variables for each subsystem, Algorithm 3.1 can stop in Step (CI) due the magnitude of the coupling terms A_{ij}. Now we consider $\tilde{\delta}_{ij} = 1$, $\forall i \in \mathcal{M}$, $\forall j \in \mathcal{N}_i$. In this case we can reduce the magnitude of the coupling terms. Solving optimization problems (3.16), we can compute matrices L_{ij} such that $\bar{A}_{ij} = \mathbf{0}_{n_i \times n_j}$, hence the Schurness of matrix T is guaranteed since sets $\mathbb{S}_i$ are λ_i-contractive. In this case, $T =$

diag(0.932, 0.843, 0.711, 0.889) and $\bar{\Theta} = \{\theta \in \mathbb{R}^4 :\ 0 \leq \theta_i \leq 1,\ \forall i = 1 : 4\}$. We note that if matrix T is diagonal, Step (CIV) of Algorithm 3.1 can be skipped since $\Theta_\infty = \bar{\Theta}$.

We performed an estimation experiment initializing the local state estimators $\tilde{\Sigma}_{[i]}$, $i \in \mathcal{M}$ with $\tilde{x}_{[i]}(0) = x_{[i]}(0) - e_{[i]}(0)$, where $e_{[i]}(0)$ is a vertex of the set $\mathbb{S}_i$. In Figure 3.1 we show the maximum state estimation error defined as

$$\tilde{e}_{[j]}(t) = \max_{i \in \mathcal{M}} |x_{[i,j]}(t) - \tilde{x}_{[i,j]}(t)| \tag{3.28}$$

where $x_{[i,j]}$ and $\tilde{x}_{[i,j]}$ are, respectively, the real and estimated state trajectory of the j-th state of the i-th subsystem. From Figure 3.1 we note that,

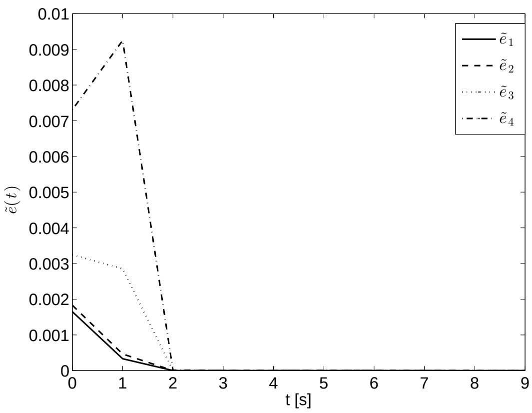


Figure 3.1: Maximum estimation errors $\tilde{e}_{[j]}$ defined as in (3.28), for Example 2.

since no disturbances act on the system, the state estimation error $e_{[i]}(t)$ converges to zero as $t \to \infty$, i.e. (3.6) is verified.

3.6.3 Example 3

We consider $\mathbb{D}_i = \{d_{[i]} \in \mathbb{R}^{n_i} :\ ||d_{[i]}||_\infty \leq 10^{-5}\}$, $\forall i \in \mathcal{M}$ and output variables given in (3.27). As in Example 2, by considering $\tilde{\delta}_{ij} = 1$, $\forall i \in \mathcal{M}$, $\forall j \in \mathcal{N}_i$, Algorithm 3.1 does not stop at any intermediate step. We have performed a similar experiment as in Example 2, but generating statistically independent random samples $d_{[i]}(t)$ from the uniform distribution on $\mathbb{D}_i$. In Figure 3.2 and 3.3, we show the maximum estimation error at the beginning of the experiment (Figure 3.2) and for $t \geq 10$ (Figure 3.3). In particular, even in presence of disturbances on the system the state estimation error $e_{[i]}(t)$ lies in the set $\mathbb{E}_i$, $\forall i \in \mathcal{M}$.

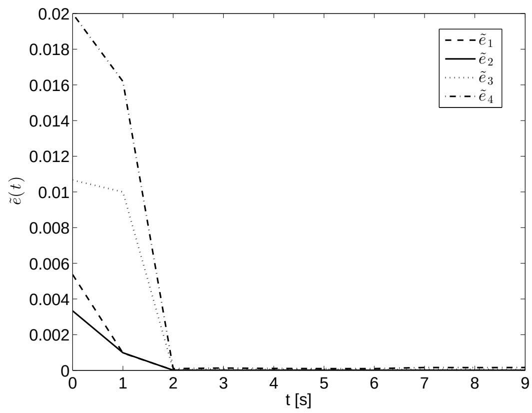


Figure 3.2: Maximum estimation error $\tilde{e}_{[j]}(t)$, $t = 0 : 9$ defined as in (3.28), for Example 3.

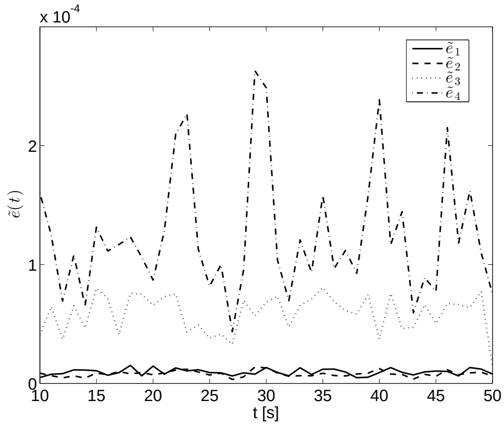


Figure 3.3: Maximum estimation error $\tilde{e}_{[j]}(t)$, $t = 10 : 50$ defined as in (3.28), for Example 3.

3.6.4 Comparison with [FS11a]

In the previous examples we considered polytopic sets $\mathbb{E}_i$, $i \in \mathcal{M}$ defined in (3.25) that are also zonotopes. Therefore we can compare the proposed DSE with the one proposed in [FS11a]. Local state estimators in [FS11a]

depend on the state of parent systems, but not on their outputs. This corresponds to setting $\tilde{\delta}_{ij} = 0$, $i, j \in \mathcal{M}$ in our scheme. Using the DSE in [FS11a], we cannot compute the observers for the PNS in Examples 1 and 2. In fact, since all parameters $\tilde{\delta}_{ij}$ are zero it is impossible to reduce the magnitude of the coupling by using parents' outputs, as we do in Example 2.
Moreover in [FS11a], the authors look for a family of sets $\mathbb{S}$ composed by mRPI sets $\mathbb{S}_i$ verifying

$$\bar{A}_{ii}\mathbb{S}_i \oplus \bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{S}_j \subseteq \mathbb{S}_i \subseteq \mathbb{E}_i. \tag{3.29}$$

We note that (3.29) is a special case of the pRPI family in (3.9) when $\tilde{\delta}_{ij} = 0$ (i.e. $\bar{A}_{ij} = A_{ij}$) and $\mathbb{D}_i = \{\mathbf{0}_{r_i}\}$. Since from (3.29) one has $\theta^+ = \theta$, the matrix T is not Schur, and for guaranteeing convergence of the estimates one has to check Schurness of the overall matrix $\mathbf{A}+\mathbf{LC}$. In our distributed estimator, convergence of the estimates can be checked by testing the Schurness of T, i.e. no assumptions on the overall matrices are needed. However, the price to pay is a loss in generality when in Step (BI) of Algorithm 3.1 one cannot reduce coupling terms using gains L_{ij}, i.e. one obtains $\bar{A}_{ij} = A_{ij}$. Indeed, in this case, $\mathbf{A}+\mathbf{LC}$ can be Schur even if matrix T is not and therefore Algorithm 3.1 stops in Step (CII).

3.7 Final comments

In this chapter, we proposed a novel partition-based state estimator for linear discrete-time subsystems affected by bounded disturbances. The proposed DSE guarantees the convergence of the overall state estimation and also boundedness on the state estimation error. Since our design method involves some degree of conservatism, we provided an algorithm for the choice of local estimator parameters. Moreover, the most burdensome steps of the design procedure involve decentralized computations. Similarly to [FS11a] our state estimation algorithm can be directly used together with the distributed model predictive control scheme proposed in Chapter 2 although further research is needed for assessing the stability properties of the closed-loop system. In Chapter 8, we will also consider the problem of decentralizing completely computations required in the design process. This would lead to state-estimators that can be designed using local computational resources only, so coping with the PnP design requirements of the model predictive control scheme proposed in Chapters 5 and 6.

3.8 Appendix

3.8.1 Proof of Proposition 3.1

The proof of Proposition 3.1 hinges on Perron-Frobenious theory for nonnegative matrices. Next, we provide relevant definition, deferring the reader to [Mey00] for further details.

Definition 3.2. The graph, $\Gamma(Q) = (V, \mathcal{E})$ of $Q \in \mathbb{R}^{M\times M}$ is the directed graph with nodes $V = 1 : M$ and edges $\mathcal{E} = \{(i,j) : q_{ij} \neq 0\}$ where q_{ij} is the ij-th element of the matrix Q.

Definition 3.3. A directed graph Γ is strongly connected if for any pair of nodes (N_i, N_j) there exists a sequence of edges which leads from N_i to N_j.

Definition 3.4. A matrix $Q \in \mathbb{R}^{M\times M}$ is irreducible if there is no permutation matrix P such that

$$Z = PQP^T = \begin{bmatrix} Q_{11} & Q_{12} \\ 0 & Q_{22} \end{bmatrix},$$

where $Q_{11} \in \mathbb{R}^{q\times q}$, $Q_{22} \in \mathbb{R}^{M-q,M-q}$ and $Q_{12} \in \mathbb{R}^{q,M-r}$, $0 < q < M$.

Proof of Proposition 3.1. The matrix T in (3.12) is nonnegative, i.e. $\mu_{ij} \geq 0,\ \forall\ i,j \in \mathcal{M}$. Moreover, $\mathcal{G} = \Gamma(T)$ and since $\mathcal{G}$ is strongly connected, T is irreducible [Mey00, p. 671]. Let

$$\mathcal{T} = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1,q-1} & 0 & \mu_{1,q+1} & \cdots & \mu_{1,M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{q-1,1} & \cdots & \mu_{q-1,q-1} & 0 & \mu_{q-1,q+1} & \cdots & \mu_{q-1,M} \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \mu_{q+1,1} & \cdots & \mu_{q+1,q-1} & 0 & \mu_{q+1,q+1} & \cdots & \mu_{q+1,M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{M,1} & \cdots & \mu_{M,q-1} & 0 & \mu_{M,q+1} & \cdots & \mu_{M,M} \end{bmatrix} \in \mathbb{R}^{M\times M} \quad (3.30)$$

From Wielandt's Theorem [Mey00, p. 675], one has $\bar{\rho}(\mathcal{T}) \leq \bar{\rho}(T)$. Moreover, up to a permutation matrix, one has $\mathcal{T} = \begin{bmatrix} 0 & 0 \\ 0 & \hat{T} \end{bmatrix}$ and hence $\bar{\rho}(\hat{T}) \leq \bar{\rho}(\mathcal{T})$. Therefore $\bar{\rho}(\hat{T}) \leq \bar{\rho}(T)$ and the proof is concluded recalling that, by assumption, $\bar{\rho}(T) < 1$. □

3.8.2 Proof of Proposition 3.2

Proof. First we define matrix $\mathcal{T}$ as in (3.30). Since $\mu_{ij} \geq 0$, $\forall i, j \in \mathcal{M}$, the elements of matrices $\mathcal{T}^k$ and T^k are nonnegative $\forall k \geq 0$. Moreover we can show that the ij-th element of $\mathcal{T}^k$ (with abuse of notation, $\mathcal{T}^k_{ij}$) is smaller than the ij-th element of T^k (with abuse of notation, T^k_{ij}), i.e. $\mathcal{T}^k_{ij} \leq T^k_{ij}$, $\forall i, j \in \mathcal{M}$ and $\forall k \geq 0$. Let $\bar{\tau} = (\hat{\bar{\theta}}_1, \ldots, \hat{\bar{\theta}}_{q-1}, 0, \hat{\bar{\theta}}_q, \ldots, \hat{\bar{\theta}}_{M-1}) \in \mathbb{R}^M$, where $\hat{\bar{\theta}}_i$ is the i-th component of vector $\hat{\bar{\theta}}$ and

$$\tilde{\alpha} = \alpha \in \mathbb{R}^M_+, \ \tilde{\alpha}_q = 0. \tag{3.31}$$

The unique equilibrium point of system (3.12) can be written as $\bar{\theta} = \sum_{k=0}^{\infty} T^k \alpha$. Moreover from (3.22) and the definitions of $\bar{\tau}$ and $\mathcal{T}$, we have that $\bar{\tau} = \sum_{k=0}^{\infty} \mathcal{T}^k \tilde{\alpha}$. Since $\mathcal{T}^k_{ij} \leq T^k_{ij}$, then $\sum_{k=0}^{\infty} \mathcal{T}^k_{ij} \leq \sum_{k=0}^{\infty} T^k_{ij}$ and hence $\bar{\tau} \leq \bar{\theta}$ element-wise. From (3.19) and Assumption 3.5-(ii), one has $\bar{\tau} \in \Theta_0$. Therefore, we can conclude that $\hat{\bar{\theta}} \in \hat{\Theta}_0$. □

3.8.3 Proof of Proposition 3.3

Proof. After subsystem q has been removed, the dynamics of contraction factors $\hat{\theta}$ is given by (3.22). In the following we show that $\hat{\Theta}$ defined in (3.23) is an RPI set for (3.22). From the invariance of set Θ_∞ we have that $T\theta + \alpha \in \Theta_\infty$, $\forall \theta \in \Theta_\infty$. Moreover, since $0 \in \Theta_\infty$, we have $\mathcal{T}\theta + \tilde{\alpha} \in \Theta_\infty$, $\forall \theta \in \Theta_\infty$ (where $\mathcal{T}$ and $\tilde{\alpha}$ are defined in (3.30) and (3.31)) i.e. the set Θ_∞ is also invariant for the LTI system $\theta^+ = \mathcal{T}\theta + \tilde{\alpha}$ and the q-th component of θ is always zero. Therefore, we can conclude that the projection of set Θ_∞ defined in (3.23) is an RPI set for system $\hat{\theta}^+ = \hat{T}\hat{\theta} + \hat{\alpha}$. □

Part II

Plug-and-play design of distributed controllers and observers

Chapter 4

Plug-and-play control for unconstrained linear systems

Contents

4.1 Introduction

In this chapter we propose a distributed controller for discrete-time LTI systems guaranteeing asymptotic stability of the closed-loop system when subsystems are plugged in and out. We focus our attention on unconstrained systems and introduce basic mathematical tools that will be used in the following chapters in order to prove stability and constraint satisfaction for constrained systems. In Section 4.2.1 we propose a static state-feedback controller and give results for PnP design. Differently from centralized control architectures, in the decentralized case, stabilizing dynamic state-feedback controllers may exist even if stability cannot be achieved using static state-feedback regulators. In Section 4.2.2, relying on a classical result in centralized control [BP70], we show that we can always rewrite the problem of designing PnP dynamic state-feedback controllers as the problem of designing PnP static state-feedback controllers. In Section 4.3 we describe the operations needed when subsystems are plugged in and out. In Section 4.4 we propose an example and Section 4.5 is devoted to concluding remarks.

4.2 Distributed control for unconstrained LTI systems

We consider a large-scale discrete-time LTI system

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.1}$$

composed of M subsystems, as described in Section 1.5. In this chapter we assume that each subsystem $\Sigma_{[i]}$, $\forall i \in \mathcal{M}$ is unconstrained.

4.2.1 Distributed static state-feedback controllers

Next, we propose a distributed controller for (4.1) guaranteeing asymptotic stability of the closed-loop system when subsystems are plugged in and out. We achieve decentralization of controller design by treating coupling terms $w_{[i]} = A_{ij}x_{[j]}$, $j \in \mathcal{N}_i$ in (1.3) as disturbances for subsystem $\Sigma_{[i]}$. However, we also decrease the coupling strength by using a distributed control architecture where a local controller receives state measurements from parent subsystems. The local controller $\mathcal{C}_{[i]}$ for (1.3) is given by

$$\mathcal{C}_{[i]}: \quad u_{[i]} = K_{ii}x_{[i]} + \sum_{j\in\mathcal{N}_i} \delta_{ij}K_{ij}x_{[j]}. \tag{4.2}$$

where $K_{ij} \in \mathbb{R}^{m_i \times n_j}$ and $\delta_{ij} \in \{0, 1\}$, $i, j \in \mathcal{M}$. Note that, if $\delta_{ij} = 0$, $\forall i \in \mathcal{M}$,$\forall j \in \mathcal{N}_i$, the control scheme is completely decentralized, since each input $u_{[i]}$ depends upon states of system $\Sigma_{[i]}$ only. Next, we clarify properties of matrices K_{ii}, $i \in \mathcal{M}$ that are required for the stability of system (1.3) controlled by (4.2). The closed-loop dynamics of $\Sigma_{[i]}$ equipped with controller $\mathcal{C}_{[i]}$ is

$$x_{[i]}^{+} = F_{ii}x_{[i]} + \sum_{j\in\mathcal{N}_i} F_{ij}x_{[j]} \tag{4.3}$$

where $F_{ii} = A_{ii} + B_iK_{ii}$ and $F_{ij} = A_{ij} + \delta_{ij}B_iK_{ij}$. Therefore, if $\delta_{ij} \neq 0$, the distributed controller $\mathcal{C}_{[i]}$ allow us to modify the open-loop coupling term A_{ij}. From (4.3), one obtains the collective closed-loop model

$$\mathbf{x}^{+} = \mathbf{F}\mathbf{x} \tag{4.4}$$

where $\mathbf{F}$ is composed by blocks F_{ij}, $i, j \in \mathcal{M}$.
Our aim is to design, in a decentralized fashion, controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ such that system (4.4) is asymptotically stable. The design procedure will exploit the following assumption.

Assumption 4.1. *The matrices $F_{ii} = A_{ii} + B_iK_{ii}$, $i \in \mathcal{M}$ are Schur.*

The next Proposition provides the main result on stability of (4.4).

Proposition 4.1. *For given matrices K_{ii}, $i \in \mathcal{M}$ verifying Assumption 4.1, matrices K_{ij} and parameters $\delta_{ij} \in \{0,1\}$, $j \in \mathcal{N}_i$, if the following conditions are fulfilled*

$$\alpha_i = \sum_{j\in\mathcal{N}_i}\sum_{k=0}^{\infty} ||F_{ii}^kF_{ij}||_\infty < 1, \;\; \forall i \in \mathcal{M} \tag{4.5}$$

then, the closed-loop system (4.4) is asymptotically stable.

Proof. The proof is given in Appendix 4.6.1. □

We highlight that condition (4.5) is based on the small-gain theorem for networks of systems [DRW07] here used for capturing the propagation of coupling terms. Moreover, for a given $i \in \mathcal{M}$, the quantities α_i in (4.5) depend only on local tunable parameters $\{K_{ii}, \{K_{ij}, \delta_{ij}\}_{j\in\mathcal{N}_i}\}$ but not on parents' tunable parameters. Hence, controllers $\mathcal{C}_{[i]}$ can be designed in parallel solving the following independent problems for $i \in \mathcal{M}$.

Problem 4.1 ($\mathcal{P}_i$)**.** Check if there exist K_{ii}, K_{ij} and δ_{ij}, $j \in \mathcal{N}_i$ such that $\alpha_i < 1$.

Algorithm 4.1 Design of distributed controllers $\mathcal{C}_{[i]}$

Input: A_{ii}, B_i, $\mathcal{N}_i$, $\{\delta_{ij}\}_{j\in\mathcal{N}_i}$, $\{A_{ij}\}_{j\in\mathcal{N}_i}$.
Output: controller $\mathcal{C}_{[i]}$ as in (4.2).

(I) $\forall j \in \mathcal{N}_i$, if $\delta_{ij} = 1$, compute the matrix K_{ij}, solving

$$\min_{K_{ij}} ||F_{ij}||_p \tag{4.6}$$

where p is a generic norm.

(II) Compute a matrix K_{ii} such that Assumption 4.1 is fulfilled and $\alpha_i < 1$. If it does not exist, then **stop** (the controller $\mathcal{C}_{[i]}$ cannot be designed).

The procedure for solving problems $\mathcal{P}_i$, $i \in \mathcal{M}$ is summarized in Algorithm 4.1.
In order to discuss Step (I) of Algorithm 4.1, assuming $p = \infty$, we upper bound α_i in (4.5) as

$$\alpha_i = \sum_{j\in\mathcal{N}_i}\sum_{k=0}^{\infty} ||F_{ii}^k F_{ij}||_\infty \leq \sum_{k=0}^{\infty} ||F_{ii}||_\infty^k \sum_{j\in\mathcal{N}_i} ||F_{ij}||_\infty.$$

Hence, computing K_{ij} as in (4.6) minimizes the upper bound. Moreover, setting $p = \infty$ makes (4.6) is an LP problem. So far, the parameters δ_{ij} have been considered fixed. However, if in Step (I) one obtains $K_{ij} = \mathbf{0}_{m_i\times n_j}$ for some $j \in \mathcal{N}_i$, it is impossible to reduce the magnitude of the coupling term F_{ij} and the knowledge of $x_{[j]}$ is useless for controller $\mathcal{C}_{[i]}$. This suggests to revise the choice of δ_{ij} and set $\delta_{ij} = 0$. Next, we propose automatic methods for computing the matrix K_i in Step (II) of Algorithm 4.1.

- The first method hinges on the solution of the following nonlinear optimization problem

$$\min_{K_i} \mu_i \tag{4.7a}$$
$$\bar{\rho}(A_{ii} + B_i K_i) < 1 \tag{4.7b}$$
$$\alpha_i < 1 \tag{4.7c}$$

where $\mu_i = \max(\alpha_i, \bar{\rho}(A_{ii} + B_i K_i))$. Since (4.7) is a nonlinear optimization problem, suitable initialization of K_i is needed, therefore we initialize K_i as the Linear Quadratic Regulator (LQR) gain associated to matrices $\hat{Q}_i \geq \mathbf{0}_{n_i \times n_i}$ and $\hat{R}_i > \mathbf{0}_{m_i \times m_i}$, i.e.

$$K_i = (\hat{R}_i + B_i^T \bar{P}_i B_i)^{-1} B_i^T \bar{P}_i A_{ii} \tag{4.8}$$

where $\bar{P}_i$ is the solution to the algebraic Riccati equation

$$A_{ii}^T \bar{P}_i A_{ii} + \hat{Q}_i - A_{ii}^T \bar{P}_i B_i (\hat{R}_i + B_i^T \bar{P}_i B_i)^{-1} B_i^T \bar{P}_i A_{ii} = \bar{P}_i. \tag{4.9}$$

We highlight that in several case studies this procedure for initializing the optimization algorithm has proved very effective and it has been implemented in the *PnPMPC-toolbox* for MatLab [RBFT12] (see Appendix C). However, we highlight that the optimized matrix K_i from (4.7) could not be an LQR. For this reason we also propose a second automatic method that allows one to optimize an LQR.

- We design K_i as the LQR gain associated to matrices $\hat{Q}_i \geq \mathbf{0}_{n_i \times n_i}$ and $\hat{R}_i > \mathbf{0}_{m_i \times m_i}$ (see formula (4.8)) solving the following nonlinear optimization problem

$$\min_{\hat{Q}_i,\ \hat{R}_i} \ \alpha_i \tag{4.10a}$$

$$\hat{Q}_i \geq \mathbf{0}_{n_i \times n_i},\ \hat{R}_i > \mathbf{0}_{m_i \times m_i} \tag{4.10b}$$

$$\alpha_i < 1 \tag{4.10c}$$

$$\text{constraints (4.8) and (4.9).} \tag{4.10d}$$

A few remarks on the computations required for solving (4.10) are in order. The series in (4.5) involve only positive terms and can be easily truncated if either (4.5) is violated or summands fall below the machine precision. In order to simplify the optimization problem (4.10) one can assume $\hat{Q}_i = \text{diag}(\hat{q}_{i,1}, \ldots, \hat{q}_{i,n_i})$, $\hat{R}_i = \text{diag}(\hat{r}_{i,1}, \ldots, \hat{r}_{i,m_i})$ and replace the matrix inequalities in (4.10b) with the scalar inequalities $\hat{q}_{i,k} \geq 0$, $k \in 1 : n_i$ and $\hat{r}_{i,k} > 0$, $k \in 1 : m_i$.

The feasibility of optimization problem (4.7) or (4.10) guarantees that the controller $\mathcal{C}_{[i]}$ can be successfully designed.

4.2.2 Distributed dynamic state-feedback controllers

In this section we generalize the PnP design proposed in Section 4.2. So far, we considered static state-feedback controllers $\mathcal{C}_{[i]}$. Next, we propose the following dynamic state-feedback controller

$$\hat{\mathcal{C}}_{[i]}: \begin{array}{l} \hat{x}_{[i]}^{+} = S_i\hat{x}_{[i]} + G_{ii}x_{[i]} + \sum_{j\in\mathcal{N}_i} \hat{\delta}_{ij}G_{ij}x_{[j]} \\ u_{[i]} = H_i\hat{x}_{[i]} + K_{ii}x_{[i]} + \sum_{j\in\mathcal{N}_i} \delta_{ij}K_{ij}x_{[j]} \end{array} \tag{4.11}$$

where $\hat{x}_{[i]} \in \mathbb{R}^{\hat{n}_i}$, $S_i \in \mathbb{R}^{\hat{n}_i\times\hat{n}_i}$, $G_{ij} \in \mathbb{R}^{\hat{n}_i\times n_j}$, $H_i \in \mathbb{R}^{m_i\times\hat{n}_i}$ and $\hat{\delta}_{ij} \in \{0,1\}$, $\forall i,j \in \mathcal{M}$. We highlight that if one assumes $\hat{n}_i = 0$, controller (4.11) can be written as in (4.2). From (1.2) and (4.11), we obtain the closed-loop subsystem

$$\begin{bmatrix} x_{[i]} \\ \hat{x}_{[i]} \end{bmatrix}^{+} = \begin{bmatrix} A_{ii}+B_iK_{ii} & B_iH_i \\ G_{ii} & S_i \end{bmatrix} \begin{bmatrix} x_{[i]} \\ \hat{x}_{[i]} \end{bmatrix} + \sum_{j\in\mathcal{N}_i} \begin{bmatrix} A_{ij}+\delta_{ij}B_iK_{ij} & \mathbf{0}_{n_i\times n_j} \\ \hat{\delta}_{ij}G_{ij} & \mathbf{0}_{\hat{n}_i\times\hat{n}_j} \end{bmatrix} \begin{bmatrix} x_{[j]} \\ \hat{x}_{[j]} \end{bmatrix} \tag{4.12}$$

and hence, by defining $\hat{\mathbf{x}} = (\hat{x}_{[1]},\dots,\hat{x}_{[M]}) \in \mathbb{R}^{\hat{n}}$, $\hat{n} = \sum_{i\in\mathcal{M}} \hat{n}_i$, we obtain the closed-loop collective system as

$$\begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix}^{+} = \begin{bmatrix} \mathbf{F} & \mathbf{BH} \\ \mathbf{G} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix} \tag{4.13}$$

where $\mathbf{H}$, $\mathbf{G}$ and $\mathbf{S}$ are composed by blocks H_i, G_{ii}, $\hat{\delta}_{ij}G_{ij}$ and S_i, $\forall i \in \mathcal{M}$, $\forall j \in \mathcal{N}_i$. As in the previous section, our aim is to design in a decentralized fashion controllers $\mathcal{C}_{[i]}$, $\in \mathcal{M}$ such that the closed-loop system (4.13) is asymptotically stable.

Assumption 4.2. *The closed-loop decoupled subsystems are asymptotically stable, i.e. matrices* $\begin{bmatrix} A_{ii}+B_iK_{ii} & B_iH_i \\ G_{ii} & S_i \end{bmatrix}$ *are Schur.*

Proposition 4.2. *For given matrices K_{ii}, H_i, S_i and G_{ii}, $i \in \mathcal{M}$, verifying Assumption 4.2 and matrices K_{ij}, G_{ij} and parameters δ_{ij}, $\hat{\delta}_{ij} \in \{0,1\}$, $j \in \mathcal{N}_i$, if the following conditions are fulfilled for all $i \in \mathcal{M}$,*

$$\hat{\alpha}_i = \sum_{j\in\mathcal{N}_i} \sum_{k=0}^{\infty} || \begin{bmatrix} A_{ii}+B_iK_{ii} & B_iH_i \\ G_{ii} & S_i \end{bmatrix}^k \begin{bmatrix} A_{ij}+\delta_{ij}B_iK_{ij} & \boldsymbol{0}_{n_i\times n_j} \\ \hat{\delta}_{ij}G_{ij} & \boldsymbol{0}_{\hat{n}_i\times\hat{n}_j} \end{bmatrix} ||_\infty < 1,$$

then, the closed-loop system (4.13) is asymptotically stable.

Proof. The proof is given in Appendix 4.6.2. □

Proposition 4.2 is based on the well-know result that a dynamic state-feedback controller for an LTI system can be viewed as a static state-feedback for an augmented system comprising $\hat{n}_i$ integrators [BP70]. Therefore, from Proposition 4.2, we can design controllers $\hat{\mathcal{C}}_{[i]}$ using a procedure at all similar to Algorithm 4.1.

4.3 Plug-and-play operations

Consider a system composed by subsystems $\Sigma_{[i]}$, $i \in \mathcal{M}$ equipped with local controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ produced by Algorithm 4.1. In case subsystems are added or removed, existing controllers have to be redesigned. In this Section we propose a PnP distributed solution, which requires the redesign of a limited number of controllers. We assume subsystems get plugged in and out offline.

4.3.1 Plug-in operation

We start considering plug in of subsystem $\Sigma_{[M+1]}$, characterized by parameters $A_{M+1,M+1}$, B_{M+1}, $\mathbb{X}_{M+1}$, $\mathbb{U}_{M+1}$, $\mathcal{N}_{M+1}$ and $\{A_{M+1,j}\}_{j\in\mathcal{N}_{M+1}}$, into an existing plant. In particular $\mathcal{N}_{M+1}$ identifies the subsystems that will be physically coupled to $\Sigma_{[M+1]}$ and $\{A_{M+1,j}\}_{j\in\mathcal{N}_{M+1}}$ are the corresponding coupling terms. For designing the controller $\mathcal{C}_{[M+1]}$ we execute Algorithm 4.1 that needs information only from subsystems $\Sigma_{[j]}$, $j \in \mathcal{N}_{M+1}$. If Algorithm 4.1 stops before the last step we declare that $\Sigma_{[M+1]}$ cannot be plugged in. Let

$$\mathcal{S}_i = \{j : i \in \mathcal{N}_j\}$$

be the set of children of subsystem i. Since $\Sigma_{[M+1]}$ is now a parent of subsystems $\Sigma_{[j]}$, $j \in \mathcal{S}_{M+1}$, existing matrices K_j, $j \in \mathcal{S}_{M+1}$ may now result in $\alpha_j \geq 1$. Indeed, when $\mathcal{N}_j$ gets larger, the quantity α_j in (4.5) can only increase and therefore the condition in (4.5) could be violated. This means that for each $j \in \mathcal{S}_{M+1}$ the controllers $\mathcal{C}_{[j]}$ must be redesigned according to Algorithm 4.1. Again, if Algorithm 4.1 stops before completion for some $j \in \mathcal{S}_{M+1}$, we declare that $\Sigma_{[M+1]}$ cannot be plugged in.
In conclusion, the addition of system $\Sigma_{[M+1]}$ triggers the design of controller $\mathcal{C}_{[M+1]}$ and the redesign of controllers $\mathcal{C}_{[j]}$, $j \in \mathcal{S}_{M+1}$ according to Algorithm 4.1. Note that controller redesign does not propagate further in the network, i.e. even without changing controllers $\mathcal{C}_{[i]}$, $i \notin \{M+1\} \bigcup \mathcal{S}_{M+1}$ stability of the origin and constraint satisfaction are guaranteed for the new closed-loop system.

4.3.2 Unplugging operation

We consider plug out of subsystem $\Sigma_{[k]}$, $k \in \mathcal{M}$. Since for each $i \in \mathcal{S}_k$ the set $\mathcal{N}_i$ gets smaller, we have that α_i in (4.5) cannot increase. This means that for each $i \in \mathcal{S}_k$ the controller $\mathcal{C}_{[i]}$ does not have to be redesigned. Moreover, since for each system $\Sigma_{[j]}$, $j \notin \{k\} \bigcup \mathcal{S}_k$, the set $\mathcal{N}_j$ does not change, the redesign of controller $\mathcal{C}_{[j]}$ is not required.
In conclusion, the removal of system $\Sigma_{[k]}$ does not require the redesign of any controller, in order to guarantee stability of the origin and constraint satisfaction for the new closed-loop system.

4.3.3 Generalizations to parameter-dependent subsystems

In many engineering applications, parameters of subsystem i are influenced by parent subsystems. We model this scenario replacing (1.2) with

$$\Sigma^p_{[i]} : \quad x^+_{[i]} = A_{ii}(\Xi_i)x_{[i]} + B_i(\Xi_i)u_{[i]} + \sum_{j\in\mathcal{N}_i} A_{ij}(\Xi_i)x_{[j]}$$

where $\Xi_i = (\xi_{ii}, \{\xi_{ij}\}_{j\in\mathcal{N}_i})$ and $\xi_{ij} \in \mathbb{R}^{\upsilon_{ij}}$ are parameter vectors. We note that, for given sets $\mathcal{N}_i$, $i \in \mathcal{M}$, matrices A_{ii} and B_i are constant and design of PnP regulators can be still done using the methods described in Section 4.2.1. Furthermore, the procedure for plug in a new subsystem can be applied with no change since it requires the redesign of controllers $\mathcal{C}_{[j]}$, $j \in \mathcal{S}_{M+1}$, i.e. controllers associated to the subsystems $\Sigma^p_{[j]}$ for which matrices A_{jj} and B_j could change. However, when system $\Sigma^p_{[k]}$ gets plug out, it is now mandatory to retune all controllers $\mathcal{C}_{[j]}$, $j \in \mathcal{S}_k$ since changes in the matrices A_{jj} and B_j could hamper the fulfillment of conditions (4.5) when using the matrices K_{jj} computed prior to the subsystem removal. Moreover, if Algorithm 4.1 stops before completing the redesign of controllers $\mathcal{C}_{[j]}$, $\forall j \in \mathcal{S}_k$, we declare that subsystem $\Sigma^p_{[k]}$ cannot be plugged out. An example of parameter-dependent subsystems is the PNS described in Appendix B.

4.4 Example

We consider a LSS composed of M masses connected as in Figure 4.1. Our purpose is to show how coupling attenuation terms in Algorithm 4.1 can substantially ease the design of local controllers. Each mass $i \in \mathcal{M} = 1 : M$, is a subsystem with input $u_{[i]}$ and state variables $x_{[i]} = (x_{[i,1]}, x_{[i,2]})$, where $x_{[i,1]}$ is the displacement with respect to a given equilibrium position, $x_{[i,2]}$

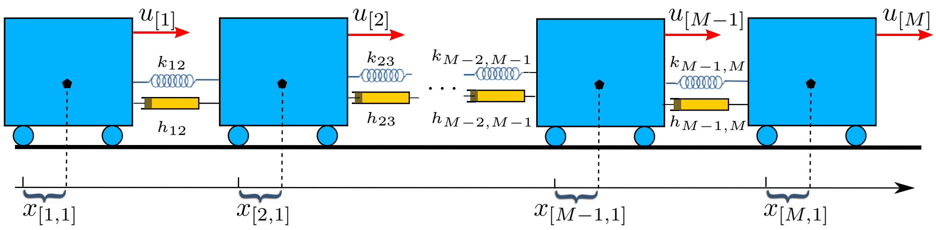


Figure 4.1: String of masses.

is the horizontal velocity and $100u_{[i]}$ is an external force in the horizontal direction. The values of m_i have been extracted randomly in the interval $[5, 10]$ while spring and damping coefficients are identical and equal to 0.5. We obtain subsystems $\Sigma_{[i]}$ by discretizing continuous-time models with 0.2 sec sampling time, using zero-order-hold discretization for the local dynamics and treating $x_{[j]}$, $j \in \mathcal{N}_i$ as exogenous signals [FCS13]. Controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ have been first designed using Algorithm 4.1 with $\delta_{ij} = 0$, $\forall j \in \mathcal{N}_i$. All subsystems fulfill condition (4.5), and therefore Algorithm 4.1 allows us to design, in a decentralized fashion, a static state-feedback decentralized controller. However, as the value of masses m_i, $i \in \mathcal{M}$ decreases, coupling terms increase, and, at the same point, it becomes impossible to design decentralized controllers $\mathcal{C}_{[i]}$. For example, if all masses m_i are in the interval $[0.01, 0.02]$, we cannot fulfill conditions (4.5). However, by setting $\delta_{ij} = 1$, $j \in \mathcal{N}_i$ we can completely remove the coupling terms and therefore the synthesis of controllers $\mathcal{C}_{[i]}$ amounts to the synthesis of a state-feedback controller for each mass without coupling. Moreover, plug in of the new mass $N + 1$ does not require the retuning of matrices K_{jj}, $j \in \mathcal{S}_{N+1}$ because the coupling terms F_{ij} are zeroed by computing $K_{j\ N+1}$ as in (4.6). This means that, for the system in Figure 4.1, PnP design of distributed controllers $\mathcal{C}_{[i]}$ is always possible.

4.5 Final comments

In this chapter we introduced the basic mathematical tools needed in the following chapters to design PnP model predictive controllers and PnP state estimators. We consider unconstrained systems and local state-feedback controllers that can be either static or dynamic. Moreover we also noted that if a subsystem depends on a set of parameters, we can still apply PnP design. This observation plays a crucial role for LSSs as power networks in which the model of each area depends on parameters of parent subsystems (see Appendix B). In the next chapters, we propose a PnP design for constrained systems.

4.6 Appendix

4.6.1 Proof of Proposition 4.1

Proof. Define a matrix $\mathbb{M}$ such that its ij-th entry μ_{ij} is

$$\begin{array}{ll} \mu_{ij} = -1 & \text{if} \quad i = j \\ \mu_{ij} = \sum_{k=0}^{\infty} \|F_{ii}^k F_{ij}\|_\infty & \text{if} \quad i \neq j. \end{array}$$

Note that all the off-diagonal entries of matrix $\mathbb{M}$ are non-negative, i.e., it is Metzler (see Section A.2 in Appendix A).
Inequalities (4.5) are equivalent to $\mathbb{M}\nu < \mathbf{0}_M$ where $\nu = \mathbf{1}_M$. Then, from Lemma A.1, $\mathbb{M}$ is Hurwitz. From Lemma A.2, (4.5) implies that matrix $\Gamma = \mathbb{M} + \mathbb{I}_M$ is Schur.
For subsystem $\Sigma_{[i]}$ in (1.2), where $u_{[i]}$ is defined as in (4.2), we have

$$x_{[i]}(t) = F_{ii}^t x_{[i]}(0) + \sum_{k=0}^{t-1} F_{ii}^k \sum_{j\in\mathcal{N}_i} F_{ij} x_{[j]}(t-k-1). \tag{4.14}$$

In view of (4.14) we can write

$$||x_{[i]}(t)||_\infty \leq ||F_{ii}^t||_\infty ||x_{[i]}(0)||_\infty + \sum_{j\in\mathcal{N}_i} \gamma_{ij} \max_{k\leq t} ||x_{[j]}(k)||_\infty \tag{4.15}$$

where γ_{ij} are the entries of Γ. In order to analyze the stability of the origin of (4.4), we consider the method proposed in [DRW07]. In view of Corollary 16 in [DRW07], using (4.15), the overall system (4.4) is asymptotically stable if the gain matrix Γ is Schur. As shown above this property is implied by (4.5). □

4.6.2 Proof of Proposition 4.2

Proof. We note that from (1.2) and (4.11) we can write

$$\bar{x}_{[i]} = \bar{A}_{ii}\bar{x}_{[i]} + \bar{B}_i\bar{u}_{[i]} + \sum_{j\in\mathcal{N}_i} \bar{A}_{ij}\bar{x}_{[j]} \tag{4.16a}$$

$$\bar{u}_{[i]} = \bar{K}_{ii}\bar{x}_{[i]} + \sum_{j\in\mathcal{N}_i} \bar{K}_{ij}\bar{x}_{[j]} \tag{4.16b}$$

where $\bar{x}_{[i]} = \begin{bmatrix} x_{[i]} \\ \hat{x}_{[i]} \end{bmatrix}$, $\bar{u}_{[i]} = \begin{bmatrix} u_{[i]} \\ z_{[i]} \end{bmatrix}$, $z_{[i]} \in \mathbb{R}^{\hat{n}_i}$ can be interpreted as additional integrators (see [BP70]), $\bar{A}_{ii} = \begin{bmatrix} A_{ii} & \mathbf{0}_{n_i \times \hat{n}_i} \\ \mathbf{0}_{\hat{n}_i \times n_i} & \mathbf{0}_{\hat{n}_i \times \hat{n}_i} \end{bmatrix}$, $\bar{A}_{ij} = \begin{bmatrix} A_{ij} & \mathbf{0}_{n_i \times \hat{n}_j} \\ \mathbf{0}_{\hat{n}_i \times n_j} & \mathbf{0}_{\hat{n}_i \times \hat{n}_j} \end{bmatrix}$, $\bar{B}_i = \begin{bmatrix} B_i & \mathbf{0}_{n_i \times \hat{n}_i} \\ \mathbf{0}_{\hat{n}_i \times m_i} & \mathbb{I}_{\hat{n}_i} \end{bmatrix}$, $\bar{K}_{ii} = \begin{bmatrix} K_{ii} & H_i \\ G_{ii} & S_i \end{bmatrix}$ and $\bar{K}_{ij} = \begin{bmatrix} \delta_{ij} K_{ij} & \mathbf{0}_{m_i \times \hat{n}_j} \\ \hat{\delta}_{ij} G_{ij} & \mathbf{0}_{\hat{n}_i \times \hat{n}_j} \end{bmatrix}$, $\forall i, j \in \mathcal{M}$. Hence subsystem (4.16a) is in the form of subsystem (1.2a) and (4.16b) in the form of (4.2), then we can prove Proposition 4.2 using the proof of Proposition 4.1. □

Chapter 5

Plug-and-play decentralized MPC based on nonlinear optimization

Contents

5.1 Introduction

In this chapter we propose a DeMPC schemes with PnP capabilities. Differently from Chapter 4, here we assume each subsystem is equipped with

local constraints. This means that we have to design local controllers that guarantee asymptotic stability and constraints satisfaction. To achieve our aims, we will exploit tube-based MPC [MSR05] for the design of robust local controllers. While this introduces an unavoidable degree of conservatism, we argue that PnP-DeMPC can be successfully applied in a number of real world plants where coupling among subsystems is sufficiently weak. As an example, we will use PnP-DeMPC for designing the Automatic Generation Control (AGC) layer for frequency control in a realistic power network and discuss plug in and out of generators areas.
The chapter is structured as follows. The design of decentralized controllers is introduced in Section 5.2 with a focus on the assumptions needed for guaranteeing asymptotic stability of the origin and constraint satisfaction. In Section 5.3 we discuss how to design the local controllers in a distributed fashion and we discuss the practical design of the local controllers. In Section 5.4 we describe PnP operations. In Section 5.5 we present the application of PnP-DeMPC to frequency control in a power network and Section 5.6 is devoted to concluding remarks.

5.2 Decentralized tube-based MPC of linear systems

We consider a discrete-time LTI LSS

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{5.1}$$

composed of M subsystems, as described in Section 1.5. In this chapter we consider that each subsystem is equipped with state and input constraints. More in detail, all subsystem $\Sigma_{[i]}$, $i \in \mathcal{M}$, are endowed with the constraints $x_{[i]} \in \mathbb{X}_i,\ u_{[i]} \in \mathbb{U}_i$. We define $\mathbb{X}_i$ as zonotopic sets

$$\begin{aligned} \mathbb{X}_i &= \{x_{[i]} \in \mathbb{R}^{n_i} : \mathcal{F}_i x_{[i]} \leq \mathbf{1}_{\bar{\tau}_i}\} \\ &= \{x_{[i]} \in \mathbb{R}^{n_i} : x_{[i]} = \Xi_i \digamma_i,\ ||\digamma_i||_\infty \leq 1\} \end{aligned} \tag{5.2}$$

where $\mathcal{F}_i = (f_{i,1}^T, \dots, f_{i,\bar{\tau}_i}^T) \in \mathbb{R}^{\bar{\tau}_i \times n_i}$, $\text{rank}(\mathcal{F}_i) = n_i$, $\digamma_i \in \mathbb{R}^{e_i}$, $\Xi_i \in \mathbb{R}^{n_i \times e_i}$. Furthermore, $\mathbb{U}_i$, $i \in \mathcal{M}$ are polytopic sets

$$\mathbb{U}_i = \{u_{[i]} \in \mathbb{R}^{m_i} : \mathcal{H}_i u_{[i]} \leq \mathbf{1}_{\tau_{u_i}}\},$$

where $\mathcal{H}_i = (h_{i,1}^T, \dots, h_{i,\tau_{u_i}}^T) \in \mathbb{R}^{\tau_{u_i} \times m_i}$.

5.2.1 Decentralized tube-based MPC

In this section we propose a decentralized controller for (5.1) guaranteeing asymptotic stability of the origin of the closed-loop system and constraint satisfaction.
In the spirit of tube-based control [MSR05], we treat $w_{[i]}$ in (1.3) as a disturbance and we define the nominal system

$$\hat{\Sigma}_{[i]}: \quad \hat{x}^+_{[i]} = A_{ii}\hat{x}_{[i]} + B_i v_{[i]} \tag{5.3}$$

with input $v_{[i]}$, obtained from (1.3) by neglecting the disturbance term $w_{[i]}$. Subsystem $\Sigma_{[i]}$ will be equipped with controller $\mathcal{C}_{[i]}$ given by

$$\mathcal{C}_{[i]}: \qquad u_{[i]} = v_{[i]} + K_i(x_{[i]} - \bar{x}_{[i]}) \tag{5.4}$$

where $K_i \in \mathbb{R}^{m_i \times n_i}$, $i \in \mathcal{M}$ and variables $v_{[i]}$ and $\bar{x}_{[i]}$ will be computed by a local state-feedback MPC scheme, i.e. there exist functions $\kappa_i(\cdot) : \mathbb{R}^{n_i} \to \mathbb{R}^{m_i}$ and $\eta_i(\cdot) : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ such that $v_{[i]} = \kappa_i(x_{[i]})$ and $\bar{x}_{[i]} = \eta_i(x_{[i]})$. Note that the controller $\mathcal{C}_{[i]}$ is completely decentralized, since it depends upon quantities of subsystem $\Sigma_{[i]}$ only.

Remark 5.1. In order to illustrate the meaning of (5.4), assume that $\bar{x}_{[i]}(t) = \hat{x}_{[i]}(t)$, $\forall t \geq 0$. Then, introducing the error $z_{[i]} = x_{[i]} - \hat{x}_{[i]}$, from (1.3), (5.3) and (5.4) we obtain

$$z^+_{[i]} = (A_{ii} + B_i K_i) z_{[i]} + w_{[i]}. \tag{5.5}$$

When $A_{ii} + B_i K_i$ is Schur and $w_{[i]}$ is bounded $\forall t \geq 0$, (5.5) guarantees that $x_{[i]} - \hat{x}_{[i]}$ remains bounded regardless of the exerted control action $v_{[i]}$. Moreover, if $w_{[i]}(t) = 0$, the state $x_{[i]}(t)$ achieves perfect tracking of the nominal state $\hat{x}_{[i]}(t)$ in the asymptotic regime.

Defining the collective variables

$$\bar{\mathbf{x}} = (\bar{x}_{[1]}, \ldots, \bar{x}_{[M]}) \in \mathbb{R}^n, \qquad \mathbf{v} = (v_{[1]}, \ldots, v_{[M]}) \in \mathbb{R}^m$$

and the matrix $\mathbf{K} = \text{diag}(K_1, \ldots, K_M) \in \mathbb{R}^{m \times n}$, from (1.3) and (5.4) one obtains the collective model

$$\mathbf{x}^+ = (\mathbf{A} + \mathbf{BK})\mathbf{x} + \mathbf{B}(\mathbf{v} - \mathbf{K}\bar{\mathbf{x}}). \tag{5.6}$$

The following assumptions will be needed for designing stabilizing controllers $\mathcal{C}_{[i]}$.

Assumption 5.1. *(i) The matrices $F_i = A_{ii} + B_i K_i$, $i \in \mathcal{M}$ are Schur.*

(ii) The matrix $\mathbf{F} = \mathbf{A} + \mathbf{BK}$ is Schur.

We discuss now constraint satisfaction, assuming subsystems $\Sigma_{[i]}$, $i \in \mathcal{M}$ are equipped with constraints $x_{[i]} \in \mathbb{X}_i$ and $u_{[i]} \in \mathbb{U}_i$. If we define the sets $\mathbb{X} = \prod_{i\in\mathcal{M}} \mathbb{X}_i$ and $\mathbb{U} = \prod_{i\in\mathcal{M}} \mathbb{U}_i$, then we obtain the following constraints for the collective system (5.1)

$$\mathbf{x} \in \mathbb{X}, \ \mathbf{u} \in \mathbb{U}. \tag{5.7}$$

As in tube-based MPC [MSR05], our goal is to compute tightened state constraints $\hat{\mathbb{X}}_i \subseteq \mathbb{X}_i$ and input constraints $\mathbb{V}_i \subseteq \mathbb{U}_i$ that, through (5.4), will allow us to fulfill (5.7) at time $k+1$ when $\bar{x}_{[i]}(k) \in \hat{\mathbb{X}}_i(k)$ and $v_{[i]}(k) \in \mathbb{V}_i(k)$. Next, we characterize the shape of sets $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$, $i \in \mathcal{M}$. Sets $\hat{\mathbb{X}}_i$ are zonotopes:

$$\begin{aligned} \hat{\mathbb{X}}_i &= \{\hat{x}_{[i]} \in \mathbb{R}^{n_i} : \hat{\mathcal{F}}_i \hat{x}_{[i]} \le \hat{l}_i \mathbf{1}_{\bar{\hat{r}}_i}\} \\ &= \{\hat{x}_{[i]} \in \mathbb{R}^{n_i} : \hat{x}_{[i]} = \hat{\Xi}_i \hat{\digamma}_i, \ ||\hat{\digamma}_i||_\infty \le \hat{l}_i\} \end{aligned} \tag{5.8}$$

where $\hat{l}_i \in \mathbb{R}_+$, $\hat{\mathcal{F}}_i = (\hat{f}^T_{i,1}, \dots, \hat{f}^T_{i,\bar{\hat{r}}_i}) \in \mathbb{R}^{\bar{\hat{r}}_i \times n_i}$, $\hat{\digamma}_i \in \mathbb{R}^{\hat{e}_i}$ and $\hat{\Xi}_i \in \mathbb{R}^{n_i \times \hat{e}_i}$. Moreover we assume that $\hat{\mathcal{F}}_i$ and $\hat{\Xi}_i$ are given, whereas $\hat{l}_i$ are free parameters that will be tuned in the control design procedure. Sets $\mathbb{V}_i$, $i \in \mathcal{M}$ are polytopes containing the origin in their interior, that, without loss of generality, are defined as follows

$$\begin{aligned} \mathbb{V}_i &= \{v_{[i]} \in \mathbb{R}^{m_i} : h^T_{i,\tau} v_{[i]} \le 1 - l_{v_{i,\tau}}, \ \forall \tau \in 1 : \tau_{u_i}\} \\ &= \{v_{[i]} \in \mathbb{R}^{m_i} : \mathcal{H}_i v_{[i]} \le \mathbf{1}_{\tau_{u_i}} - l_{v_i}\}, \end{aligned} \tag{5.9}$$

where $l_{v_i} = (l_{v_{i,1}}, \dots, l_{v_{i,\tau_{u_i}}})$ and $l_{v_{i,\tau}} \in \mathbb{R}_+$, $\tau \in 1 : \tau_{u_i}$ are free parameters. Similarly to [KG98], under Assumption 5.1-(i) and the definition of sets $\mathbb{X}_i$, $i \in \mathcal{M}$ there exist nonempty RPI sets $\mathbb{Z}_i \subseteq \mathbb{R}^{n_i}$, $i \in \mathcal{M}$ for the dynamics (5.5) and

$$w_{[i]} \in \mathbb{W}_i = \bigoplus_{j\in\mathcal{N}_i} A_{ij} \mathbb{X}_j. \tag{5.10}$$

In particular, for $\delta_i > 0$, we denote with $\mathbb{Z}_i(\delta_i)$ an RPI set that is a δ_i-outer approximation of the mRPI for (5.5) and $w_{[i]} \in \mathbb{W}_i$.
For guaranteeing (5.7) we introduce the following assumption.

Assumption 5.2. *For all $i \in \mathcal{M}$ there exist $\delta_i > 0$ and nonempty constraint sets $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$ verifying*

$$\hat{\mathbb{X}}_i \oplus \mathbb{Z}_i(\delta_i) \subseteq \mathbb{X}_i \tag{5.11a}$$

$$\mathbb{V}_i \oplus K_i\mathbb{Z}_i(\delta_i) \subseteq \mathbb{U}_i. \tag{5.11b}$$

Under Assumptions 5.1 and 5.2, as in [MSR05], we set in (5.4)

$$v_{[i]}(t) = \kappa_i(x_{[i]}(t)) = v_{[i]}(0|t), \quad \bar{x}_{[i]}(t) = \eta_i(x_{[i]}(t)) = \hat{x}_{[i]}(0|t) \tag{5.12}$$

where $v_{[i]}(0|t)$ and $\hat{x}_{[i]}(0|t)$ are optimal values of variables $v_{[i]}(0)$ and $\hat{x}_{[i]}(0)$, respectively, obtained by solving the following MPC-i problem at time t

$$\mathbb{P}_i^N(x_{[i]}(t)) :$$

$$\min_{\substack{\hat{x}_{[i]}(0)\\ v_{[i]}(0:N_i-1)}} \sum_{k=0}^{N_i-1} \ell_i(\hat{x}_{[i]}(k), v_{[i]}(k)) + V_{f_i}(\hat{x}_{[i]}(N_i)) \tag{5.13a}$$

$$x_{[i]}(t) - \hat{x}_{[i]}(0) \in \mathbb{Z}_i(\delta_i) \tag{5.13b}$$

$$\hat{x}_{[i]}(k+1) = A_{ii}\hat{x}_{[i]}(k) + B_i v_{[i]}(k) \qquad k \in 0 : N_i - 1 \tag{5.13c}$$

$$\hat{x}_{[i]}(k) \in \hat{\mathbb{X}}_i, \ \ v_{[i]}(k) \in \mathbb{V}_i \qquad k \in 0 : N_i - 1 \tag{5.13d}$$

$$\hat{x}_{[i]}(N_i) \in \hat{\mathbb{X}}_{f_i} \tag{5.13e}$$

In (5.13), $N_i \in \mathbb{N}$ is the prediction horizon, $\ell_i(\hat{x}_{[i]}(k), v_{[i]}(k)) : \mathbb{R}^{n_i \times m_i} \to \mathbb{R}_{0+}$ is the stage cost and $V_{f_i}(\hat{x}_{[i]}(N_i)) : \mathbb{R}^{n_i} \to \mathbb{R}_{0+}$ is the final cost, fulfilling the following assumption.

Assumption 5.3. *For all $i \in \mathcal{M}$, there exists an auxiliary control law $\kappa_i^{aux}(\hat{x}_{[i]})$ and a $\mathcal{K}_\infty$ function $\mathcal{B}_i$ such that:*

(i) $\ell_i(\hat{x}_{[i]}, v_{[i]}) \geq \mathcal{B}_i(||(\hat{x}_{[i]}, v_{[i]})||)$, *for all* $\hat{x}_{[i]} \in \mathbb{R}^{n_i}$, $v_{[i]} \in \mathbb{R}^{m_i}$ *and* $\ell_i(\boldsymbol{0}_{n_i}, \boldsymbol{0}_{m_i}) = 0$*;*

(ii) $\hat{\mathbb{X}}_{f_i} \subseteq \hat{\mathbb{X}}_i$ *is an invariant set for* $\hat{x}_{[i]}^+ = A_{ii}\hat{x}_{[i]} + B_i\kappa_i^{aux}(\hat{x}_{[i]})$*;*

(iii) $\forall \hat{x}_{[i]} \in \hat{\mathbb{X}}_{f_i}$, $\kappa_i^{aux}(\hat{x}_{[i]}) \in \mathbb{V}_i$*;*

(iv) $\forall \hat{x}_{[i]} \in \hat{\mathbb{X}}_{f_i}$, $V_{f_i}(\hat{x}_{[i]}^+) - V_{f_i}(\hat{x}_{[i]}) \leq -\ell_i(\hat{x}_{[i]}, \kappa_i^{aux}(\hat{x}_{[i]}))$.

We highlight that there are several methods, discussed e.g. in [RM09], for computing $\ell_i(\cdot)$, $V_{f_i}(\cdot)$ and $\mathbb{X}_{f_i}$ verifying Assumption 5.3.
The next theorem provides the main results on stability of the closed-loop system (5.6) and (5.12) equipped with constraints (5.7).

Theorem 5.1. *Let Assumptions 5.1-5.3 hold. Define the feasibility region for the MPC-i problem as*

$$\mathbb{X}_i^N = \{s_{[i]} \in \mathbb{X}_i : \quad (5.13) \text{ is feasible for } x_{[i]}(t) = s_{[i]}\}$$

and the collective feasibility region as $\mathbb{X}^N = \prod_{i\in\mathcal{M}} \mathbb{X}_i^N$.
Then

(i) if $\mathbf{x}(0) \in \mathbb{X}^N$*, i.e.* $x_{[i]}(0) \in \mathbb{X}_i^N$ *for all* $i \in \mathcal{M}$*, constraints (5.7) are fulfilled at all time instants;*

(ii) the origin of the closed-loop system (5.6) and (5.12) is asymptotically stable and $\mathbb{X}^N$ *is a region of attraction.*

Proof. The proof of Theorem 5.1 is given in Appendix 5.7.1. □

5.3 Decentralized synthesis of DeMPC

In this section we discuss the decentralized design of the DeMPC scheme given by (5.4) and (5.13). Our method hinges on the following proposition.

Proposition 5.1. *For given matrices* K_i, $i \in \mathcal{M}$*, verifying Assumption 5.1-(i), if the following conditions are fulfilled*

$$\alpha_i = \sum_{j\in\mathcal{N}_i} \sum_{k=0}^{\infty} ||\mathcal{F}_i F_i^k A_{ij} \mathcal{F}_j^\flat||_\infty < 1, \ \forall i \in \mathcal{M} \tag{5.14}$$

then

(I) Assumption 5.1-(ii) holds.

(II) For all $i \in \mathcal{M}$*, defining for* $\tau \in 1 : \bar{\tau}_i$

$$\underline{\hat{L}}_{i,\tau} = \frac{1 - \sum_{j\in\mathcal{N}_i} \sum_{k=0}^{\infty} ||f_{i,\tau}^T F_i^k A_{ij} \Xi_j||_\infty}{||f_{i,\tau}^T \hat{\Xi}_i||_\infty}, \tag{5.15}$$

there is $\delta_i > 0$ *such that*

$$\hat{L}_i = \min_{\tau\in 1:\bar{\tau}_i} \underline{\hat{L}}_{i,\tau} - \frac{||f_{i,\tau}^T||_\infty \delta_i}{||f_{i,\tau}^T \hat{\Xi}_i||_\infty} > 0. \tag{5.16}$$

Furthermore, choosing $\hat{l}_i \in (0, \hat{L}_i]$ *and the set* $\hat{\mathbb{X}}_i$ *as in* (5.8), *the inclusion* (5.11a) *holds.*

(III) For $\delta_i > 0$ *verifying* (5.16) *assume the following condition is fulfilled*

$$\beta_i(\delta_i) = \max_{\tau \in 1:\tau_{u_i}} \hat{l}_{v_{i,\tau}}(\delta_i) < 1 \tag{5.17}$$

with

$$\hat{l}_{v_{i,\tau}}(\delta_i) = \sup_{z_i \in \mathbb{Z}_i(\delta_i)} h_{i,\tau}^T K_i z_i, \ \tau \in 1 : \tau_{u_i}. \tag{5.18}$$

Then, choosing $\mathbb{V}_i$ *as in* (5.9) *for* $l_{v_{i,\tau}} = \hat{l}_{v_{i,\tau}}(\delta_i)$, *the inclusion* (5.11b) *holds.*

Proof. The proof of Proposition 5.1 is given in Appendix 5.7.2. □

As for conditions (4.5), the main tool used in the proof of Proposition 5.1, is the small gain theorem for networks [DRW07]. However, differently from Proposition 4.1, here conditions (5.14) take into account that local states (and hence coupling terms) are bounded. This explains why matrices $\mathcal{F}_i$ defined in (5.2) appear in (5.14).
We highlight that, for a given $i \in \mathcal{M}$, the quantities α_i in (5.14), $\hat{L}_i$ in (5.16), and $\beta_i(\delta_i)$ in (5.17) depend only upon local fixed parameters $\{A_{ii}, B_i, \mathcal{F}_i, \mathcal{H}_i\}$, parents' fixed parameters $\{A_{ij}, \Xi_j\}_{j \in \mathcal{N}_i}$ (or equivalently $\{A_{ij}, \mathcal{F}_j\}_{j \in \mathcal{N}_i}$) and local tunable parameters $\{K_i, \delta_i\}$ but not on parents' tunable parameters. Moreover, also the computation of sets $\mathbb{Z}_i(\delta_i)$ depends upon the same parameters. This implies that the choice of $\{K_i, \delta_i\}$ does not influence the choice of $\{K_j, \delta_j\}_{j \neq i}$ and therefore, in order to verify Assumptions 5.1 and 5.2, we need to solve the following independent problems for $i \in \mathcal{M}$.

Problem 5.1 ($\mathcal{P}_i$)**.** Check if there exist K_i and $\delta_i > 0$ such that $\alpha_i < 1$, $\hat{L}_i > 0$ and $\beta_i(\delta_i) < 1$.

The solution to Problems $\mathcal{P}_i$ also allows for the decentralized design of controllers MPC-i that, using the synthesis methods reviewed in [RM09], can also satisfy Assumption 5.3. The overall procedure for the decentralized synthesis of local controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ is summarized in Algorithm 5.1.
Next, we propose an automatic method for computing the matrix K_i and $\delta_i > 0$ in Step (I) of Algorithm 5.1. Differently from [RFFT13b], next we describe the method implemented in the *PnPMPC-toolbox* for MatLab

Algorithm 5.1 Design of controller $\mathcal{C}_{[i]}$ for subsystem $\Sigma_{[i]}$

Input: A_{ii}, B_i, $\mathbb{X}_i$, $\mathbb{U}_i$, $\mathcal{N}_i$, $\{A_{ij}\}_{j\in\mathcal{N}_i}$, $\{\mathbb{X}_j\}_{j\in\mathcal{N}_i}$
Output: controller $\mathcal{C}_{[i]}$ in (5.4)

(I) Find K_i and $\delta_i > 0$ such that Assumption 5.1-(i) is fulfilled, $\alpha_i < 1$, (5.16) holds and $\beta_i(\delta_i) < 1$. If they do not exist **stop** (the controller $\mathcal{C}_{[i]}$ cannot be designed).

(II) Compute sets $\mathbb{W}_i = \bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{X}_j$ and $\mathbb{Z}_i(\delta_i)$.

(III) Compute $\hat{L}_i$ as in (5.16), choose $\hat{l}_i = \hat{L}_i$ and define $\hat{\mathbb{X}}_i$ as in (5.8).

(IV) Compute $\hat{l}_{v_{i,\tau}}(\delta_i)$ as in (5.18), set $l_{v_{i,\tau}} = \hat{l}_{v_{i,\tau}}(\delta_i)$ and define $\mathbb{V}_i$ as in (5.9).

(V) Compute $\ell_i(\cdot)$, $V_{f_i}(\cdot)$ and $\hat{\mathbb{X}}_{f_i}$ verifying Assumption 5.3.

[RBFT12] (see Appendix C). We solve the nonlinear optimization problem

$$\min_{\delta_i,\ K_i} \ \mu_i \tag{5.19a}$$

$$\bar{\rho}(A_{ii} + B_iK_i) < 1 \tag{5.19b}$$

$$\delta_i > 0,\ \ \alpha_i < 1,\ \ \hat{L}_i > 0,\ \ \beta_i(\delta_i) < 1 \tag{5.19c}$$

where $\mu_i = \max(\alpha_i, \beta_i(\delta_i), \bar{\rho}(A_{ii} + B_iK_i))$. Since (5.19) is a nonlinear optimization problem, suitable initializations of K_i and δ_i are needed, therefore we initialize δ_i with a small value and K_i as the LQR gain associated to matrices $\hat{Q}_i \geq \mathbf{0}_{n_i\times n_i}$ and $\hat{R}_i > \mathbf{0}_{m_i\times m_i}$ (see formula (4.8)), that are considered as inputs for the nonlinear optimization. We highlight that in several case studies this procedure for initializing the optimization algorithm has proved very effective. We also note that problem (5.19) allows us to compute a matrix K_i that fulfills all assumptions of Proposition 5.1.
A few remarks on the computations required for solving (5.19) are in order. First, beside the computation of K_i, problem (5.19) requires the computation of the set $\mathbb{Z}_i(\delta_i)$ that can be done using methods in [RKKM05], simplified as follows. Under the definition (5.2) of sets $\mathbb{X}_i$, the set $\mathbb{W}_i = \oplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{X}_j$ is a zonotope and can be written as $\mathbb{W}_i = \{w_{[i]} \in \mathbb{R}^{n_i} : w_{[i]} = \Xi_{w_i}\digamma_{w_i},\ ||\digamma_{w_i}||_\infty \leq 1\}$, with $\Xi_{w_i} \in \mathbb{R}^{n_i\times n_{w_i}}$ and $\digamma_{w_i} \in \mathbb{R}^{n_{w_i}}$. Hence, using

the procedure proposed in [RKKM05], the set $\mathbb{Z}_i(\delta_i)$ is also a zonotope, that can be written as $\mathbb{Z}_i(\delta_i) = \{z_{[i]} \in \mathbb{R}^{n_i} : z_{[i]} = \Xi_{z_i} \digamma_{z_i},\ ||\digamma_{z_i}||_\infty \leq 1\}$, with $\Xi_{z_i} \in \mathbb{R}^{n_i \times n_{z_i}}$ and $\digamma_{z_i} \in \mathbb{R}^{n_{z_i}}$, where $\Xi_{z_i} = \begin{bmatrix} \Xi_{w_i} & F_{ii}\Xi_{w_i} & \dots & F_{ii}^{s_i-1}\Xi_{w_i} \end{bmatrix}$ with $s_i \in \mathbb{N}_+$ computed using Algorithm 5.1 in [RKKM05]. Since $\mathbb{W}_i$ and $\mathbb{Z}_i(\delta_i)$ are zonotopes, using (A.2) we can derive an explicit formula for the support functions used in Algorithm 5.1 in [RKKM05] and rewrite (5.18) as

$$\hat{l}_{v_{i,\tau}}(\delta_i) = ||\Xi_{z_i}^T K_i^T h_{i,\tau}||_1,\ \forall \tau \in 1 : \tau_{u_i}.$$

Second, we highlight that in absence of input constraints $\mathbb{U}_i$, constraint $\beta_i(\delta_i) < 1$ is not necessary, hence in (5.19) the explicit computation of RPI sets $\mathbb{Z}_i(\delta_i)$ is not required. Indeed if $\mathbb{U}_i = \mathbb{R}^{m_i}$, the inclusion (5.11b) holds for all sets $\mathbb{V}_i \subseteq \mathbb{R}^{m_i}$. Third, the series in (5.14) and (5.15) involve only positive terms and can be easily truncated if either the involved inequalities are violated or summands fall below the machine precision. Finally, we highlight that the feasibility of problem (5.19) guarantees that the controller $\mathcal{C}_{[i]}$ can be successfully designed. In this respect, feasibility of (5.19) provides an automatic way for testing the applicability of our design method.

5.4 Plug-and-play operations

Consider a plant composed by subsystems $\Sigma_{[i]}$, $i \in \mathcal{M}$ equipped with local controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ produced by Algorithm 5.1. In case subsystems are added or removed, existing controllers have to be redesigned. In this Section we propose a PnP distributed solution, which requires the redesign of a limited number of controllers. As mentioned in Section 1.1.5, we assume subsystems get plugged in and out offline.

5.4.1 Plug-in operation

We start considering plug in of subsystem $\Sigma_{[M+1]}$, characterized by parameters $A_{M+1,M+1}$, B_{M+1}, $\mathbb{X}_{M+1}$, $\mathbb{U}_{M+1}$, $\mathcal{N}_{M+1}$ and $\{A_{M+1,j}\}_{j\in\mathcal{N}_{M+1}}$, into an existing plant. In particular $\mathcal{N}_{M+1}$ identifies the subsystems that will be physically coupled to $\Sigma_{[M+1]}$ and $\{A_{M+1,j}\}_{j\in\mathcal{N}_{M+1}}$ are the corresponding coupling terms. For designing the controller $\mathcal{C}_{[M+1]}$ we execute Algorithm 5.1 that needs information only from systems $\Sigma_{[j]}$, $j \in \mathcal{N}_{M+1}$. If Algorithm 5.1 stops before the last step we declare that $\Sigma_{[M+1]}$ cannot be plugged in. Let

$$\mathcal{S}_i = \{j : i \in \mathcal{N}_j\} \tag{5.20}$$

be the set of children of subsystem i. Since $\Sigma_{[M+1]}$ is now a new parent of subsystems $\Sigma_{[j]}$, $j \in \mathcal{S}_{M+1}$, existing matrices K_j, $j \in \mathcal{S}_{M+1}$ may now result in $\alpha_j \geq 1$ or $\hat{L}_j \leq 0$ or $\beta_i(\delta_i) \geq 1$. Indeed, when $\mathcal{N}_j$ gets larger, the quantity α_j in (5.14) (respectively $\hat{L}_j$ in (5.16)) can only increase (respectively decrease). Furthermore, the size of the set $\mathbb{Z}_j(\delta_j)$ increases (because the set $\mathbb{W}_i$ in (5.10), gets bigger) and therefore the condition in (5.17) could be violated. This means that for each $j \in \mathcal{S}_{M+1}$ the controllers $\mathcal{C}_{[j]}$ must be redesigned according to Algorithm 5.1. Again, if Algorithm 5.1 stops before completion for some $j \in \mathcal{S}_{M+1}$, we declare that $\Sigma_{[M+1]}$ cannot be plugged in.
In conclusion, the addition of system $\Sigma_{[M+1]}$ triggers the design of controller $\mathcal{C}_{[M+1]}$ and the redesign of controllers $\mathcal{C}_{[j]}$, $j \in \mathcal{S}_{M+1}$ according to Algorithm 5.1. Note that controller redesign does not propagate further in the network, i.e. even without changing controllers $\mathcal{C}_{[i]}$, $i \notin \{M+1\} \bigcup \mathcal{S}_{M+1}$ stability of the origin and constraint satisfaction are guaranteed for the new closed-loop system.

5.4.2 Unplugging operation

We consider plug out of subsystem $\Sigma_{[k]}$, $k \in \mathcal{M}$ and define $\mathcal{S}_k$ as in (5.20). Since for each $i \in \mathcal{S}_k$ the set $\mathcal{N}_i$ gets smaller, we have that α_i in (5.14) (respectively $\hat{L}_i$ in (5.16)) cannot increase (respectively decrease). Furthermore, the size of the set $\mathbb{Z}_i(\delta_i)$ cannot increase and therefore the inequality (5.17) cannot be violated. This means that for each $i \in \mathcal{S}_k$ the controller $\mathcal{C}_{[i]}$ does not have to be redesigned. Moreover, since for each system $\Sigma_{[j]}$, $j \notin \{k\} \bigcup \mathcal{S}_k$, the set $\mathcal{N}_j$ does not change, the redesign of controller $\mathcal{C}_{[j]}$ is not required.
In conclusion, the removal of system $\Sigma_{[k]}$ does not require the redesign of any controller, in order to guarantee stability of the origin and constraint satisfaction for the new closed-loop system. However systems $\Sigma_{[i]}$, $i \in \mathcal{S}_k$, have one parent less and redesign of controllers $\mathcal{C}_{[i]}$ through Algorithm 5.1 could improve the performance. Furthermore, as discussed in Section 4.3.3, redesign is mandatory when matrices A_{ii} and B_i contain parameters that depend upon parent subsystems (see Section 5.5.3 for an example).

5.5 Example: power network system

In this section, we apply the proposed DeMPC scheme to the PNS proposed in Appendix B. In particular we will show advantages brought about by PnP-DeMPC when generation areas are connected/disconnected to/from an existing network. In the following we first design the AGC layer for the PNS of Scenario 1 in B.1.1 and then we show how in presence of connection/disconnection of an area (Scenario 2 and 3, in Sections B.1.2 and B.1.2, respectively) the AGC can be redesigned via plug-in and unplugging operations. All controllers have been designed using the *PnPMPC-toolbox* for MatLab [RBFT12][1] (see also Appendix C).

5.5.1 Scenario 1

We consider the PNS proposed in Section B.1.2. For each subsystem $\Sigma_{[i]}$ we synthesize the controller K_i, $i \in \mathcal{M}$ solving the nonlinear optimization problem (5.19) with $\hat{Q}_i$ and $\hat{R}_i$ as in (B.4), $\forall i \in \mathcal{M}$, and obtain the following matrices

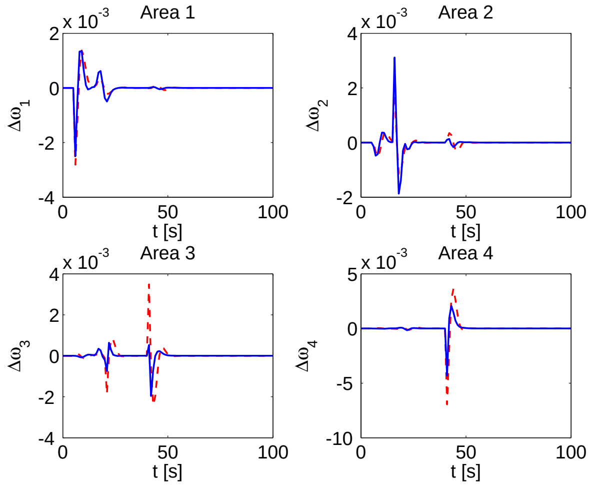
$$
\begin{aligned}
K_1 &= -\begin{bmatrix} 5.431 & 1.632 & 0.246 & -0.202 \end{bmatrix}, \\
K_2 &= -\begin{bmatrix} 3.315 & -0.660 & 0.628 & -0.934 \end{bmatrix}, \\
K_3 &= -\begin{bmatrix} 6.705 & 10.943 & 0.510 & -0.193 \end{bmatrix}, \\
K_4 &= -\begin{bmatrix} 6.461 & 13.008 & 0.574 & 0.176 \end{bmatrix},
\end{aligned}
$$

that allow inequalities (5.14) to be fulfilled. Hence $\mathbf{K}$ verifies Assumption 5.1-(ii). Setting $\delta_i = 10^{-5}$, $\forall i \in \mathcal{M}$ and applying Steps (I)-(V) of Algorithm 5.1, we can compute sets $\mathbb{Z}_i(\delta_i)$, $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$ such that inclusions (5.11a) and (5.11b) hold, $\forall i \in \mathcal{M}$. Control variables $u_{[i]}$ are obtained through (5.4) where $v_{[i]} = \kappa_i(x_{[i]})$ and $\bar{x}_{[i]} = \eta_i(x_{[i]})$ are computed at each time t solving the optimization problem (5.13) and replacing the cost function in (5.13a) with the following one depending upon $x^O_{[i]}$ and $u^O_{[i]}$ (see Section B.2 for more details)

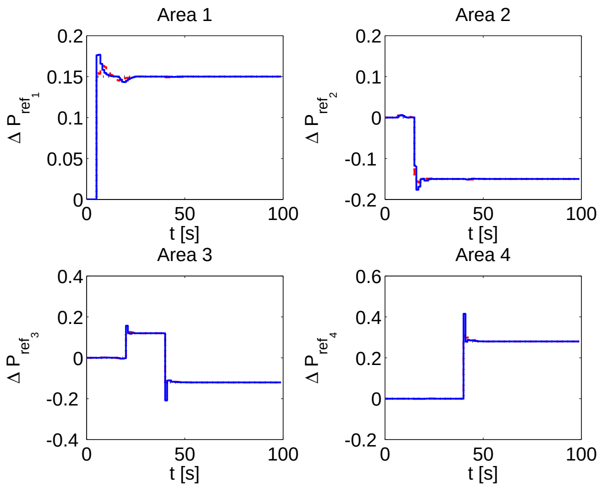
$$
\sum_{k=t}^{t+N_i-1} (||\hat{x}_{[i]}(k) - x^O_{[i]}||^2_{Q_i} + ||v_{[i]}(k) - u^O_{[i]}||^2_{R_i}) + ||x_{[i]}(t+N_i) - x^O_{[i]}||^2_{S_i}.
$$

Note that, except for the above modification of the cost function, that is needed for counteracting load disturbances, we followed exactly the design procedure described in Section 5.2. Moreover, we highlight that each area can locally absorb the load steps specified in Table B.3 of Appendix B. This is also shown by convergence to zero of the power transfer between areas i and j represented in Figure 5.2.

[1] All simulations have been done using a MacOS 10.7.5, with processor Intel Core i5, 1.7 GHz, MatLab r2013a, solver CPLEX [IBM11], YALMIP [Lö04] and MPT [KGB04].



(a) Frequency deviation in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).



(b) Load reference set-point in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

Figure 5.1: Simulation Scenario 1: 5.1a Frequency deviation and 5.1b Load reference in each area.

In Figure 5.1 we compare the performance of proposed DeMPC with the performance of CeMPC. For CeMPC we consider the controller proposed in Section B.2. In the control experiment, step power loads ΔP_{L_i} specified in Table B.3 of Appendix B have been used and they account for the step-

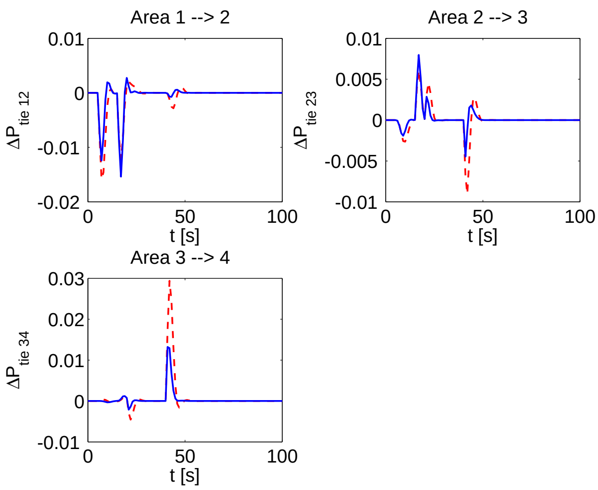


Figure 5.2: Simulation Scenario 1: tie-line power between each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

like changes of the control variables in Figure 5.1. We highlight that the performance of DeMPC and CeMPC are similar, in terms of frequency deviation (Figure 5.1a), control variables (Figure 5.1b) and power transfers $\Delta P_{tie_{ij}}$ (Figure 5.2). A detailed analysis, using performance criteria in Section B.2.1, shows that, in each area, the proposed DeMPC allows to reject power loads producing more local power, instead of importing power from parent areas. This is highlighted in Tables 5.1 and 5.2 where since DeMPC import less power than CeMPC, the performance index η is bigger, while the performance index Φ is much lower than CeMPC.

5.5.2 Scenario 2

We consider the power network proposed in Scenario 1 and we add a fifth area connected as in Section B.1.2. Therefore, the set of children of subsystem 5 is $\mathcal{S}_5 = \{2, 4\}$. As described in Section 5.4.1, only systems $\Sigma_{[j]}$, $j \in \mathcal{S}_5$ update their controller $\mathcal{C}_{[j]}$. For subsystems $\Sigma_{[j]}$, $j \in \mathcal{S}_5$, since the set $\mathcal{N}_j$ changes, we retune controllers $\mathcal{C}_{[j]}$ using Algorithm 5.1. In particular, we compute K_j, $j \in \mathcal{S}_5$ and K_5 using the procedure described in Section

5.3 with $\hat{Q}_k$ and $\hat{R}_k$ as in (B.4), $k \in \{5\} \bigcup \mathcal{S}_5$ and obtain

$$K_2 = -\begin{bmatrix}2.692 & -0.143 & 0.641 & -0.869\end{bmatrix},$$
$$K_4 = -\begin{bmatrix}2.474 & 0.102 & 0.598 & -0.740\end{bmatrix},$$
$$K_5 = -\begin{bmatrix}2.769 & -2.966 & 0.484 & -0.472\end{bmatrix},$$

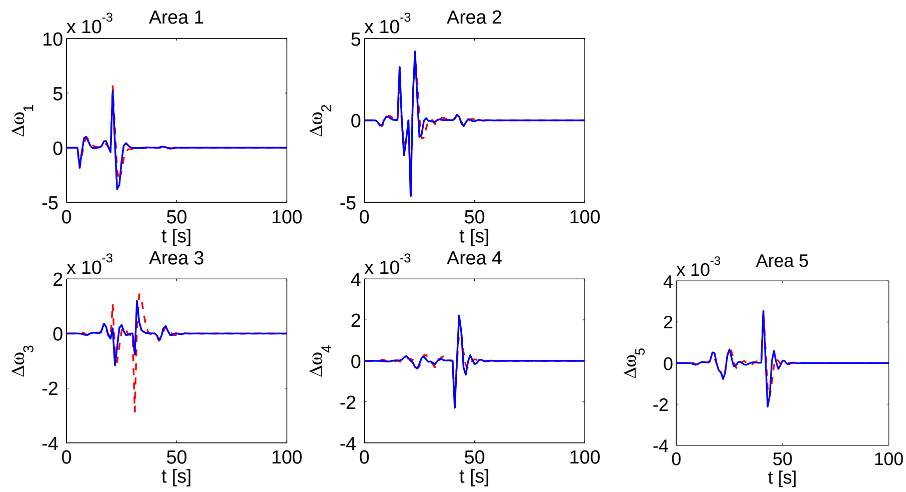
that allow inequalities (5.14) to be verified for systems $\Sigma_{[j]}$, $j \in \mathcal{S}_5$ and $\Sigma_{[5]}$. Therefore $\mathbf{K}$ fulfills Assumption 5.1-(ii). Setting $\delta_j = 10^{-5}$, $j \in \mathcal{S}_5$ and $\delta_5 = 10^{-5}$, the execution of Algorithm 5.1 does not stop before completion and hence we compute the new sets $\mathbb{Z}_j(\delta_j)$, $\hat{\mathbb{X}}_j$ and $\mathbb{V}_j$, $j \in \{5\} \bigcup \mathcal{S}_5$. We highlight that no retuning of controllers $\mathcal{C}_{[1]}$ and $\mathcal{C}_{[3]}$ is needed since subsystems $\Sigma_{[1]}$ and $\Sigma_{[3]}$ are not children of subsystem $\Sigma_{[5]}$.
In Figure 5.3 we compare the performance of proposed DeMPC with the performance of CeMPC. In the control experiment, step power loads ΔP_{L_i} specified in Table B.4 in Section B.1.2 have been used and they account for the step-like changes of the control variables in Figure 5.3. We highlight that the performance of DeMPC and CeMPC are similar, in terms of frequency deviation (Figure 5.3a), control variables (Figure 5.3b) and power transfers $\Delta P_{tie_{ij}}$ (Figure 5.4). Moreover similarly to Scenario 1, DeMPC as better performance than CeMPC in terms of power exchanged (see performance criteria Φ in Table 5.2).

5.5.3 Scenario 3

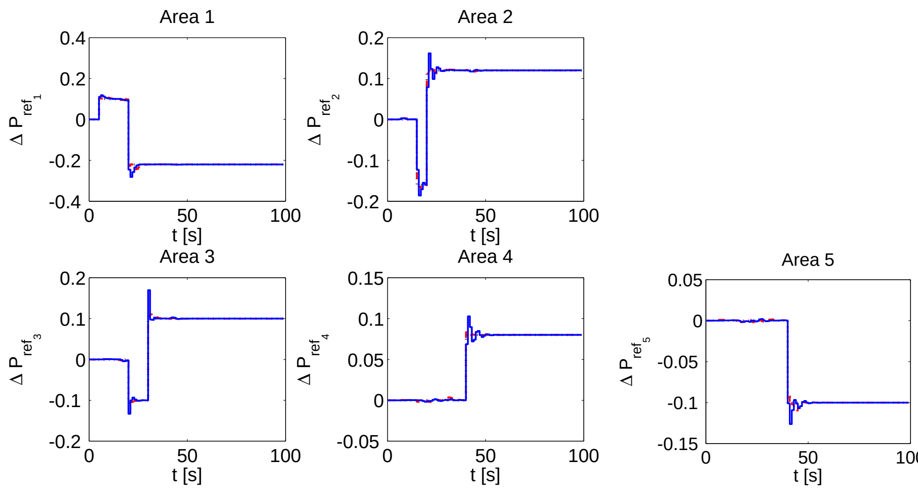
We consider the power network described in Scenario 2 and disconnect the area 4, as in Section B.1.3. The set of children of subsystem 4 is $\mathcal{S}_4 = \{3, 5\}$. Because of disconnection, subsystems $\Sigma^C_{[j]}$, $j \in \mathcal{S}_4$ change their parents and local dynamics A_{jj}. Moreover, it is possible to verify that matrices K_j computed in Scenario 2 do not solve Problem $\mathcal{P}_j$, $j \in \mathcal{S}_4$. Then as described in Section 5.4.2, each subsystem $\Sigma^C_{[j]}$, $j \in \mathcal{S}_4$ must retune controller $\mathcal{C}_{[j]}$ by running Algorithm 5.1. In particular, we compute K_3 and K_5 using the procedure proposed in Section 5.3 with $\hat{Q}_i$ and $\hat{R}_i$ as in (B.4), $j \in \mathcal{S}_4$ and obtain

$$K_3 = -\begin{bmatrix}3.402 & 2.110 & 0.163 & -0.273\end{bmatrix},$$
$$K_5 = -\begin{bmatrix}3.230 & -2.087 & 0.236 & -0.150\end{bmatrix},$$

that allows one to verify inequalities (5.14) for systems $\Sigma_{[j]}$, $j \in \mathcal{S}_4$. Therefore $\mathbf{K}$ is such that Assumption 5.1-(ii) holds. Setting $\delta_j = 10^{-5}$, $j \in \mathcal{S}_4$,



(a) Frequency deviation in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).



(b) Load reference set-point in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

Figure 5.3: Simulation Scenario 2: 5.3a Frequency deviation and 5.3b Load reference in each area.

the execution of Algorithm 5.1 does not stop before completion and hence we compute the new sets $\mathbb{Z}_j(\delta_j)$, $\hat{\mathbb{X}}_j$ and $\mathbb{V}_j$, $j \in \mathcal{S}_4$. We highlight that retuning of controllers $\mathcal{C}_{[1]}$ and $\mathcal{C}_{[2]}$ is not needed since systems $\Sigma_{[1]}$ and $\Sigma_{[2]}$ are not children of subsystem $\Sigma_{[4]}$.

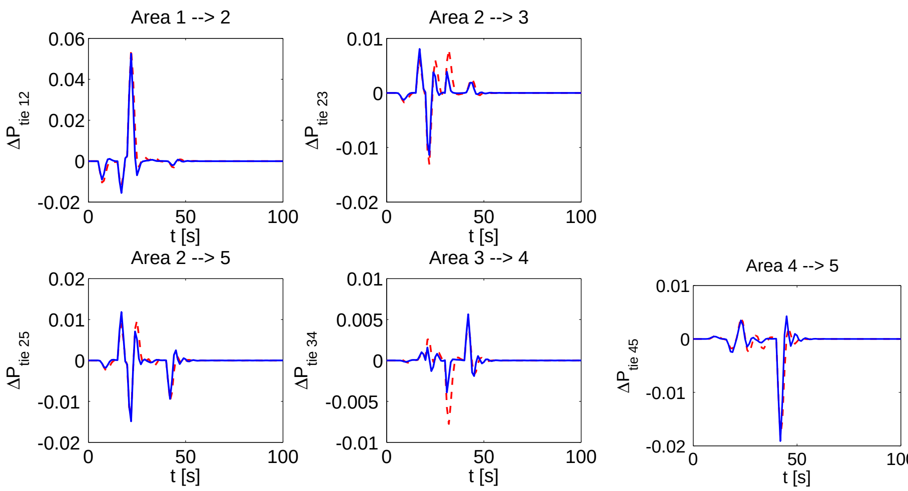
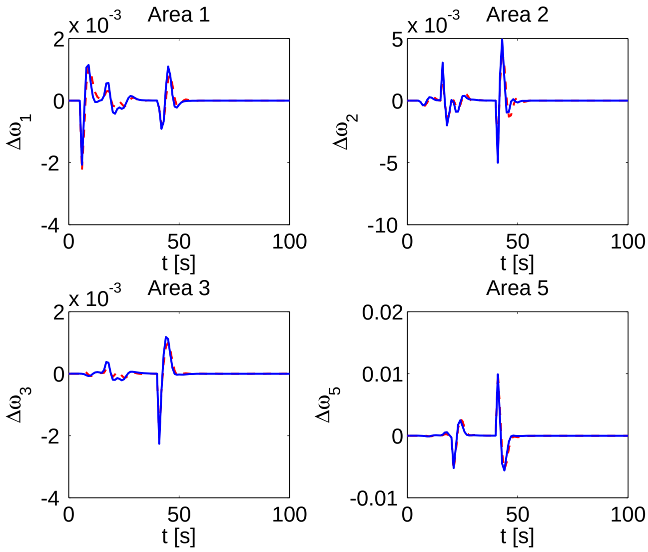


Figure 5.4: Simulation Scenario 2: tie-line power between each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

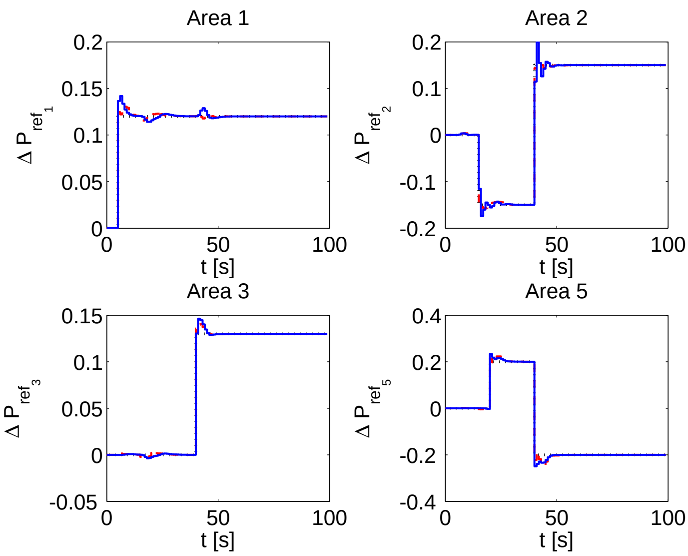
In Figure 5.5 we compare the performance of proposed DeMPC with the performance of CeMPC. In the control experiment, step power loads ΔP_{L_i} specified in Table B.5 in Section B.1.3 have been used also in this case. We highlight that the performance of DeMPC and CeMPC are similar in terms of frequency deviation (Figure 5.5a), control variables (Figure 5.5b) and power transfers $\Delta P_{tie_{ij}}$ (Figure 5.6). Moreover similarly to Scenarios 1 and 2, DeMPC as better performance than CeMPC in terms of power exchanged (see performance criteria Φ in Table 5.2).

	Scenario 1		Scenario 2		Scenario 3	
	MPCdiag	*MPCzero*	*MPCdiag*	*MPCzero*	*MPCdiag*	*MPCzero*
CeMPC	0.0249	0.0249	0.0346	0.0347	0.0510	0.0511
PnP-DeMPC	+20.88%	+20.88%	+10.40%	+10.09%	+4.51%	+4.31%

Table 5.1: Value of the performance parameter η for CeMPC (first line) and percentage change using DeMPC schemes proposed in this chapter for the AGC layer.



(a) Frequency deviation in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).



(b) Load reference set-point in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

Figure 5.5: Simulation Scenario 3: 5.5a Frequency deviation and 5.5b Load reference in each area.

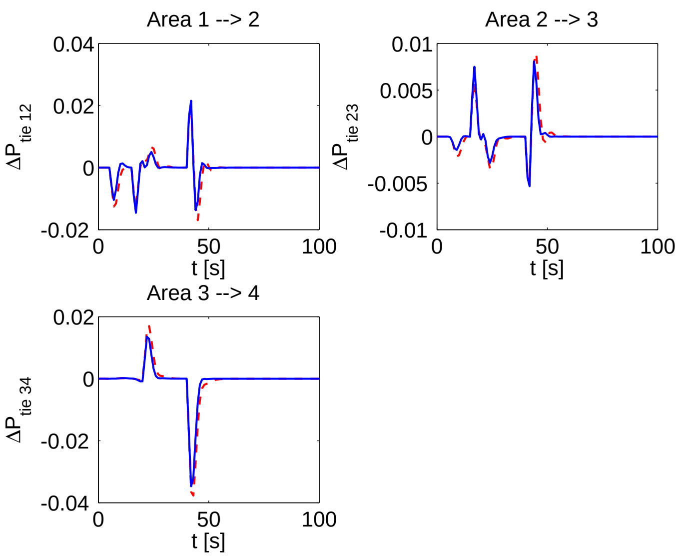


Figure 5.6: Simulation Scenario 3: tie-line power between each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

	Scenario 1		Scenario 2		Scenario 3	
	MPCdiag	*MPCzero*	*MPCdiag*	*MPCzero*	*MPCdiag*	*MPCzero*
CeMPC	0.0030	0.0029	0.0063	0.0061	0.0060	0.0058
PnP-DeMPC	−46.67%	−44.83%	−25.40%	−22.95%	−25.00%	−22.41%

Table 5.2: Value of the performance parameter Φ for CeMPC (first line) and percentage change using DeMPC schemes proposed in this chapter for the AGC layer.

5.6 Final comments

In this chapter we proposed a tube-based DeMPC scheme for linear constrained systems, with the goal of stabilizing the origin of the closed-loop system and guaranteeing constraints satisfaction. The key feature of our approach is that the design procedure does not require any centralized computation. Furthermore, it enables PnP operations, that can be performed solving suitable nonlinear optimization problems. In Chapter 6 we will propose a tube-base DeMPC that allows one to design each controller $\mathcal{C}_{[i]}$ solving a suitable LP problem. Moreover in Chapter 7 we will show how to attenuate coupling terms with parent subsystems and how to enhance robustness of the DeMPC scheme. In Chapter 9 we will extend the proposed PnP-DeMPC to the output-feedback case.

5.7 Appendix

5.7.1 Proof of Theorem 5.1

Proof. The proof uses arguments similar to the ones adopted in [FS12] for proving Theorem 1.
We first show recursive feasibility, i.e. that $x_{[i]}(t) \in \mathbb{X}_i^N,\ \forall i \in \mathcal{M}$ implies $x_{[i]}(t+1) \in \mathbb{X}_i^N$.
Assume that, at instant t, $x_{[i]}(t) \in \mathbb{X}_i^N$. The optimal nominal input and state sequences obtained by solving each MPC-i problem $\mathbb{P}_i^N$ are

$$v_{[i]}(0 : N_i - 1|t) = \{v_{[i]}(0|t), \dots, v_{[i]}(N_i - 1|t)\}$$

and

$$\hat{x}_{[i]}(0 : N_i|t) = \{\hat{x}_{[i]}(0|t), \dots, \hat{x}_{[i]}(N_i|t)\},$$

respectively. Define $v_{[i]}^{aux}(N_i|t) = \kappa_i^{aux}(\hat{x}_{[i]}(N_i|t))$ and compute $\hat{x}_{[i]}^{aux}(N_i + 1|t)$ according to (5.13c) from $\hat{x}_{[i]}(N_i|t)$ and $v_{[i]}(N_i|t) = v_{[i]}^{aux}(N_i|t)$. Note that, in view of constraint (5.13e) and points (ii) and (iii) of Assumption 5.3, $v_{[i]}^{aux}(N_i|t) \in \mathbb{V}_i$ and $\hat{x}_{[i]}^{aux}(N_i + 1|t) \in \hat{\mathbb{X}}_{f_i} \subseteq \hat{\mathbb{X}}_i$. We also define the input sequence

$$\bar{v}_{[i]}(1 : N_i|t) = \{v_{[i]}(1|t), \dots, v_{[i]}(N_i - 1|t), v_{[i]}^{aux}(N_i|t)\} \tag{5.21}$$

and the state sequence produced by (5.13c) from the initial condition $\hat{x}_{[i]}(0|t)$ and the input sequence $\bar{v}_{[i]}(1 : N_i|t)$, i.e.

$$\bar{\hat{x}}_{[i]}(1 : N_i + 1|t + 1) = \{\hat{x}_{[i]}(1|t), \dots, \hat{x}_{[i]}(N_i|t), \hat{x}_{[i]}^{aux}(N_i + 1|t)\}. \tag{5.22}$$

In view of the constraints (5.13) at time t and recalling that $\mathbb{Z}_i(\delta_i)$ is an RPI for (5.13) and $w_{[i]} \in \mathbb{W}_i = \bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{X}_j$, we have that $x_{[i]}(t+1) - \hat{x}_{[i]}(1|t) \in \mathbb{Z}_i(\delta_i)$. Therefore, we can conclude that the state and the input sequences $\bar{\hat{x}}_{[i]}(1 : N_i + 1|t)$ and $\bar{v}_{[i]}(1 : N_i|t)$ are feasible at $t + 1$, since constraints (5.13b)-(5.13e) are satisfied. This proves recursive feasibility.
We now prove convergence of the optimal cost function.
We define

$$\mathbb{P}_i^{N,0}(\hat{x}_{[i]}(0|t)) = \min_{v_{[i]}(0:N_i-1|t)} \sum_{k=0}^{N_i-1} \ell_i(\hat{x}_{[i]}(k), v_{[i]}(k)) + V_{f_i}(\hat{x}_{[i]}(N_i))$$

subject to the constraints (5.13c)-(5.13e). By optimality, using the feasible control law (5.21) and the corresponding state sequence (5.22) one has

$$\mathbb{P}_i^{N,0}(\hat{x}_{[i]}(1|t)) \leq \sum_{k=1}^{N_i} \ell_i(\hat{x}_{[i]}(k|t), v_{[i]}(k|t)) + V_{f_i}(\hat{x}_{[i]}^{aux}(N_i+1|t+1)) \quad (5.23)$$

where it has been set $v_{[i]}(N_i|t) = v_{[i]}^{aux}(N_i|t)$. Therefore we have

$$\begin{aligned}\mathbb{P}_i^{N,0}(\hat{x}_{[i]}(1|t)) - \mathbb{P}_i^{N,0}(\hat{x}_{[i]}(0|t)) &\leq -\ell_i(\hat{x}_{[i]}(0|t), v_{[i]}(0|t)) + \\ &+ \ell_i(\hat{x}_{[i]}(N_i|t), v_{[i]}^{aux}(N_i|t)) + V_{f_i}(\hat{x}_{[i]}^{aux}(N_i+1|t)) - V_{f_i}(\hat{x}_{[i]}^{aux}(N_i|t)).\end{aligned} \quad (5.24)$$

In view of Assumption 5.3-(iv), from (5.24) we obtain

$$\mathbb{P}_i^{N,0}(\hat{x}_{[i]}(1|t)) - \mathbb{P}_i^{N,0}(\hat{x}_{[i]}(t)) \leq -\ell_i(\hat{x}_{[i]}(t), v_{[i]}(t))$$

and therefore $\hat{x}_{[i]}(0|t) \to \mathbf{0}_{n_i}$ and $v_{[i]}(0|t) \to \mathbf{0}_{m_i}$ as $t \to \infty$.
Next we prove convergence to zero of state trajectories $\mathbf{x}(t)$ of the closed-loop system with $\mathbf{x}(0) \in \mathbb{X}^N$.
Recall that the state $\mathbf{x}(t)$ evolves according to the equation (5.6). By asymptotic convergence to zero of the nominal state and input signals $\hat{x}_{[i]}(0|t)$ and $v_{[i]}(0|t)$ respectively, using the diagonal structure of $\mathbf{B}$ and $\mathbf{K}$, we obtain that $\mathbf{B}(\mathbf{v}(0|t) - \mathbf{K}\hat{\mathbf{x}}(0|t))$ is an asymptotically vanishing term. Under Assumption 5.1-(ii), $\mathbf{A}+\mathbf{BK}$ is Schur, hence we obtain $\mathbf{x}(t) \to \mathbf{0}_n$ as $t \to \infty$. □

5.7.2 Proof of Proposition 5.1

Proof of (I)

Proof. Define a matrix $\mathbb{M}$ such that its ij-th entry μ_{ij} is

$$\begin{array}{ll}\mu_{ij} = -1 & \text{if} \quad i = j \\ \mu_{ij} = \sum_{k=0}^{\infty} \|\mathcal{F}_i F_i^k A_{ij} \mathcal{F}_j^\flat\|_\infty & \text{if} \quad i \neq j.\end{array}$$

Note that all the off-diagonal entries of matrix $\mathbb{M}$ are non-negative, i.e., $\mathbb{M}$ is Metzler (see Section A.2 in Appendix A).
Inequalities (5.14) are equivalent to $\mathbb{M}\nu < \mathbf{0}_M$ where $\nu = \mathbf{1}_M$. Then, from Lemma A.1, $\mathbb{M}$ is Hurwitz. From Lemma A.2, (5.14) implies that matrix $\Gamma = \mathbb{M} + \mathbb{I}_M$ is Schur.

For system $\Sigma_{[i]}$ in (1.3), when $u_{[i]}$ is defined as in (5.4), $v_{[i]} = \mathbf{0}_{m_i}$ and $\bar{x}_{[i]} = \mathbf{0}_{n_i}$, we have

$$x_{[i]}(t) = F_i^t x_{[i]}(0) + \sum_{k=0}^{t-1} F_i^k \sum_{j \in \mathcal{N}_i} A_{ij} x_{[j]}(t-k-1). \tag{5.25}$$

In view of (5.25) we can write

$$||\mathcal{F}_i x_{[i]}(t)||_\infty \leq ||\mathcal{F}_i F_i^t \mathcal{F}_i^\flat||_\infty ||\mathcal{F}_i x_{[i]}(0)||_\infty + \sum_{j \in \mathcal{N}_i} \gamma_{ij} \max_{k \leq t} ||\mathcal{F}_j x_{[j]}(k)||_\infty.$$

where γ_{ij} are the entries of Γ. Denoting $\tilde{x}_{[i]} = \mathcal{F}_i x_{[i]}$, we can collectively define $\tilde{\mathbf{x}} = \tilde{\mathcal{F}}\mathbf{x}$, where $\tilde{\mathcal{F}} = \text{diag}(\mathcal{F}_1, \dots, \mathcal{F}_M)$. From the definition of sets $\mathbb{X}_i$, we have $\text{rank}(\tilde{\mathcal{F}}) = n$. We define the system

$$\tilde{\mathbf{x}}^+ = (\tilde{\mathbf{A}} + \tilde{\mathbf{B}}\tilde{\mathbf{K}})\tilde{\mathbf{x}} \tag{5.26}$$

where $\tilde{\mathbf{A}} = \tilde{\mathcal{F}}\mathbf{A}\tilde{\mathcal{F}}^\flat$, $\tilde{\mathbf{B}} = \tilde{\mathcal{F}}\mathbf{B}$ and $\tilde{\mathbf{K}} = \mathbf{K}\tilde{\mathcal{F}}^\flat$. In order to analyze the stability of the origin of (5.26), we consider the method proposed in [DRW07]. In view of Corollary 16 in [DRW07], the overall system (5.26) is asymptotically stable if the gain matrix Γ is Schur. As shown above this property is implied by (5.14).
Moreover, system (5.26) is an expansion of the original system (see Chapter 3.4 in [Lun92]). In view of the inclusion principle (see Theorem 3.3 in [Lun92] and [Sta04] for a discrete-time version), the asymptotic stability of (5.26) implies the asymptotic stability of the original system. □

Proof of (II)

Proof. First note that, for $i \in \mathcal{M}$, in view of (5.2) $||f_{i,\tau}^T \Xi_i||_\infty = 1$ for all $\tau \in 1 : \bar{\tau}_i$ and therefore $||\mathcal{F}_i \Xi_i||_\infty = 1$. This implies that $||f_{i,\tau}^T F_i^k A_{ij} \Xi_j||_\infty \leq ||f_{i,\tau}^T F_i^k A_{ij} \mathcal{F}_j^\flat||_\infty ||\mathcal{F}_j \Xi_j||_\infty = ||f_{i,\tau}^T F_i^k A_{ij} \mathcal{F}_j^\flat||_\infty \leq ||\mathcal{F}_i F_i^k A_{ij} \mathcal{F}_j^\flat||_\infty$. Therefore, in view of (5.14), for all $\tau \in 1 : \bar{\tau}_i$

$$\sum_{k=0}^{\infty} \sum_{j \in \mathcal{N}_i} ||f_{i,\tau}^T F_i^k A_{ij} \Xi_j||_\infty \leq \sum_{k=0}^{\infty} \sum_{j \in \mathcal{N}_i} ||\mathcal{F}_i F_i^k A_{ij} \mathcal{F}_j^\flat||_\infty < 1.$$

Now we want to find $\hat{l}_i > 0$ such that, simultaneously, the inclusion (5.11a) holds and $\mathbb{Z}_i(\delta_i)$ is a δ_i-outer approximation of the mRPI $\underline{\mathbb{Z}}_i$. The mRPI for (5.5) is given by [RKKM05]

$$\underline{\mathbb{Z}}_i = \bigoplus_{k=0}^{\infty} F_i^k \bigoplus_{j \in \mathcal{N}_i} A_{ij} \mathbb{X}_j. \tag{5.27}$$

From [RKKM05], for given $\delta_i > 0$ there exist $\gamma_i \in [0,1)$ and $s_i \in \mathbb{N}_+$ such that the set

$$\mathbb{Z}_i(\delta_i) = (1-\gamma_i)^{-1} \bigoplus_{k=0}^{s_i-1} F_i^k \bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{X}_j$$

is a δ_i-outer approximation of the mRPI $\underline{\mathbb{Z}}_i$.
Define $\bar{\mathbb{X}}_i = \hat{\mathbb{X}}_i \oplus \mathbb{Z}_i(\delta_i)$. Following the proof of Proposition 2 in [FS12] and using arguments from Section 3 of [KG98], we can then guarantee (5.11a) if $\bar{\mathbb{X}}_i \subseteq \mathbb{X}_i$, which holds if, for all $\tau \in 1:\bar{\tau}_i$

$$\sup_{\substack{z_{[i]}\in\mathbb{Z}_i(\delta_i)\\ \hat{x}_{[i]}\in\hat{\mathbb{X}}_i}} f_{i,\tau}^T(z_{[i]}+\hat{x}_{[i]}) \leq 1. \tag{5.28}$$

Using (A.3) and (5.27), the inequalities (5.28) are verified if

$$\sup_{\substack{\{x_{[j]}(k)\in\mathbb{X}_j\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty}\\ \hat{x}_{[i]}\in\hat{\mathbb{X}}_i\\ \sigma_i\in B_{\delta_i}(\mathbf{0}_{n_i})}} h_{i,\tau}^x(\{x_{[j]}(k)\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty},\hat{x}_{[i]}) + ||f_{i,\tau}^T\sigma_i||_\infty \leq 1 \tag{5.29}$$

where $h_{i,\tau}^x(\cdot) = f_{i,\tau}^T(\sum_{k=0}^\infty F_i^k \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]}(k) + \hat{x}_{[i]})$.
Since $||f_{i,\tau}^T\sigma_i||_\infty \leq ||f_{i,\tau}^T||_\infty\delta_i$, conditions (5.29) are satisfied if

$$\sup_{\substack{\{x_{[j]}(k)\in\mathbb{X}_j\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty}\\ \hat{x}_{[i]}\in\hat{\mathbb{X}}_i}} h_{i,\tau}^x(\{x_{[j]}(k)\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty},\hat{x}_{[i]}) \leq 1 - ||f_{i,\tau}^T||_\infty\delta_i. \tag{5.30}$$

Using (5.2) and (5.8) we can rewrite (5.30) as

$$\sup_{\substack{\{||\digamma_j(k)||_\infty\leq 1\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty}\\ ||\hat{\digamma}_i||_\infty\leq\hat{l}_i}} h_{i,\tau}^d(\{\digamma_j(k)\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty},\hat{\digamma}_i) \leq 1 - ||f_{i,\tau}^T||_\infty\delta_i \tag{5.31}$$

where $h_{i,\tau}^d(\cdot) = f_{i,\tau}^T(\sum_{k=0}^\infty F_i^k \sum_{j\in\mathcal{N}_i} A_{ij}\Xi_j\digamma_j(k) + \hat{\Xi}_i\hat{\digamma}_i)$.
The inequalities (5.31) are satisfied if

$$\sum_{k=0}^{\infty}\sum_{j\in\mathcal{N}_i} ||f_{i,\tau}^T F_i^k A_{ij}\Xi_j||_\infty + ||f_{i,\tau}^T\hat{\Xi}_i||_\infty\hat{l}_i \leq 1 - ||f_{i,\tau}^T||_\infty\delta_i \tag{5.32}$$

for all $\tau \in 1:\bar{\tau}_i$.
In view of (5.15) there exist sufficiently small $\delta_i > 0$ and $\hat{l}_i > 0$ satisfying (5.32) (and therefore verifying (5.11a)), e.g. choosing $\hat{l}_i \in (0,\hat{L}_i]$. □

Proof of (III)

Proof. For each $i \in \mathcal{M}$, we want to find tightened input constraint $\mathbb{V}_i$ such that (5.11b) holds. Following the rational used in Section 3 of [KG98], from definition of sets $\mathbb{U}_i$ and $\mathbb{V}_i$, (5.11b) holds if (5.17) is satisfied. Hence, choosing $\mathbb{V}_i$ as in (5.9), for $l_{v_{i,\tau}} = \hat{l}_{v_{i,\tau}}(\delta_i)$ the inclusion (5.11b) holds. □

Chapter 6

Plug-and-play MPC based on robust control invariant sets

Contents

6.1 Introduction

As in Chapter 5, we propose a PnP design procedure hinging on the notion of tube MPC [RM05] for handling coupling among subsystems, and aim at stabilizing the origin of the whole closed-loop system while guaranteeing satisfaction of constraints on local inputs and states. However, we advance the design procedure in Chapter 5 in several directions. First, in Chapter 5 the most critical step in the design of local MPC controllers requires the solution to nonlinear optimization problems. In this Chapter, using local tube MPC regulators based on Robust Control Invariant (RCI) sets, we guarantee overall stability and constraints satisfaction solving LP problems only. Second, in Chapter 5 stability requirements where fulfilled imposing an aggregate sufficient small-gain condition for networks. In this Chapter, we resort instead to set-based conditions that are usually less conservative. Third, while methods in Chapter 5 were tailored to decentralized control only, the new PnP-DeMPC can also admit a distributed implementation tacking advantage of pieces of information transmitted online from parent subsystems to their children. As for any decentralized synthesis procedure our method involves some degree of conservativeness [BL88] and its potential application to real-world systems will be discussed through examples. In particular, as in Chapter 5, we present an application of PnP-DeMPC to frequency control in power networks. Furthermore we highlight computational advantages brought about by our method by considering the control of a large array of masses connected by springs and dampers.
The chapter is structured as follows. The design of decentralized controllers is introduced in Section 6.2 with a focus on the assumptions needed for guaranteeing asymptotic stability of the origin and constraint satisfaction. In Section 6.3 we discuss how to design local controllers in a distributed fashion by solving LP problems and in Section 6.4 we describe PnP operations. In Section 6.5 we show how to enhance the control scheme taking advantage of pieces of information received from parents. In Section 6.6 we present applicative examples and Section 6.7 is devoted to concluding remarks.

6.2 Decentralized tube-based MPC of linear systems

We consider a large-scale discrete-time LTI system

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{6.1}$$

composed of M subsystems, in accordance with the notation introduced in Section 1.5. In this chapter we consider subsystems equipped with state and input constraints (see Section 1.5). The next assumption characterizes the shape of constraints $\mathbb{X}_i$ and $\mathbb{U}_i$, $i \in \mathcal{M}$.

Assumption 6.1. *Constraints $\mathbb{X}_i$ and $\mathbb{U}_i$ are PC-sets.*

From Assumption 6.1 constraints $\mathbb{X}_i$ and $\mathbb{U}_i$ are polytopes given by

$$\mathbb{X}_i = \{x_{[i]} \in \mathbb{R}^{n_i} : c_{x_{i,\tau}}^T x_{[i]} \leq 1,\ \forall \tau \in 1 : \tau_i^x\} \tag{6.2a}$$

$$\mathbb{U}_i = \{u_{[i]} \in \mathbb{R}^{m_i} : c_{u_{i,\tau}}^T u_{[i]} \leq 1,\ \forall \tau \in 1 : \tau_i^u\}, \tag{6.2b}$$

where $c_{x_{i,\tau}} \in \mathbb{R}^{n_i}$ and $c_{u_{i,\tau}} \in \mathbb{R}^{m_i}$.

6.2.1 Decentralized efficient tube-based MPC

In this section we propose a decentralized controller for (6.1) guaranteeing asymptotic stability of the origin of the closed-loop system and constraint satisfaction.
In the spirit of optimized tube-based control [RM05], we treat $w_{[i]}$ in (1.3) as a disturbance and define the nominal system $\hat{\Sigma}_{[i]}$ as

$$\hat{\Sigma}_{[i]} : \quad \hat{x}_{[i]}^+ = A_{ii}\hat{x}_{[i]} + B_i v_{[i]} \tag{6.3}$$

where $v_{[i]}$ is the input. Note that (6.3) has been obtained from (1.3a) by neglecting the disturbance term $w_{[i]}$.
As in [RM05] our goal is to relate inputs $v_{[i]}$ in (6.3) to $u_{[i]}$ in (1.3a) and compute sets $\mathbb{Z}_i \subseteq \mathbb{R}^n$, $i \in \mathcal{M}$ such that

$$x_{[i]}(0) \in \hat{x}_{[i]}(0) \oplus \mathbb{Z}_i \Rightarrow x_{[i]}(t) \in \hat{x}_{[i]}(t) \oplus \mathbb{Z}_i,\ \forall t \geq 0.$$

In other words, we want to confine $x_{[i]}(t)$ in a tube around $\hat{x}_{[i]}(t)$ of section $\mathbb{Z}_i$.
To achieve our aim, we define the set $\mathbb{Z}_i$, $\forall i \in \mathcal{M}$ as an RCI set for the constrained system (1.3a), with respect to the disturbance $w_i \in \mathbb{W}_i = \bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{X}_j$. From the definition of RCI set, we have that if $x_{[i]} \in \mathbb{Z}_i$, then there exist $u_{[i]} = \bar{\kappa}_i(x_{[i]}) : \mathbb{Z}_i \to \mathbb{U}_i$ such that $x_{[i]}^+ \in \mathbb{Z}_i$, $\forall w_{[i]} \in \mathbb{W}_i$. Note that, by construction, one has $\mathbb{Z}_i \subseteq \mathbb{X}_i$ and therefore the RCI set $\mathbb{Z}_i$ could not exist if $\mathbb{W}_i$ is "too big", i.e. $\mathbb{W}_i \supseteq \mathbb{X}_i$. Moreover if $x_{[i]} \in \hat{x}_{[i]} \oplus \mathbb{Z}_i$ and one uses the controller

$$\mathcal{C}_{[i]} : \quad u_{[i]} = v_{[i]} + \bar{\kappa}_i(x_{[i]} - \bar{x}_{[i]}) \tag{6.4}$$

then for all $v_{[i]} \in \mathbb{R}^{m_i}$, one has $x^+_{[i]} \in \hat{x}^+_{[i]} \oplus \mathbb{Z}_i$.
The next goal is to compute tightened constraints $\hat{\mathbb{X}}_i \subseteq \mathbb{X}_i$ and input constraints $\mathbb{V}_i \subseteq \mathbb{U}_i$ guaranteeing that

$$\hat{x}_{[i]}(t) \in \hat{\mathbb{X}}_i,\ v_{[i]}(t) \in \mathbb{V}_i,\ \forall i \in \mathcal{M} \Rightarrow \mathbf{x}(t+1) \in \mathbb{X},\ \mathbf{u}(t) \in \mathbb{U}.$$

To this purpose, we introduce the following assumption.

Assumption 6.2. *There exist $\rho_{i,1} > 0$, $\rho_{i,2} > 0$ such that $\mathbb{Z}_i \oplus B_{\rho_{i,1}}(\mathbf{0}_{n_i}) \subseteq \mathbb{X}_i$ and $\mathbb{U}_{z_i} \oplus B_{\rho_{i,2}}(\mathbf{0}_{m_i}) \subseteq \mathbb{U}_i$, where $B_{\rho_{i,1}}(\mathbf{0}_{n_i}) \subset \mathbb{R}^{n_i}$ and $B_{\rho_{i,2}}(\mathbf{0}_{m_i}) \subset \mathbb{R}^{m_i}$ and $\mathbb{U}_{z_i} = \bar{\kappa}_i(\mathbb{Z}_i)$.*

Assumption 6.2 implies that the coupling of subsystems connected in a cyclic fashion must be sufficiently small. As an example, for two subsystems $\Sigma_{[1]}$ and $\Sigma_{[2]}$ where each one is parent of the other one, Assumption 6.2 implies that $\mathbb{Z}_1 \subseteq \mathbb{X}_1$ and $\mathbb{Z}_2 \subseteq \mathbb{X}_2$. Since, by construction, $\mathbb{Z}_i \supseteq \mathbb{W}_i$, one has $A_{21}\mathbb{X}_1 \subseteq \mathbb{X}_2$ and $A_{12}\mathbb{X}_2 \subseteq \mathbb{X}_1$ that implies $A_{12}A_{21}\mathbb{X}_1 \subseteq \mathbb{X}_1$. These conditions are similar to the ones arising in the small gain theorem for networks [DRW07].
If Assumption 6.2 is verified, there are constraint sets $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$, $i \in \mathcal{M}$, that verify

$$\hat{\mathbb{X}}_i \oplus \mathbb{Z}_i \subseteq \mathbb{X}_i \tag{6.5a}$$
$$\mathbb{V}_i \oplus \mathbb{U}_{z_i} \subseteq \mathbb{U}_i. \tag{6.5b}$$

Similarly to Section 5.2.1 in Chapter 5, under Assumptions 1.1, 6.1 and 6.2, we set in (6.4)

$$v_{[i]}(t) = \kappa_i(x_{[i]}(t)) = v_{[i]}(0|t), \quad \bar{x}_{[i]}(t) = \eta_i(x_{[i]}(t)) = \hat{x}_{[i]}(0|t) \tag{6.6}$$

where $v_{[i]}(0|t)$ and $\hat{x}_{[i]}(0|t)$ are optimal values of variables $v_{[i]}(0)$ and $\hat{x}_{[i]}(0)$, respectively, appearing in the MPC-i problem (5.13), that is solved at time t, where in (5.13b) we substitute the RPI set $\mathbb{Z}_i(\delta_i)$ with the RCI set $\mathbb{Z}_i$, therefore we have the constraint

$$x_{[i]}(t) - \hat{x}_{[i]}(0) \in \mathbb{Z}_i. \tag{6.7}$$

Moreover, in order to design a stabilizing MPC-i controller, Assumption 5.3 must be fulfilled. With abuse of notation, we will refer the MPC-i controller using equation (5.13). In summary, the controller $\mathcal{C}_{[i]}$ is given by (6.4), (6.6) and (5.13) and depends upon quantities of system $\Sigma_{[i]}$ only. Therefore the collective controller for (6.1) is decentralized. The main problem that still has to be solved in the design of local controllers is the following one.

Problem 6.1 ($\mathcal{P}$)**.** Compute RCIs $\mathbb{Z}_i$, $i \in \mathcal{M}$ for (1.3), if they exist, verifying Assumption 6.2.

In the next section, we show how to solve it in a distributed fashion with efficient computations under Assumption 1.1 and 6.1. In this case, we will also show how sets $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$ verifying (6.5a) and (6.5b) can be readily computed.

Remark 6.1. In this section, as in Chapter 5, we introduced a DeMPC scheme based on tube-based control. In Chapter 5, using the robust control scheme proposed in [MSR05], we set $\bar{\kappa}_i(\cdot)$ as a linear function, i.e. $\bar{\kappa}_i(x_{[i]} - \hat{x}_{[i]}) = K_i(x_{[i]} - \hat{x}_{[i]})$. This choice has the disadvantage of requiring the computation of matrices K_i, $i \in \mathcal{M}$, fulfilling a global stability assumption. Differently, in the next section, using the control scheme proposed in [RM05], we will guarantee the overall stability for the closed-loop system through a suitable local computation of sets $\mathbb{Z}_i$.

6.3 Decentralized synthesis of DeMPC

Through the procedure proposed in [RB10], we can compute an RCI set $\mathbb{Z}_i \subset \mathbb{X}_i$ using an appropriate parametrization. As in Section VI of [RB10], we define $\forall i \in \mathcal{M}$ the set of variables θ_i as

$$
\begin{aligned}
\theta_i = \{\bar{z}_{[i]}^{(s,f)} &\in \mathbb{R}^{n_i} && \forall s \in \mathcal{A}_i^5,\ \forall f \in \mathcal{A}_i^1; && (6.8a)\\
\bar{u}_{[i]}^{(s,f)} &\in \mathbb{R}^{m_i} && \forall s \in \mathcal{A}_i^3,\ \forall f \in \mathcal{A}_i^1; && (6.8b)\\
\rho_i^{(f_1,f_2)} &\in \mathbb{R} && \forall f_1 \in \mathcal{A}_i^1,\ \forall f_2 \in \mathcal{A}_i^1; && (6.8c)\\
\psi_i^{(r,s)} &\in \mathbb{R} && \forall r \in \mathcal{A}_i^2,\ \forall s \in \mathcal{A}_i^3; && (6.8d)\\
\gamma_i^{(r,s)} &\in \mathbb{R} && \forall r \in \mathcal{A}_i^4,\ \forall s \in \mathcal{A}_i^3; && (6.8e)\\
\alpha_i &\in \mathbb{R}\} && && (6.8f)
\end{aligned}
$$

with $\mathcal{A}_i^1 = 1 : q_i$, $\mathcal{A}_i^2 = 1 : \tau_i^u$, $\mathcal{A}_i^3 = 0 : k_i - 1$, $\mathcal{A}_i^4 = 1 : \tau_i^x$ and $\mathcal{A}_i^5 = 0 : k_i$, where k_i, $q_i \in \mathbb{N}$ are parameters of the procedure that can be chosen by the user as well as the set

$$\bar{\mathbb{Z}}_i^0 = \text{convh}(\{\bar{z}_{[i]}^{(0,f)} \in \mathbb{R}^{n_i},\ \forall f \in \mathcal{A}_i^1\}),\ \bar{z}_{[i]}^{(0,1)} = \mathbf{0}_{n_i}.$$

The following assumption is needed to compute the RCI set $\mathbb{Z}_i$.

Assumption 6.3. *The set* $\bar{\mathbb{Z}}_i^0$ *is such that there is* $\omega_i > 0$ *verifying* $\mathbb{W}_i \oplus B_{\omega_i}(\mathbf{0}_{n_i}) \subseteq \bar{\mathbb{Z}}_i^0$.

We highlight that, in view of Assumption 6.1, the set $\mathbb{W}_i$ contains the origin in its nonempty relative interior. Hence, under Assumption 6.3, the set $\bar{\mathbb{Z}}_i^0$ also contains the origin in its nonempty interior.
Let us define the sets

$$\bar{\mathbb{Z}}_i^s = \text{convh}(\{\bar{z}_{[i]}^{(s,f)} \in \mathbb{R}^{n_i},\ \forall f \in \mathcal{A}_i^1\}),\ \forall s \in \mathcal{A}_i^5,\ \bar{z}_{[i]}^{(s,1)} = \mathbf{0}_{n_i},$$
$$\bar{\mathbb{U}}_{z_i}^s = \text{convh}(\{\bar{u}_{[i]}^{(s,f)} \in \mathbb{R}^{m_i},\ \forall f \in \mathcal{A}_i^1\}),\ \forall s \in \mathcal{A}_i^3,\ \bar{u}_{[i]}^{(s,1)} = \mathbf{0}_{m_i}.$$

Consider the following set of affine constraints on the decision variable θ_i

$$\Theta_i = \{\theta_i :$$
$$\alpha_i < 1,\ -\alpha_i \leq 0 \tag{6.9a}$$
$$z_{[i]}^{(k_i,f_1)} = \sum_{f_2=1}^{q_i} \rho_i^{(f_1,f_2)} z_{[i]}^{(0,f_2)} \qquad \forall f_1 \in \mathcal{A}_i^1; \tag{6.9b}$$
$$-\alpha_i + \sum_{f_2=1}^{q_i} \rho_i^{(f_1,f_2)} \leq 0 \qquad \forall f_1 \in \mathcal{A}_i^1; \tag{6.9c}$$
$$-\rho_i^{(f_1,f_2)} \leq 0 \qquad \forall f_1 \in \mathcal{A}_i^1,\ \forall f_2 \in \mathcal{A}_i^1; \tag{6.9d}$$
$$\sum_{s=0}^{k_i-1} \psi_i^{(\tau,s)} + \alpha_i < 1 \qquad \forall \tau \in \mathcal{A}_i^2; \tag{6.9e}$$
$$c_{u_{i,\tau}}^T \bar{u}_{[i]}^{(s,f)} \leq \psi_i^{(\tau,s)} \qquad \forall \tau \in \mathcal{A}_i^2, \forall s \in \mathcal{A}_i^3, \forall f \in \mathcal{A}_i^1; \tag{6.9f}$$
$$\sum_{s=0}^{k_i-1} \gamma_i^{(\tau,s)} + \alpha_i < 1 \qquad \forall \tau \in \mathcal{A}_i^4; \tag{6.9g}$$
$$c_{x_{i,\tau}}^T \bar{z}_{[i]}^{(s,f)} \leq \gamma_i^{(\tau,s)} \qquad \forall \tau \in \mathcal{A}_i^4, \forall s \in \mathcal{A}_i^3, \forall f \in \mathcal{A}_i^1; \tag{6.9h}$$
$$\bar{z}_{[i]}^{(s+1,f)} = A_{ii}\bar{z}_{[i]}^{(s,f)} + B_i\bar{u}_{[i]}^{(s,f)} \qquad \forall s \in \mathcal{A}_i^3, \forall f \in \mathcal{A}_i^1\ \}. \tag{6.9i}$$

The relation between elements of Θ_i and the RCI sets in Problem $\mathcal{P}$ is established in the next proposition.

Proposition 6.1. *Let Assumptions 1.1 and 6.3 hold and sets $\mathbb{X}_i$ and $\mathbb{U}_i$ be defined as in (6.2a) and (6.2b) respectively. Let $k_i > 0$. If there exist an admissible $\theta_i \in \Theta_i$, then*

$$\mathbb{Z}_i = (1-\alpha_i)^{-1} \bigoplus_{s=0}^{k_i-1} \bar{\mathbb{Z}}_i^s \subset \mathbb{X}_i \tag{6.10}$$

is an RCI set and the corresponding set $\mathbb{U}_{z_i}$ *is given by*

$$\mathbb{U}_{z_i} = (1-\alpha_i)^{-1} \bigoplus_{s=0}^{k_i-1} \bar{\mathbb{U}}_{z_i}^s \subset \mathbb{U}_i. \tag{6.11}$$

Proof. In Section 3 of [RB10], the authors prove that set $\mathbb{Z}_i$ defined as in (6.10) is an RCI set and that the inclusions in (6.5a) and (6.5b) hold. □

Remark 6.2. Under Assumption 6.1 the feasibility problem (6.9) is an LP problem, since the constraints in Θ_i are affine. In [RB10] the authors propose to find $\theta_i \in \Theta_i$ while minimizing different cost functions under constraints Θ_i in order to achieve different aims. In our context the most important goal is the minimization of α_i that corresponds to the minimization of the size of the set $\mathbb{Z}_i$. We also note that the inclusion of $\mathbf{0}_{n_i}$ in the definition of sets $\bar{\mathbb{Z}}_i^s$, $\forall s \in 0 : k_i$, ensures that $\bar{\mathbb{Z}}_i^s$ contains the origin and hence, under Assumption 6.3, $\mathbb{Z}_i$ contains the origin in its nonempty interior.

We highlight that the set of constraints Θ_i depends only upon local fixed parameters $\{A_{ii}, B_i, \mathbb{X}_i, \mathbb{U}_i\}$, fixed parameters $\{A_{ij}, \mathbb{X}_j\}_{j\in\mathcal{N}_i}$ of parents of $\hat{\Sigma}_{[i]}$ (because from Assumption 6.3 the set $\bar{\mathbb{Z}}_i^0$ must be defined in such a way that $\bar{\mathbb{Z}}_i^0 \supseteq \mathbb{W}_i = \bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{X}_j$) and local tunable parameters θ_i (the decision variables (6.8)). Moreover Θ_i does not depend on tunable parameters of parents. This implies that the computation of sets $\mathbb{Z}_i$ and $\mathbb{U}_{z_i}$ in (6.10) and (6.11) does not influence the choice of $\mathbb{Z}_j$ and $\mathbb{U}_{z_j}$, $j \neq i$ and therefore Problem $\mathcal{P}$ is decomposed in the following independent LP problems for $i \in \mathcal{M}$.

Problem 6.2 ($\mathcal{P}_i$)**.** Solve the feasibility LP problem $\theta_i \in \Theta_i$.

If Problem $\mathcal{P}_i$ is solved, then $\forall i \in \mathcal{M}$ we can compute sets $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$ in (6.5a) and (6.5b) as

$$\hat{\mathbb{X}}_i = \mathbb{X}_i \ominus \mathbb{Z}_i, \qquad \mathbb{V}_i = \mathbb{U}_i \ominus \mathbb{U}_{z_i}. \tag{6.12}$$

The overall procedure for the decentralized synthesis of local controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ is summarized in Algorithm 6.1 whose properties are summarized in the next proposition.

Proposition 6.2. *Under Assumption 1.1 and 6.1 if, for all* $i \in \mathcal{M}$, *controllers* $\mathcal{C}_{[i]}$ *are designed according to Algorithm 6.1, then also Assumptions 6.2, 5.3 and 6.3 are verified.*

Proof. Assumptions 5.3 and 6.3 are enforced in Steps (IIIiii) and (I) of Algorithm 6.1, respectively. As for Assumption 6.2, because of the inequality in (6.9e), constraints (6.9e)-(6.9f) guarantee the existence of $\rho_{i,2} > 0$ such that $\mathbb{U}_{z_i} \oplus B_{\rho_{i,2}}(\mathbf{0}_{m_i}) \subseteq \mathbb{U}_i$. Similarly, because of the inequality in (6.9g), one has that (6.9g) and (6.9h) imply the existence of $\rho_{i,1} > 0$ such that $\mathbb{Z}_i \oplus B_{\rho_{i,1}}(\mathbf{0}_{n_i}) \subseteq \mathbb{X}_i$. □

Algorithm 6.1 Design of controller $\mathcal{C}_{[i]}$ for subsystem $\Sigma_{[i]}$

Input: A_{ii}, B_i, $\mathbb{X}_i$, $\mathbb{U}_i$, $\mathcal{N}_i$, $\{A_{ij}\}_{j\in\mathcal{N}_i}$, $\{\mathbb{X}_j\}_{j\in\mathcal{N}_i}$, $k_i > 0$
Output: controller $\mathcal{C}_{[i]}$ given by (6.4), (6.6) and (5.13)

(I) Compute the set $\mathbb{W}_i = \bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{X}_j$ and choose $\bar{\mathbb{Z}}_i^0$ such that $\mathbb{X}_i \supseteq \bar{\mathbb{Z}}_i^0 \supseteq \mathbb{W}_i \oplus B_{\omega_i}(\mathbf{0}_{n_i})$ for a sufficiently small $\omega_i > 0$. If $\bar{\mathbb{Z}}_i^0$ does not exist **stop** (the controller $\mathcal{C}_{[i]}$ cannot be designed)

(II) Solve the feasibility LP problem $\theta_i \in \Theta_i$. If it is unfeasible **stop** (the controller $\mathcal{C}_{[i]}$ cannot be designed).

(III) Design controller MPC-i by

  (i) Computing $\mathbb{Z}_i$ as in (6.10) and $\mathbb{U}_{z_i}$ as in (6.11).

  (ii) Computing $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$ as in (6.12).

  (iii) Choosing $\ell_i(\cdot)$, $V_{f_i}(\cdot)$ and $\hat{\mathbb{X}}_{f_i}$ verifying Assumption 5.3.

(IV) Choose the function $\bar{\kappa}_i$ in (6.4).

If in Step (II) of Algorithm 6.1 the LP problem is infeasible, we can restart the Algorithm with a different k_i. However the existence of a parameter k_i such that the LP problem is feasible is not guaranteed [RB10]. Steps (IIIi), (IIIii) and (IIIiii) of Algorithm 6.1, that provide constraints appearing in the MPC-i problem (5.13), are the most computationally expensive ones because they involve Minkowski sums and differences of polytopic sets. Next, we show how to avoid burdensome computations.

6.3.1 Implicit representation of sets $\mathbb{Z}_i$ and $\mathbb{U}_{z_i}$

In this section we show how to rewrite constraint (6.7) by exploiting the implicit representation of set $\mathbb{Z}_i$ proposed in Section VI.B of [RB10]. Recalling that $\mathbb{Z}_i$ is the Minkowski sum of k_i polytopes and that, for all $s \in 0 : k_i - 1$, polytope $\bar{\mathbb{Z}}_i^s$ is described by the convex combination of points $\bar{z}_{[i]}^{(s,f)}$, we have

$$\tilde{z}_{[i]}^s \in \bar{\mathbb{Z}}_i^s \quad \text{if } \forall f \in 1:q_i, \exists \beta_i^{(s,f)} \geq 0$$
$$\text{such that } \sum_{f=1}^{q_i} \beta_i^{(s,f)} = 1 \text{ and } \tilde{z}_{[i]}^s = \sum_{f=1}^{q_i} \beta_i^{(s,f)} \bar{z}_{[i]}^{(s,f)}.$$

Hence we have that $x_{[i]}(t) - \hat{x}_{[i]}(0|t) \in \mathbb{Z}_i$ if and only if $\forall f \in 1:q_i, \forall s \in 0:k_i-1$ there exist $\beta_i^{(s,f)} \in \mathbb{R}$ such that

$$\beta_i^{(s,f)} \geq 0 \tag{6.13a}$$
$$\sum_{f=1}^{q_i} \beta_i^{(s,f)} = 1 \tag{6.13b}$$
$$x_{[i]}(t) - \hat{x}_{[i]}(0|t) = (1-\alpha_i)^{-1} \sum_{s=0}^{k_i-1} \sum_{f=1}^{q_i} \beta_i^{(s,f)} \bar{z}_{[i]}^{(s,f)}. \tag{6.13c}$$

In other words we add to the optimization problem (5.13) the variables $\beta_i^{(s,f)}$ and replace (6.7) with constraints (6.13a)-(6.13c).
With similar arguments, we can also provide an implicit representation of sets $\mathbb{U}_{z_i}$. In particular, we have that $u_{z[i]} \in \mathbb{U}_{z_i}$ if and only if $\forall f \in 1:q_i, \forall s \in 0:k_i-1$ there exist $\phi_i^{(s,f)} \in \mathbb{R}$ such that

$$\phi_i^{(s,f)} \geq 0$$
$$\sum_{f=1}^{q_i} \phi_i^{(s,f)} = 1$$
$$u_{z[i]} = (1-\alpha_i)^{-1} \sum_{s=0}^{k_i-1} \sum_{f=1}^{q_i} \phi_i^{(s,f)} \bar{u}_{[i]}^{(s,f)}.$$

6.3.2 Computation of sets $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$

In this section we show how to compute sets $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$ in (6.12) using the implicit representation of $\mathbb{Z}_i$ and $\mathbb{U}_{z_i}$.
Using (6.10) we can rewrite $\hat{\mathbb{X}}_i = \mathbb{X}_i \ominus (1-\alpha_i)^{-1} \bigoplus_{s=0}^{k_i-1} \bar{\mathbb{Z}}_i^s$. Recalling that $\bar{\mathbb{Z}}_i^s, \forall s \in 0:k_i-1$ are defined as the convex hull of points $\bar{z}_{[i]}^{(s,f)}$, $f \in 1:q_i$, we can compute the set $\hat{\mathbb{X}}_i$ using Algorithm 6.2.

Algorithm 6.2 Computation of the set $\hat{\mathbb{X}}_i$

Input: set $\mathbb{X}_i$ defined as in (6.2a), points $\bar{z}_{[i]}^{(s,f)}, \forall s \in 0 : k_i - 1, \forall f \in 1 : q_i$ and scalar α_i.
Output: set $\hat{\mathbb{X}}_i$.

(I) $\bar{C}_i = (c_{x_{i,1}}^T, \ldots, c_{x_{i,\tau_i^x}}^T) \in \mathbb{R}^{\tau_i^x \times n_i}$ and $\bar{D}_i = \mathbf{1}_{\tau_i^x}$

(II) **For each** $s \in 0 : k_i - 1$

  (i) **For each** $f \in 1 : q_i$
  $\tilde{C}_i = (\bar{C}_i, \bar{C}_i)$ and $\tilde{D}_i = (\bar{D}_i, \bar{D}_i - (1 - \alpha_i)^{-1} \bar{C}_i \bar{z}_{[i]}^{(s,f)})$

  (ii) Remove redundant constraints from $\tilde{C}_i \hat{x}_{[i]} \leq \tilde{D}_i$ so obtaining the inequalities $\bar{C}_i \hat{x}_{[i]} \leq \bar{D}_i$

(III) Set $\hat{\mathbb{X}}_i = \{\hat{x}_{[i]} : \bar{C}_i \hat{x}_{[i]} \leq \bar{D}_i\}$ where $\bar{C}_i \in \mathbb{R}^{\hat{\tau}_i^x \times n_i}$ and $\bar{D}_i \in \mathbb{R}^{\hat{\tau}_i^x}$

In particular, the operation in Step (IIii) amounts to solve suitable LP problems. We can compute $\mathbb{V}_i$ using the implicit representation of $\mathbb{U}_{z_i}$ in a similar way. Indeed it suffices to use Algorithm 6.2 replacing $\mathbb{X}_i$ with $\mathbb{U}_i$ defined in (6.2b) and points $\bar{z}_{[i]}^{(s,f)}$ with points $\bar{u}_{[i]}^{(s,f)}, \forall s \in 0 : k_i - 1, \forall f \in 1 : q_i$.

6.3.3 Evaluation of control law $\bar{\kappa}_i(\cdot)$

In Section 6.2, we introduced local controllers $\mathcal{C}_{[i]}$. Note that in (6.4) the control law $u_{[i]}$ is composed by the term $v_{[i]}$, that is computed by solving the local MPC-i problem (5.13), and the term $\bar{\kappa}(z_{[i]})$ with $z_{[i]} = x_{[i]} - \hat{x}_{[i]}$. Since $\bar{\kappa}_i(\cdot)$ depends on $\hat{x}_{[i]}$, we need to solve the MPC-i problem (5.13) and then compute $\bar{\kappa}_i(z_{[i]})$. The control law $\bar{\kappa}(z_{[i]}) \in \mathbb{U}_{z_i}$ guarantees that if $x_{[i]}(t) - \hat{x}_{[i]}(0|t) \in \mathbb{Z}_i$ (i.e. MPC-i problem (5.13) is feasible) then there is a $\lambda_i > 0$ such that $x_{[i]}(t+1) - \hat{x}_{[i]}(1|t) \in \lambda_i \mathbb{Z}_i$. To compute the control law $\bar{\kappa}_i(z_{[i]})$ one can use the methods proposed in [Bla91] or in [RB10]. In [Bla91] the authors propose to solve an LP problem in order to maximize the contractivity parameter λ_i, i.e. for a given $z_{[i]}$ we compute $\bar{\kappa}_i(z_{[i]}) \in \mathbb{U}_{z_i}$ by minimizing the scalar λ_i such that $A_{ii} z_{[i]} + B_i \bar{\kappa}_i(z_{[i]}) \in \lambda \mathbb{Z}_i \ominus \mathbb{W}_i$. In [RB10] the authors propose an implicit representation of controller $\bar{\kappa}_i(z_{[i]})$ based on the implicit representation (6.13) of set $\mathbb{Z}_i$. In our framework we want to take advantages of both approaches and compute the control law $\bar{\kappa}_i(\cdot)$ solving the following LP problem

$$\bar{\mathbb{P}}_i(z_{[i]}) : \min_{\mu, \beta_i^{(s,f)}} \mu \tag{6.14a}$$

$$\beta_i^{(s,f)} \geq 0, \qquad \forall f \in 1 : q_i, \forall s \in 0 : k_i - 1 \tag{6.14b}$$

$$\sum_{f=1}^{q_i} \beta_i^{(s,f)} = \mu \qquad \forall s \in 0 : k_i - 1 \tag{6.14c}$$

$$\mu \geq 0 \tag{6.14d}$$

$$z_{[i]} = (1 - \alpha_i)^{-1} \sum_{s=0}^{k_i - 1} \sum_{f=1}^{q_i} \beta_i^{(s,f)} \bar{z}_{[i]}^{(s,f)} \tag{6.14e}$$

and setting

$$\bar{\kappa}_i(z_{[i]}) = (1 - \alpha_i)^{-1} \sum_{s=0}^{k_i - 1} \bar{\kappa}_i^s(z_{[i]}), \text{ where } \bar{\kappa}_i^s(z_{[i]}) = \sum_{f=1}^{q_i} \bar{\beta}_i^{(s,f)} \bar{u}_{[i]}^{(s,f)} \tag{6.15}$$

where $\bar{\beta}_i^{(s,f)}$ are the optimizers to (6.14). Solving LP problem (6.14), we compute a control law $\bar{\kappa}_i(\cdot)$ that tries to keep the state $x_{[i]}$ and the nominal state $\hat{x}_{[i]}$ as close as possible. According to [Gal95] we can assume without loss of generality that $\bar{\kappa}_i(\cdot)$ is a continuous piecewise affine map. Note that since $\bar{\mathbb{Z}}_i^0 \subseteq \mathbb{Z}_i$, if $z_{[i]} = \mathbf{0}_{n_i}$ no control action is needed in order to guarantee robust invariance. Indeed, in this case an optimal solution to (6.14) is $\mu = 0$ and $\beta_i^{(s,f)} = 0$, $\forall f \in 1 : q_i, \forall s \in 0 : k_i - 1$ and therefore $\bar{\kappa}_i(z_{[i]}) = 0$.

6.3.4 Analysis of the closed-loop system

Defining the collective variables

$$\bar{\mathbf{x}} = (\bar{x}_{[1]}, \ldots, \bar{x}_{[M]}) \in \mathbb{R}^n, \ \mathbf{v} = (v_{[1]}, \ldots, v_{[M]}) \in \mathbb{R}^m$$

and the function

$$\bar{\boldsymbol{\kappa}}(\mathbf{x}) = (\bar{\kappa}_1(x_{[1]}), \ldots, \bar{\kappa}_M(x_{[M]})) : \mathbb{R}^n \to \mathbb{R}^m,$$

from (1.3) and (6.4) one obtains the collective model

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{v} + \mathbf{B}\bar{\boldsymbol{\kappa}}(\mathbf{x} - \bar{\mathbf{x}}). \tag{6.16}$$

Definition 6.1. The feasibility region for the MPC-i problem is

$$\mathbb{X}_i^N = \{s_{[i]} \in \mathbb{X}_i : \quad (5.13) \text{ is feasible for } x_{[i]}(t) = s_{[i]}\}$$

and the collective feasibility region is

$$\mathbb{X}^N = \prod_{i\in\mathcal{M}} \mathbb{X}_i^N. \tag{6.17}$$

The next theorem summarizes the key properties of the closed-loop system (6.16).

Theorem 6.1. *Let Assumptions 1.1 and 6.1 hold. Assume controllers $\mathcal{C}_{[i]}$ in (6.4) are computed using Algorithm 6.1 and let the function $\bar{\kappa}_i$ be given by (6.15). Then, the origin of (6.16) is asymptotically stable, $\mathbb{X}^N$ is a region of attraction and $\mathbf{x}(0) \in \mathbb{X}^N$ guarantees constraints $\mathbf{x}(t) \in \mathbb{X}$ and $\mathbf{u}(t) \in \mathbb{U}$ are fulfilled at all time instants.*

Proof. The proof of Theorem 6.1 is given in Appendix 6.8.1. ☐

Remark 6.3. In Remark 6.1, we highlighted that the main difference with the PnP scheme proposed in Chapter 5 is the computation of sets $\mathbb{Z}_i$ and functions $\bar{\kappa}_i(\cdot)$, $\forall i \in \mathcal{M}$. We note that in Chapter 5, the computation of K_i and $\mathbb{Z}_i$ requires the solution to a nonlinear optimization problem. In this section, we have shown that for the PnP scheme proposed in Section 6.2, using results from [RB10], we can compute set $\mathbb{Z}_i$ and function $\bar{\kappa}_i(\cdot)$ solving LP problems only.

6.4 Plug-and-play operations

In this section we discuss the synthesis of new controllers and the redesign of existing ones when subsystems are added to or removed from system (1.3). The goal is to preserve stability of the origin and constraint satisfaction for the new closed-loop system. Note that plug in and out of subsystems are here considered as offline operations, i.e. they do not induce a switching dynamics. As a starting point, we consider a plant composed by subsystems $\Sigma_{[i]}$, $i \in \mathcal{M}$ equipped with local controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ produced by Algorithm 6.1. We also define $\mathcal{S}_k = \{k : i \in \mathcal{N}_k\}$ as the set of children of $\Sigma_{[k]}$.

6.4.1 Plug-in operation

Assume subsystem $\Sigma_{[M+1]}$, characterized by parameters $A_{M+1\ M+1}$, B_{M+1}, $\mathbb{X}_{M+1}$, $\mathbb{U}_{M+1}$, $\mathcal{N}_{M+1}$ and $\{A_{M+1\ j}\}_{j\in\mathcal{N}_{M+1}}$, is plugged in. For building the controller $\mathcal{C}_{[M+1]}$ we execute Algorithm 6.1 that needs information only from subsystems $\Sigma_{[j]}$, $j \in \mathcal{N}_{M+1}$. If there is no solution to the feasibility LP problem in Step (II) of Algorithm 6.1, we declare that $\Sigma_{[M+1]}$ cannot be plugged in. Since each subsystem $\Sigma_{[j]}$, $j \in \mathcal{S}_{M+1}$ has the new parent $\Sigma_{[M+1]}$, the set $\mathbb{W}_j$ gets bigger and the set $\bar{\mathbb{Z}}_j^0$ already computed could fail to verify the inclusions in Step (I) of Algorithm 6.1. In this case, the controller $\mathcal{C}_{[j]}$ must be redesigned. Again, if Algorithm 6.1 stops in Step (II), we declare that $\Sigma_{[M+1]}$ cannot be plugged in.
Note that redesign of controllers $\mathcal{C}_{[i]}$, $i \notin \{M+1\}\bigcup\mathcal{S}_{M+1}$ is not required in order to guarantee convergence to zero of the origin and constraint satisfaction for the new closed-loop system.

6.4.2 Unplugging operation

Assume that subsystem $\Sigma_{[k]}$, $k \in \mathcal{M}$ gets plug out. Since for each $i \in \mathcal{S}_k$ the set $\mathcal{N}_i$ gets smaller, also $\mathbb{W}_i$ gets smaller and the set $\bar{\mathbb{Z}}_i^0$ already computed still verifies the inclusions in Step (I) of Algorithm 6.1. This means that, for each $i \in \mathcal{S}_k$, the previously computed θ_i in Step (II) of Algorithm 6.1 still verifies $\theta_i \in \Theta_i$ and hence the controller $\mathcal{C}_{[i]}$ does not have to be redesigned. Also controllers $\mathcal{C}_{[j]}$, $j \notin \{k\}\bigcup\mathcal{S}_k$ do not have to be redesigned because sets $\mathcal{N}_j$ do not change. However, we highlight that since systems $\Sigma_{[i]}$, $i \in \mathcal{S}_k$ have one parent less, the redesign of controllers $\mathcal{C}_{[i]}$ through Algorithm 6.1 could improve the performance.

6.5 Distributed online implementation of $\mathcal{C}_{[i]}$

In Section 6.2, we introduced decentralized local controllers $\mathcal{C}_{[i]}$ that, using the nominal model (6.3) and local information only, can control system i without the knowledge of the state of the parents. However, our framework allows one to take advantage of information from parents systems without redesigning controllers $\mathcal{C}_{[i]}$.
If at time t the controller of system $\Sigma_{[i]}$ can receive the value of states $x_{[j]}(t), \forall j \in \mathcal{N}_i$ from parents, we can define the new controller $\mathcal{C}_{[i]}^{dis}$ as

$$\mathcal{C}_{[i]}^{dis}: \quad u_{[i]} = v_{[i]} + \bar{\kappa}_i^{dis}(x_{[i]} - \hat{x}_{[i]}, \{x_{[j]}\}_{j\in\mathcal{N}_i}). \tag{6.18}$$

In (6.18) the term $v_{[i]}$ is the same appearing in the controller $\mathcal{C}_{[i]}$ and is obtained by solving the MPC-i problem (5.13). Similarly to the control law $\bar{\kappa}_i(\cdot)$ in (6.4), the second term in (6.18) must guarantee robust invariance of the set $\mathbb{Z}_i$ and it can be computed by solving (6.14) with constraint (6.14e) replaced by

$$A_{ii}z_{[i]} + B_i u_{z[i]} + \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]} = (1-\alpha_i)^{-1}\sum_{s=0}^{k_i-1}\sum_{f=1}^{q_i}\beta_i^{(s,f)}\bar{z}_{[i]}^{(s,f)} \tag{6.19}$$

where $u_{z[i]} \in \mathbb{R}^{m_i}$ are additional optimization variables. The desired control term is then given by $\bar{\kappa}_i^{dis}(x_{[i]}-\hat{x}_{[i]}, \{x_{[j]}\}_{j\in\mathcal{N}_i}) = u_{z[i]}$. Note that constraint (6.19) allows us to compute $\bar{\kappa}_i^{dis}(\cdot)$ taking into account the real state of parents at time t. Using (6.5b) and (6.18), we can still guarantee input constraints (6.2b) adding the following constraints in the LP problem $\bar{\mathbb{P}}_i$ in (6.14)

$$c_{u_{i,\tau}}^T u_{z[i]} \leq 1 - c_{u_{i,\tau}}^T v_{[i]},\ \forall \tau \in 1:\tau_i^u.$$

We highlight that the LP problem (6.14) is feasible if and only if the new LP problem is feasible. In fact, using the definition of robust control invariance, the LP problem (6.14) is feasible if there exist $u_{z[i]} \in \mathbb{U}_{z_i}$ such that $z_{[i]}^+ = A_{ii}z_{[i]} + B_i u_{z[i]} + w_{[i]} \in \mathbb{Z}_i, \forall w_{[i]} \in \mathbb{W}_i$. The fact that $\sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]} \in \mathbb{W}_i$ guarantees the feasibility of both LP problems.

We show advantages of including information from parents through an example. Consider two dynamically coupled systems equipped with controllers synthesized using Algorithm 6.1 and assume $x_{[1]}(0) = \mathbf{0}_{n_1}$ and $x_{[2]}(0) \neq \mathbf{0}_{n_2} \notin \mathbb{Z}_2$. Without exchange of information, the solution to the MPC-i problem (5.13) is $v_{[1]}(0) = \mathbf{0}_{m_1}$ and $\hat{x}_{[1]}(0) = \mathbf{0}_{n_1}$ for the first subsystem and $v_{[2]}(0) \neq \mathbf{0}_{m_2}$ and $\hat{x}_{[2]}(0) \neq \mathbf{0}_{n_2}$ for the second subsystem, hence the solution of the LP problem (6.14) will be $\bar{\kappa}_1(z_{[1]}) = \mathbf{0}_{m_1}$ and $\bar{\kappa}_2(z_{[2]}) \neq \mathbf{0}_{m_2}$. This means we apply a control action to subsystem 2 only. However, $x_{[1]}(1) \neq \mathbf{0}_{n_1}$ because of coupling. Differently, solving the LP problem (6.14) with constraint (6.14e) replaced by (6.19), we obtain $\bar{\kappa}_1(z_{[1]}) \neq \mathbf{0}_{m_1}$ and $\bar{\kappa}_2(z_{[2]}) \neq \mathbf{0}_{m_2}$. Therefore, we apply a control action on both subsystems because subsystem 1 tries to counteract in advance coupling with subsystem 2.

6.6 Examples

In this section, we illustrate three examples.

1. A low-order system composed by the interconnection of two mass-spring-damper systems, allowing decentralized and distributed implementations of local controllers to be compared.

2. The PNSs proposed in Appendix B, where we compare the performance of the proposed controllers with CeMPC and with the PnP controllers proposed in Chapter 5 (PnP-DeMPC-rpi). Furthermore, we discuss PnP operations corresponding to the addition and removal of power generation areas;

3. A LSS composed by an array of 1024 mass-spring-damper systems.

All examples and simulations are implemented using the *PnPMPC-toolbox* for Matlab [RBFT12] (see Appendix C).

6.6.1 Comparison of DeMPC and DiMPC controllers

In this section, we compare the performance of controllers $\mathcal{C}_{[i]}$ and $\mathcal{C}_{[i]}^{dis}$. We consider the example illustrated in Figure 6.1.

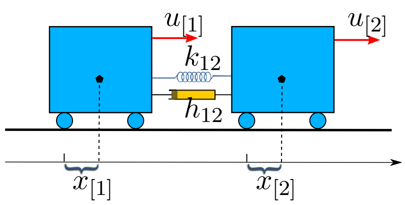


Figure 6.1: Example system.

The system is composed by two trucks coupled by a spring and a damper. Parameters values are: $m_1 = 2$, $m_2 = 4$, $k_{12} = 0.4$ and $h_{12} = 0.3$. Each truck $i \in \mathcal{M} = \{1, 2\}$, is a subsystem with state variables $x_{[i]} = (x_{[i,1]}, x_{[i,2]})$ and input $u_{[i]}$, where $x_{[i,1]}$ is the displacement of truck i with respect to a given equilibrium position, $x_{[i,2]}$ is the velocity of the truck i and $100u_{[i]}$ is a force applied to truck i. Subsystems are equipped with the state constraints $|x_{[i,1]}| \leq 4.5$, $|x_{[i,2]}| \leq 2$, $i \in \mathcal{M}$ and with the input constraints $|u_{[i]}| \leq 1.5$, $i \in \{1, 2\}$. We obtain models $\Sigma_{[i]}$ by discretizing the second order continuous-time system representing each truck with 0.1 sec sampling time, using exact discretization and treating $u_{[i]}$ and $x_{[j]}$, $j \in \mathcal{N}_i$ as exogenous signals [FCS13]. We synthesized controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ using Algorithm 6.1. At time t we compute the control action $u_{[i]}$ and apply it to the continuous-time system, keeping the value constant between time t and $t + 1$. We assume $x_{[1]}(0) = (0, 0)$ and $x_{[2]}(0) = (3, 0)$.

In Figure 6.2 we show the results obtained using controllers $\mathcal{C}_{[i]}$ and $\mathcal{C}^{dis}_{[i]}$ in the time interval from 0 to 0.3 sec. We note that for the controller $\mathcal{C}_{[1]}$, since $x_{[1]}(0) = \mathbf{0}_2$ one has $u_{[1]}(0) = 0$. Indeed the solutions to optimization problems (5.13) and (6.14) are $v_{[1]}(0) = 0$ and $\kappa_1(z_{[1]}(0)) = 0$. For the second truck the control action is $u_{[2]}(0) = -0.76$ because $x_{[2]}(0) \neq \mathbf{0}_2$. However, one has $x_{[1]}(1) \neq \mathbf{0}_2$ because of coupling. Using the distributed controller $\mathcal{C}^{dis}_{[1]}$, since $x_{[1]}(0) = \mathbf{0}_2$ and $x_{[2]}(0) \neq \mathbf{0}_2$, one has $u_{[1]}(0) = -0.012$. Indeed the solution to the LP problem (6.14), with (6.14e) replaced by (6.19), gives $\bar{\kappa}_1(z_{[1]}) \neq 0$. Figure 6.2a shows the position of each truck: we note that using controllers $\mathcal{C}^{dis}_{[i]}$, the position of the first truck does not change significantly because the controller tries to counteract in advance coupling with subsystem 2. This shows the benefits of a distributed implementation of local controllers. The state and input trajectories of the second truck are almost identical when using controllers $\mathcal{C}_{[i]}$ and $\mathcal{C}^{dis}_{[i]}$ because the state of the first truck is approximately zero.

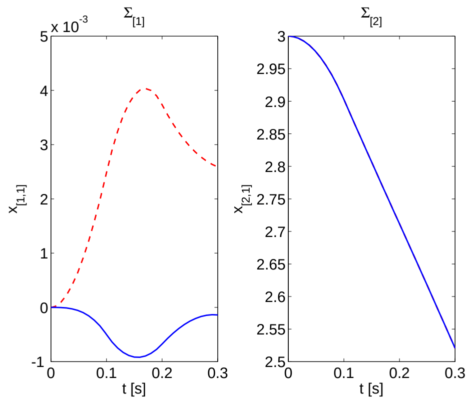
6.6.2 Power network system

In this section, we apply the proposed DeMPC and DiMPC schemes to the PNS proposed in Appendix B. We will show advantages brought about by PnP-DeMPC when generation areas are connected/disconnected to/from an existing network. In the following we first design the AGC layer for the PNS of Scenario 1 in B.1.1 and then we show how in presence of connection/disconnection of an area (Scenario 2 and 3, in Sections B.1.2 and B.1.2, respectively) the AGC can be redesigned via plug-in and unplugging operations[1].

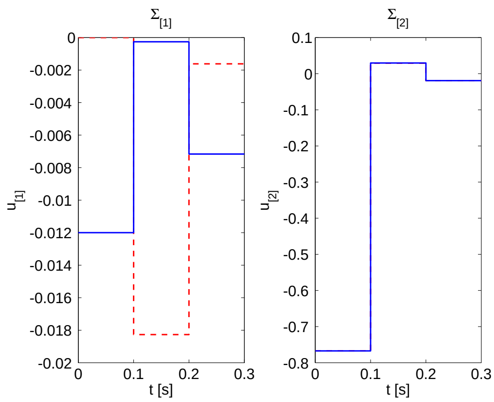
Scenario 1

We consider the PNS proposed in Section B.1.2. For each system $\Sigma_{[i]}$ we synthesize the controller $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ using Algorithm 6.1. Note that in Step (II) of Algorithm 6.1 only the feasibility of LP problem is required. Therefore the synthesis of controllers $\mathcal{C}_{[i]}$ is computationally more efficient than the nonlinear procedure proposed in Step (I) of Algorithm 5.1.
In Figure 6.3 we compare the performance of the proposed DeMPC scheme with the performance of the CeMPC controller. For CeMPC we consider the controller proposed in Section B.2. In the control experiment, step power loads ΔP_{L_i} specified in Table B.3 of Appendix B have been used

[1] All simulations have been done using a MacOS 10.7.5, with processor Intel Core i5, 1.7 GHz, MatLab r2013a, solver CPLEX [IBM11], YALMIP [Lö04] and MPT [KGB04].



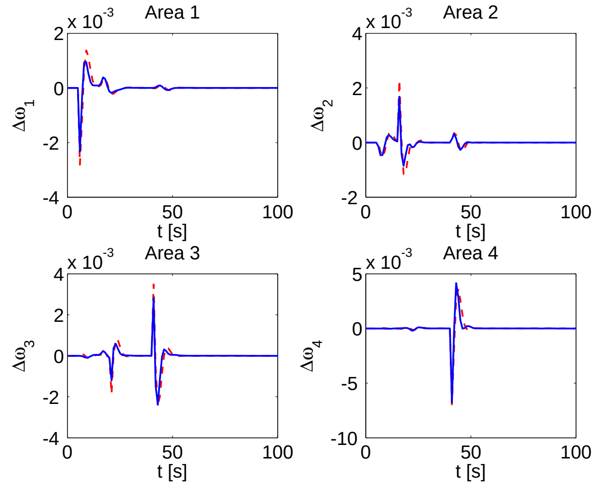
(a) Displacement of truck i controlled by $\mathcal{C}_{[i]}$ (dashed line) and $\mathcal{C}_{[i]}^{dis}$ (continuous line).



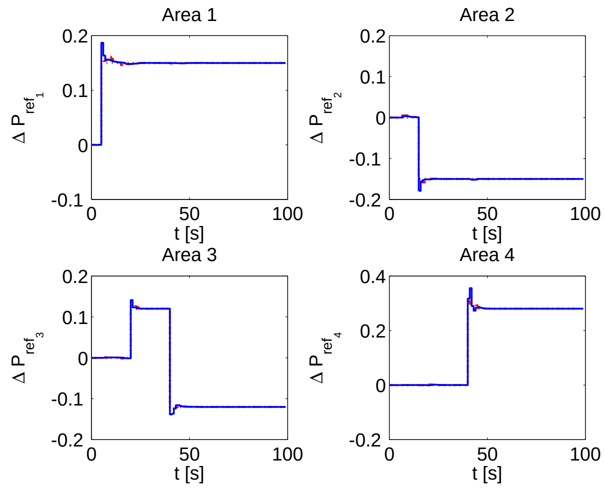
(b) Control law computed by using $\mathcal{C}_{[i]}$ (dashed line) and $\mathcal{C}_{[i]}^{dis}$ (continuous line).

Figure 6.2: Simulation over the first three time-instants with initial state $\mathbf{x} = (0, 0, 3, 0)$.

and they account for the step-like changes of the control variables in Figure 6.3. We highlight that the performance of DeMPC and CeMPC are totally comparable, in terms of frequency deviation (Figure 6.3a), control variables (Figure 6.3b) and power transfers $\Delta P_{tie_{ij}}$ (Figure 6.4). The values of performance parameter η and Φ using different controllers are reported in Table 6.1 and Table 6.2, respectively. In terms of parameter η, PnP controllers with decentralized and distributed online implementation are equivalent to centralized controller. However the performance of PnP-DeMPC-rpi are such that each area can absorb the local loads by producing more power lo-



(a) Frequency deviation in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).



(b) Load reference set-point in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

Figure 6.3: Simulation Scenario 1: 6.3a Frequency deviation and 6.3b Load reference in each area.

cally ($\Delta P_{ref,i}$) instead of receiving power from parent areas: for this reason, performances of PnP-DiMPC are more similar to CeMPC. Compared with PnP controllers proposed in Chapter 5, PnP-DeMPC has better tracking performance: we reduce the value of parameter η (PnP-DeMPC 0.0264,

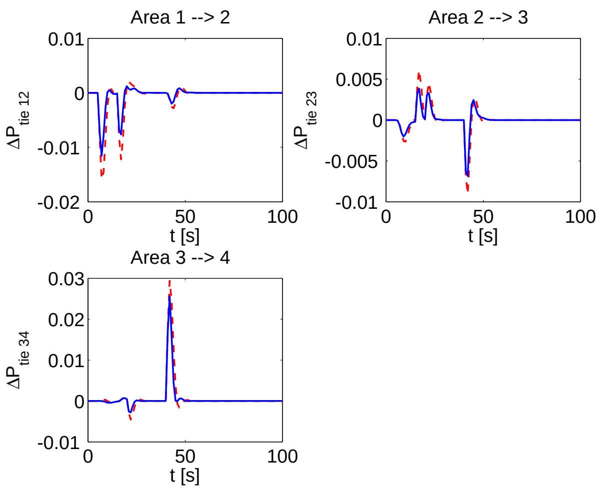
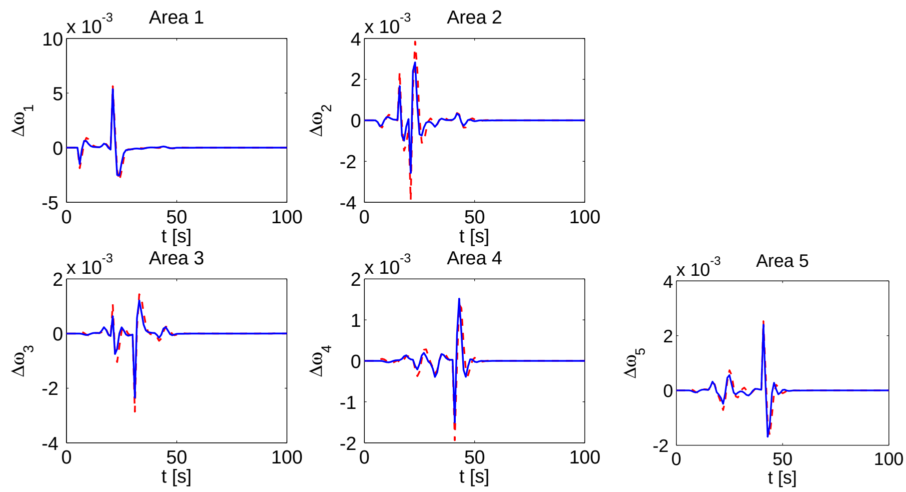


Figure 6.4: Simulation Scenario 1: tie-line power between each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

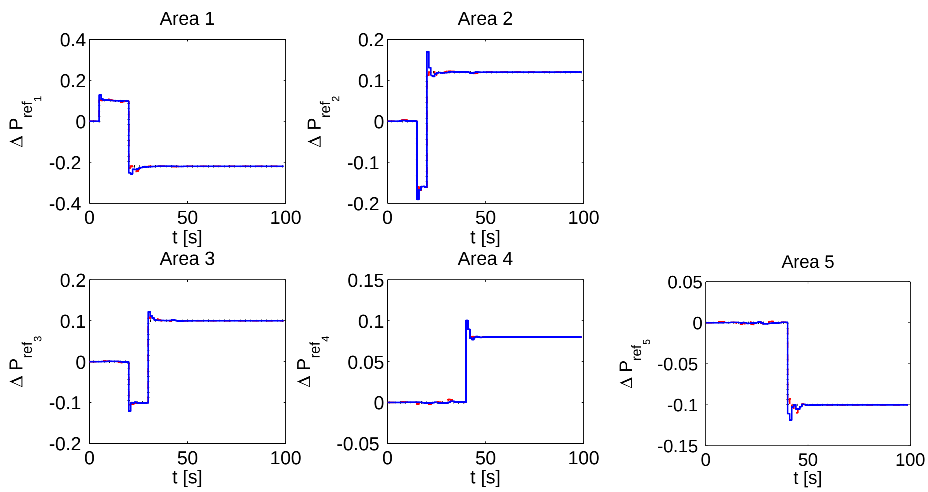
PnP-DeMPC-rpi 0.0301), but not the value of parameter Φ (PnP-DeMPC 0.0019, PnP-DeMPC-rpi 0.0016).

Scenario 2

We consider the power network proposed in Scenario 1 and we add a fifth area connected as in Section B.1.2. Therefore, the set of children of subsystem 5 is $\mathcal{S}_5 = \{2, 4\}$. Since systems $\Sigma_{[j]}$, $j \in \mathcal{S}_5$ depend on a parameter related to the added subsystem $\Sigma_{[5]}$, a retuning of their controllers is needed. We highlight that the proposed framework, as also the PnP method proposed in Chapter 5, allows for subsystems with parameters that depend upon their parents. In this case, as discussed in Chapter 5, even in the unplugging operation the child subsystems have to retune their controllers to guarantee overall asymptotic stability and constraints satisfaction. The controllers $\mathcal{C}_{[j]}$, $j \in \{5\} \bigcup \mathcal{S}_5$ are tuned using Algorithm 6.1. We highlight that no retuning of controllers $\mathcal{C}_{[1]}$ and $\mathcal{C}_{[3]}$ is needed since $\Sigma_{[1]}$ and $\Sigma_{[3]}$ are not parents of system $\Sigma_{[5]}$.
In Figure 6.5 we compare the performance of proposed DeMPC with the performance of CeMPC. In the control experiment, step power loads ΔP_{L_i} specified in Table B.4 in Section B.1.2 have been used and they account for the step-like changes of the control variables in Figure 6.5. We highlight



(a) Frequency deviation in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).



(b) Load reference set-point in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

Figure 6.5: Simulation Scenario 2: 6.5a Frequency deviation and 6.5b Load reference in each area.

that the performance of DeMPC and CeMPC are totally comparable, in terms of frequency deviation (Figure 6.5a), control variables (Figure 6.5b) and power transfers $\Delta P_{tie_{ij}}$ (Figure 6.6). The values of performance parameter η and Φ using different controllers are reported in Table 6.1 and Table

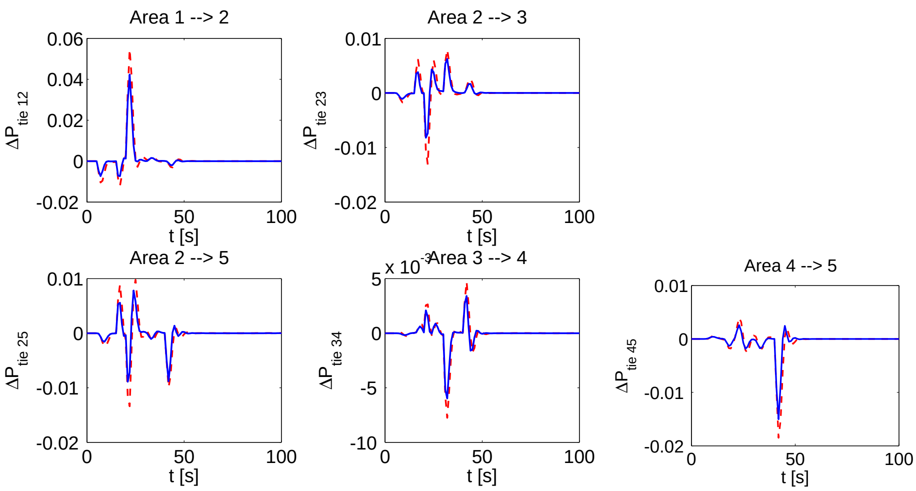
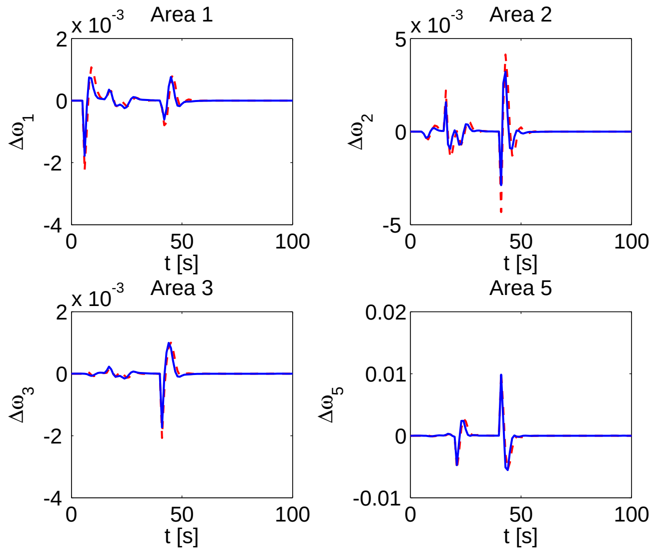


Figure 6.6: Simulation Scenario 2: tie-line power between each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

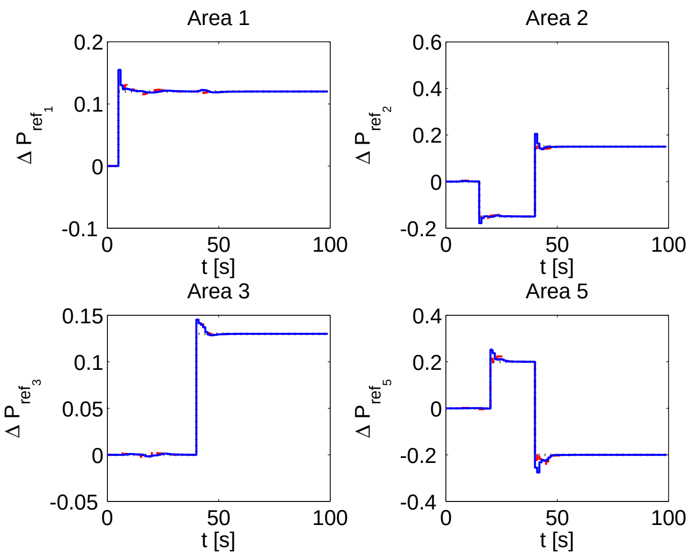
6.2, respectively. In terms of parameter η, PnP controllers with decentralized and distributed online implementation are equivalent to centralized controller, however, as in Scenario 1, the performance of PnP-DeMPC are such that each area can absorb the local loads by producing more power locally ($\Delta P_{ref,i}$) instead of receiving power from parent areas: for this reason, PnP-DiMPC has performance more similar to CeMPC. Compared with PnP-DeMPC-rpi controllers, PnP-DeMPC has better performances in terms of parameter Φ: this corresponds to a reduction of the exchanged power at the price of slightly worse tracking capabilities (η increases).

Scenario 3

We consider the power network described in Scenario 2 and disconnect the area 4, as in Section B.1.3. The set of children of subsystem 4 is $\mathcal{S}_4 = \{3, 5\}$. Because of disconnection, systems $\Sigma_{[j]}$, $j \in \mathcal{S}_4$ change their parents and local dynamics A_{jj}. Then, as explained in Section 6.6.2, the retuning of controllers of child subsystems is needed. We highlight that retuning of controllers $\mathcal{C}_{[1]}$ and $\mathcal{C}_{[2]}$ is not needed since systems $\Sigma_{[1]}$ and $\Sigma_{[2]}$ are not children of subsystem $\Sigma_{[4]}$.
In Figure 6.7 we compare the performance of proposed DeMPC with the performance of the CeMPC. In the control experiment, step power loads



(a) Frequency deviation in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).



(b) Load reference set-point in each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

Figure 6.7: Simulation Scenario 3: 6.7a Frequency deviation and 6.7b Load reference in each area.

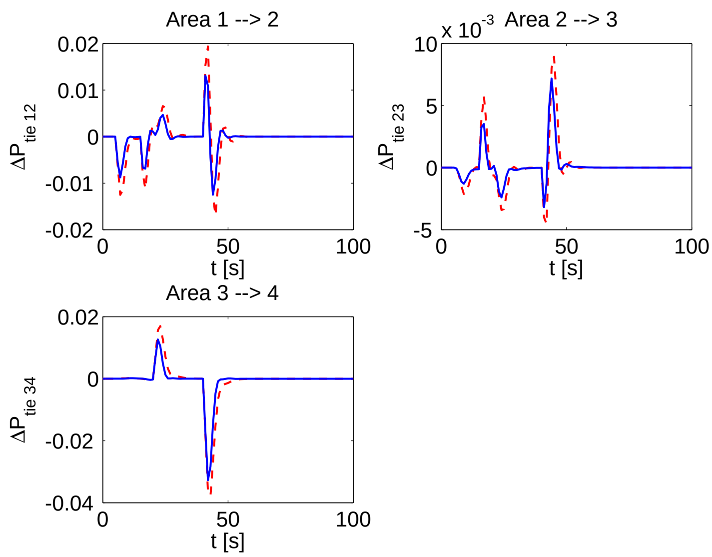


Figure 6.8: Simulation Scenario 3: tie-line power between each area controlled by the proposed DeMPC (continuous line) and CeMPC (dashed line).

ΔP_{L_i} specified in Table B.5 in Section B.1.3 have been used also in this case.

Performances of DeMPC and CeMPC are totally comparable in terms of frequency deviation (Figure 6.7a), control variables (Figure 6.7b) and power transfers $\Delta P_{tie_{ij}}$ (Figure 6.8). The values of performance parameter η and Φ using different controllers are reported in Table 6.1 and Table 6.2, respectively. In terms of parameter η, PnP controllers with decentralized and distributed online implementation are equivalent to centralized controller, however the performance of PnP-DeMPC are such that each area can absorb the local loads by producing more power locally ($\Delta P_{ref,i}$) instead of receiving power from parent areas. For this reason, PnP-DiMPC has performances more similar to CeMPC and PnP-DeMPC reduces of 40% the performance index Φ. Similarly to Scenario 2, compared with PnP-DeMPC-rpi, PnP-DeMPC has better performance in terms of index Φ but worse tracking capabilities (η increases).

	Scenario 1		Scenario 2		Scenario 3	
	MPCdiag	*MPCzero*	*MPCdiag*	*MPCzero*	*MPCdiag*	*MPCzero*
CeMPC	0.0249	0.0249	0.0346	0.0347	0.0510	0.0511
PnP-DeMPC	+6.02%	+6.02%	+7.23%	+6.92%	+8.24%	+8.02%
PnP-DiMPC	+**3.61**%	+**3.61**%	+**2.02**%	+**1.73**%	+**1.96**%	+**1.76**%
PnP-DeMPC-rpi	+20.88%	+20.88%	+10.40%	+10.09%	+4.51%	+4.31%

Table 6.1: Value of the performance parameter η for CeMPC (first line) and percentage change using DeMPC and DiMPC schemes for the AGC layer. Best values for PnP controllers are in bold.

	Scenario 1		Scenario 2		Scenario 3	
	MPCdiag	*MPCzero*	*MPCdiag*	*MPCzero*	*MPCdiag*	*MPCzero*
CeMPC	0.0030	0.0029	0.0063	0.0061	0.0060	0.0058
PnP-DeMPC	−36.67%	−34.48%	−**38.10**%	−**38.10**%	−**40.00**%	−**37.93**%
PnP-DiMPC	+0.00%	+0.00%	−7.94%	−4.92%	−5.00%	−1.72%
PnP-DeMPC-rpi	−**46.67**%	−**44.83**%	−25.40%	−22.95%	−25.00%	−22.41%

Table 6.2: Value of the performance parameter Φ for CeMPC (first line) and percentage change using DeMPC and DiMPC schemes for the AGC layer. Best values for PnP controllers are in bold.

6.6.3 Large-scale mechanical system

We consider a LSS composed of 1024 masses ($\mathcal{M} = 1 : 1024$) coupled as in Figure 6.10b through springs and dampers arranged as in Figure 6.9.

Each mass $i \in \mathcal{M}$, is a subsystem with state $x_{[i]} = (x_{[i,1]}, x_{[i,2]}, x_{[i,3]}, x_{[i,4]})$ and input $u_{[i]} = (u_{[i,1]}, u_{[i,2]})$, where $x_{[i,1]}$ and $x_{[i,3]}$ are the displacements of mass i with respect to a given equilibrium position on the plane (equilibria lie on the regular grid in Figure 6.10b), $x_{[i,2]}$ and $x_{[i,4]}$ are the horizontal and vertical velocities of the mass i and $100u_{[i,1]}$ (respectively $100u_{[i,2]}$) is the force applied to mass i in the horizontal (respectively, vertical) direction. The values of m_i have been extracted randomly in the interval $[5, 10]$ while spring constants and damping coefficients are identical and equal to 0.5. Subsystems are equipped with the state constraints $||x_{[i,j]}||_\infty \leq 1.5$, $j = 1, 3$, $||x_{[i,l]}||_\infty \leq 0.8$, $i \in \mathcal{M}$, $l = 2, 4$ and with the input constraints $||u_{[i]}||_\infty \leq \Gamma_i$, where Γ_i have been randomly chosen in the interval $[1, 1.5]$. We obtain models $\Sigma_{[i]}$ by discretizing continuous-time models with 0.2 sec sampling time, using zero-order hold discretization for the local dynamics and treating $x_{[j]}$, $j \in \mathcal{N}_i$ as exogenous signals [FCS13]. We synthesized controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ using Algorithm 6.1 and plug in a new mass at each iteration. In the worst case the time required to solve Step (II) of Algo-

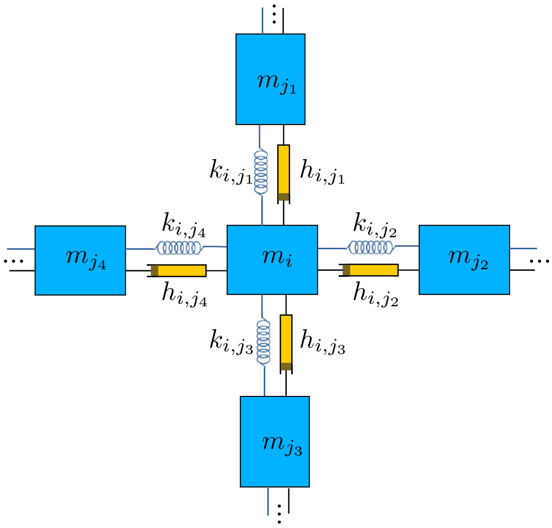
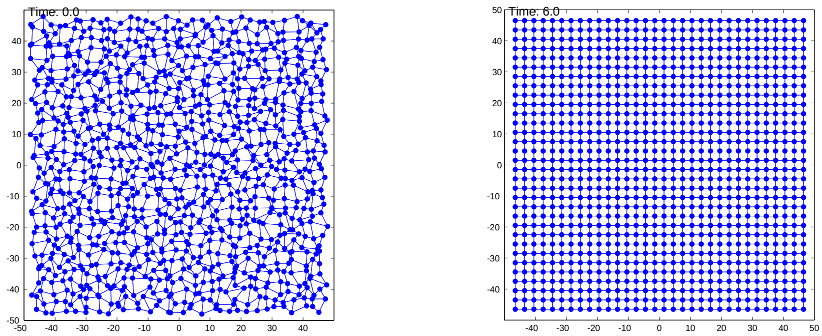


Figure 6.9: Array of masses: details of interconnections.

rithm 6.1 is 0.2598 sec (best case 0.0140 sec)[2]. Note also that the use of a CeMPC is prohibitive since the overall system has $\mathbf{x} \in \mathbb{R}^{4096}$, $\mathbf{u} \in \mathbb{R}^{2048}$ and therefore $8192 + 4096$ scalar affine constraints. In Figure 6.10 we show a simulation where, at time $t = 0$, the masses are still and placed as in Figure 6.10a. At all time steps t, the control action $u_{[i]}(t)$ computed by the controller $\mathcal{C}_{[i]}$, for all $i \in \mathcal{M}$, is kept constant during the sampling interval and applied to the continuous-time system. In the worst case, the computation of the control law (6.4) requires 0.1047 sec. Convergence is obtained for all masses to their equilibrium position while fulfilling input and state constraints. State and input variables are depicted in Figure 6.11. From Figure 6.11, the estimated settling time at 95% is 5.37 sec. For this LSS, we also have considered the use of PnP-DeMPC controllers proposed in Chapter 5, but since the design of local controllers requires the solution to nonlinear optimization problem and also the number of local constraints is large, the design procedure in Chapter 5 did not give conclusive results after several hours of computation.

[2]All simulations have been done using a Linux distribution (Kubuntu 12.04), with processor Intel Core i7-2600, 3.4 GHz, MatLab r2011b, solver CPLEX [IBM11], YALMIP [Lö04] and MPT [KGB04].



(a) Position of the masses at initial time. (b) Position of the masses at time 6 sec.

Figure 6.10: Position of the 1024 masses on the plane.

6.7 Final comments

In this chapter we proposed a DeMPC scheme, based on the notion of tube-based control [RM05], for guaranteeing asymptotic stability of the closed-loop system and constraints satisfaction at each time instant. Our design procedures allows PnP operations: if a new subsystem enters the network we can design a local controller using information from parent subsystems. Differently from the methods discussed in Chapter 5, testing the feasibility of PnP operations and computing local controllers amounts to solve LP problems only.

6.8 Appendix

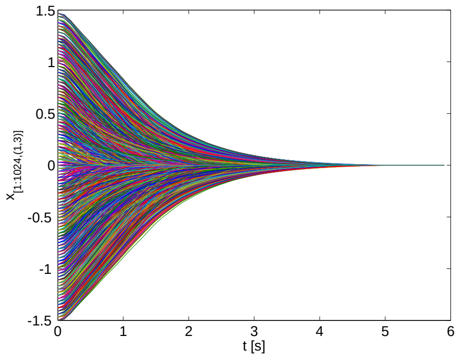
6.8.1 Proof of Theorem 6.1

We start introducing a definition and a few Lemmas that will be used in the proof of Theorem 6.1.

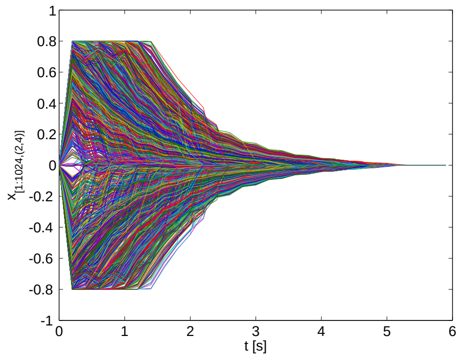
Definition 6.2. A convex body is a nonempty convex and compact set.

Lemma 6.1. *[Sch93] Let A and B be convex bodies. The support function of A is $h_A(x) = \sup_{y\in A} x'y$ and it has the following properties: h_A is sublinear (i.e. $h_A(\alpha x) = \alpha h_A(x)$, $\forall\alpha \geq 0$ and h_A, $h_A(x+y) \leq h_A(x) + h_A(y)$), $h_{\lambda A} = \lambda h_A$, $\forall\lambda > 0$, $h_{A\oplus B} = h_A + h_B$ and $A \subseteq B \Leftrightarrow h_A \leq h_B$.*

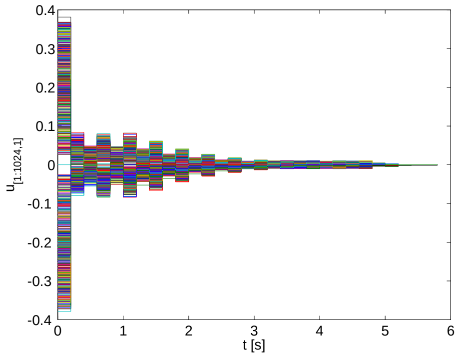
Lemma 6.2. *Let $A = \{x \in \mathbb{R}^n : \mathcal{A}x \leq \mathbf{1}_p\}$, $\mathcal{A} = [a_1,\dots,a_p]^T \in \mathbb{R}^{p\times n}$ and assume that $A \ominus B_\beta(\mathbf{0}_n)$ strictly contains the origin in its interior. Then,*



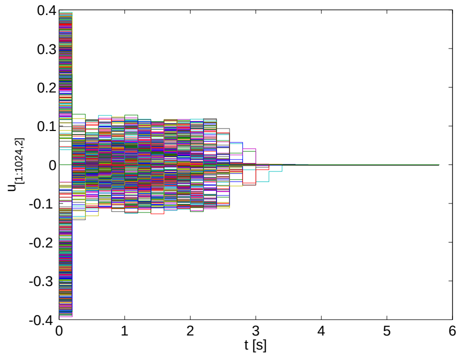
(a) Displacements of the masses with respect their equilibrium positions.



(b) Velocities, i.e. states $x_{[i,2]}$ and $x_{[i,4]}$, $i \in \mathcal{M}$.



(c) Inputs $u_{[i,1]}$, $i \in \mathcal{M}$.



(d) Inputs $u_{[i,2]}$, $i \in \mathcal{M}$.

Figure 6.11: State and input trajectories of the 1024 masses with $\mathbf{x}(0)$ as in Figure 6.10a.

a) $A \ominus B_\beta(\boldsymbol{0}_n) = \{x \in \mathbb{R}^n : \mathcal{A}x \leq \boldsymbol{1}_p - \beta(||a_1||, \dots, ||a_p||)\}$

b) $\beta||a_i|| < 1$, $\forall i \in 1:p$

c) defining $\psi = \min_{i\in 1:p} ||a_i||$ *one has* $A \ominus B_\beta(\boldsymbol{0}_n) \subseteq (1-\beta\psi)A$

Proof. Proceeding as is the proof of point 8 of Proposition 3.28 of [BM08] one gets

$$A \ominus B_\beta(\mathbf{0}_n) = \{x \in \mathbb{R}^n : \mathcal{A}x \leq \tilde{g}\}$$

where $\tilde{g} = (\tilde{g}_1, \dots, \tilde{g}_p)$, $\tilde{g}_i = 1 - h_{B_\beta(\mathbf{0}_n)}(a_i)$, and $h_{B_\beta(\mathbf{0}_n)}(\cdot)$ is the support function of $B_\beta(\mathbf{0}_n)$. Point (a) follows from $h_{B_\beta(\mathbf{0}_n)}(x) = \beta||x||$ (that can be verified using the definition of support functions).

From point (a) one has $x \in A \ominus B_\beta(\mathbf{0}_n)$ if and only if

$$a_i^T x \leq 1 - \beta||a_i||,\ \forall i \in 1:p \tag{6.20}$$

and in order to have that all constraints (6.20) are fulfilled for $x = \mathbf{0}_n$ and no one is active at $x = \mathbf{0}_n$, point (b) must be verified.
For point (c), one has

$$\begin{aligned} A \ominus B_\beta(\mathbf{0}_n) &\subseteq \{x \in \mathbb{R}^n : \mathcal{A}x \leq (1-\beta\psi)\mathbf{1}_p\} \\ &= \{x \in \mathbb{R}^n : \mathcal{A}(\frac{x}{1-\beta\psi}) \leq \mathbf{1}_p\} \end{aligned} \tag{6.21}$$

where the last equality holds because, from point (b), $\beta\psi < 1$. The right-most set in (6.21) is $(1-\beta\psi)A$ and this concludes the proof. □

The next result shows that the control law $\bar{\kappa}_i(x_{[i]})$ defined in (6.15) is homogeneous.

Lemma 6.3. *For $z_{[i]} \in \mathbb{Z}_i$ and $\rho \geq 0$ one has*

$$\bar{\kappa}_i^s(\rho z_{[i]}) = \rho \bar{\kappa}_i^s(z_{[i]}),\ s = 0, \dots, k_i - 1$$

and hence $\bar{\kappa}_i(\rho z_{[i]}) = \rho\bar{\kappa}_i(z_{[i]})$.

Proof. Let $\bar{\beta}_i^{(s,f)}$, $f \in 1:q_i$, $s \in 0:k_i-1$ and $\bar{\mu}$ be the optimizers to $\bar{\mathbb{P}}_i(z_{[i]})$. One can easily verify that $\beta_i^{(s,f)} = \rho\bar{\beta}_i^{(s,f)}$ and $\mu = \rho\bar{\mu}$ fulfill the constraints (6.14b)-(6.14e) for $\bar{\mathbb{P}}_i(\rho z_{[i]})$. We show now that these values are also optimal for $\bar{\mathbb{P}}_i(\rho z_{[i]})$. By contradiction, assume that $\tilde{\beta}_i^{(s,f)}$, $\tilde{\mu}$ are the optimizers to $\bar{\mathbb{P}}_i(\rho z_{[i]})$ giving the optimal cost $\tilde{\mu} < \rho\bar{\mu}$. One can easily verify that $\beta_i^{(s,f)} = \rho^{-1}\tilde{\beta}_i^{(s,f)}$ and $\mu = \rho^{-1}\tilde{\mu}$ verify the constraints (6.14b)-(6.14e) for $\bar{\mathbb{P}}_i(z_{[i]})$ and yield a cost $\rho^{-1}\tilde{\mu} < \bar{\mu}$. This contradicts the optimality of $\bar{\mu}$ for $\bar{\mathbb{P}}_i(z_{[i]})$. □

Lemma 6.4. *If $x_{[i]} \in \rho_i\mathbb{Z}_i$ and $\tilde{w}_{[i]} \in \eta_i\bar{\mathbb{Z}}_i^0$ where $\rho_i \geq \eta_i > 0$, then $x_{[i]}^+ = A_{ii}x_{[i]} + B_i\bar{\kappa}_i(x_{[i]}) + \tilde{w}_{[i]} \in \rho_i\mathbb{Z}_i \ominus (\rho_i - \eta_i)\bar{\mathbb{Z}}_i^0$.*

Proof. Let $\tilde{x}_{[i]} = \frac{x_{[i]}}{\rho_i}$ and $\tilde{w}_{[i]} = \frac{w_{[i]}}{\eta_i}$. From standard arguments in [RB10] one can write $\tilde{x}_{[i]} \in \mathbb{Z}_i$ as

$$\tilde{x}_{[i]} = (1-\alpha_i)^{-1}\sum_{s=0}^{k_i-1}\sigma_i^s(\tilde{x}_{[i]})$$

where $\sigma_i^s(\tilde{x}_{[i]}) \in \bar{\mathbb{Z}}_i^s$ are suitable functions. Then one has

$$x_{[i]} = \rho_i(1-\alpha_i)^{-1} \sum_{s=0}^{k_i-1} \sigma_i^s(\tilde{x}_{[i]})$$

Furthermore, always from [RB10] one has

$$\bar{\kappa}_i(\tilde{x}_{[i]}) = (1-\alpha_i)^{-1} \sum_{s=0}^{k_i-1} \bar{\kappa}_i^s(\tilde{x}_{[i]})$$

and, from Lemma 6.3,

$$\bar{\kappa}_i(x_{[i]}) = \rho_i(1-\alpha_i)^{-1} \sum_{s=0}^{k_i-1} \bar{\kappa}_i^s(\tilde{x}_{[i]}).$$

Computing $x_{[i]}^+$ one gets

$$x_{[i]} = \rho_i(1-\alpha_i)^{-1} \sum_{s=0}^{k_i-1} \underbrace{(A_{ii}\sigma_i^s(\tilde{x}_{[i]}) + B_i\bar{\kappa}_i(\tilde{x}_{[i]}))}_{t_i^s} + \tilde{w}_{[i]} \tag{6.22a}$$

$$\in \left[\rho_i(1-\alpha_i)^{-1} \bigoplus_{s=0}^{k_i-1} \bar{\mathbb{Z}}_i^{s+1}\right] \oplus \left[\eta_i \bar{\mathbb{Z}}_i^0\right] \tag{6.22b}$$

$$= \left[\rho_i(1-\alpha_i)^{-1} \bigoplus_{s=1}^{k_i-1} \bar{\mathbb{Z}}_i^{s}\right] \oplus \left[\rho_i(1-\alpha_i)^{-1}\bar{\mathbb{Z}}_i^{k_i}\right] \oplus \left[\eta_i \bar{\mathbb{Z}}_i^0\right] \tag{6.22c}$$

$$= \left\{\left[\rho_i(1-\alpha_i)^{-1} \bigoplus_{s=1}^{k_i-1} \bar{\mathbb{Z}}_i^{s}\right] \oplus \rho_i\alpha_i(1-\alpha_i)^{-1}\bar{\mathbb{Z}}_i^0 \oplus \rho_i\bar{\mathbb{Z}}_i^0 \oplus \eta_i\bar{\mathbb{Z}}_i^0\right\} \ominus \ominus \rho_i\bar{\mathbb{Z}}_i^0 \tag{6.22d}$$

$$= \left\{\left[\rho_i(1-\alpha_i)^{-1} \bigoplus_{s=0}^{k_i-1} \bar{\mathbb{Z}}_i^{s}\right] \oplus \eta_i\bar{\mathbb{Z}}_i^0\right\} \ominus \rho_i\bar{\mathbb{Z}}_i^0 \tag{6.22e}$$

$$= \left\{\rho_i\mathbb{Z}_i \oplus \eta_i\bar{\mathbb{Z}}_i^0\right\} \ominus \rho_i\bar{\mathbb{Z}}_i^0 \tag{6.22f}$$

$$\subseteq \rho_i\mathbb{Z}_i \ominus (\rho_i - \eta_i)\mathbb{Z}_i^0 \tag{6.22g}$$

that is the desired result. Note that

- in (6.22b) we used $t_i^s \in \bar{\mathbb{Z}}_i^{s+1}$ (that holds by construction of sets $\bar{\mathbb{Z}}_i^s$);

- in (6.22d) we used the property that $(A \oplus B) \ominus B = A$, if $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^n$ are convex bodies (see Lemma 3.18 in [Sch93]), and the fact that (6.9) implies $\bar{\mathbb{Z}}_i^{k_i} \subseteq \alpha_i \bar{\mathbb{Z}}_i^0$;
- in (6.22e) we used $\rho_i \alpha_i (1-\alpha_i)^{-1} \bar{\mathbb{Z}}_i^0 \oplus \rho_i \bar{\mathbb{Z}}_i^0 = \rho_i(\alpha_i(1-\alpha_i)^{-1}+1)\bar{\mathbb{Z}}_i^0 = \rho_i(1-\alpha_i)^{-1}\bar{\mathbb{Z}}_i^0$;
- in (6.22f) we used (6.10);
- in (6.22g) we used the property that if $\rho > \eta > 0$ and A and B are convex bodies, then

$$(\rho A \oplus \eta B) \ominus \rho B \subseteq \rho A \ominus (\rho - \eta) B. \tag{6.23}$$

The inclusion (6.23) can be shown as follows. One has, from the definition of the operator $\ominus$,

$$x \in (\rho A \oplus \eta B) \ominus \rho B \Leftrightarrow x \oplus \rho B \subseteq \rho A \oplus \eta B$$

and therefore, using Lemma 6.1

$$\begin{aligned} h_{\{x\}} + \rho h_B &\leq \rho h_A + \eta h_B \\ h_{\{x\}} + (\rho - \eta) h_B &\leq \rho h_A. \end{aligned} \tag{6.24}$$

Since $(\rho - \eta) > 0$, one has that $(\rho - \eta) h_B$ is the support function of $(\rho - \eta)B$. Then (6.24) is equivalent to

$$x \oplus (\rho - \eta) B \subseteq \rho A$$

that is $x \in \rho A \ominus (\rho - \eta) B$.

□

Proof of Theorem 6.1

Proof. The first part of the proof uses arguments similar to the ones adopted for proving Theorem 1 both in [FS12] and Chapter 5. Indeed we can easily prove that $\hat{x}_{[i]}(0|t) \rightarrow \mathbf{0}_{n_i}$ and $v_{[i]}(0|t) \rightarrow \mathbf{0}_{m_i}$ as $t \rightarrow \infty$. We highlight that constraint (5.13b) in the MPC-i problem is replaced by (6.7).
Next we prove stability of the origin for the closed-loop system. We highlight that this part of the proof differs substantially from the proof of Theorem 1 both in [FS12] and Chapter 5. For the sake of clarity, the proof is split in three distinct steps.

Step 1: *Prove that if* $\mathbf{x}(0) \in \mathbb{X}^N$ *there is* $\tilde{T} > 0$ *such that* $\mathbf{x}(\tilde{T}) \in \mathbb{Z}$.
Recalling that the state $\mathbf{x}(t)$ evolves according to the equation (6.16), we can write

$$\mathbf{x}(t+1) = \mathbf{A_D}\mathbf{x}(t) + \mathbf{B}\bar{\kappa}(\mathbf{x}(t)) + \mathbf{A_c}\mathbf{x}(t) + \bar{\boldsymbol{\eta}}(t) \tag{6.25}$$

where $\mathbf{A_D} = \mathrm{diag}(A_{11}, \ldots, A_{MM})$, $\mathbf{A_C} = \mathbf{A} - \mathbf{A_D}$,

$$\bar{\boldsymbol{\eta}}(t) = \mathbf{B}(\mathbf{v}(t) + \bar{\boldsymbol{\kappa}}(\mathbf{z}(t)) - \bar{\boldsymbol{\kappa}}(\mathbf{x}(t))) \tag{6.26}$$

and $\mathbf{z}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(0|t)$. In particular, if $\mathbf{x}(0) \in \mathbb{X}^N$, recursive feasibility shown above implies that (6.25) holds for all $t \geq 0$.
Note that in view of Assumption 6.3, the LP problem (6.14) is feasible for all $z_{[i]} \in \mathbb{R}^{n_i}$. Indeed (6.14c) and (6.14e) require that there are $\mu > 0$ and $\bar{z}^s_{[i]} \in \mu\bar{\mathbb{Z}}^s_i$, $s \in 0 : k_i - 1$ such that $z_{[i]} = \sum_{i=0}^{k_i-1} \bar{z}^s_{[i]}$ and since $\bar{\mathbb{Z}}^0_i \supset B_{\omega_i}(\mathbf{0}_{n_i})$ (i.e. $\bar{\mathbb{Z}}^0_i$ is full dimensional), these quantities always exist. This implies that the function $\bar{\boldsymbol{\kappa}}(\mathbf{x}(t))$ in (6.25) is always well defined.
We already proved the asymptotic convergence to zero of the nominal state $\hat{\mathbf{x}}(0|t)$ and the input signal $\mathbf{v}(0|t)$ and hence it holds

$$\forall \delta > 0,\ \exists T_1 > 0 :\ ||\hat{\mathbf{x}}(0|t)|| \leq \delta \text{ and } ||\mathbf{v}(0|t)|| \leq \delta,\ \forall t \geq T_1. \tag{6.27}$$

Moreover, according to [Gal95], we can assume without loss of generality that $\bar{\kappa}_i(\cdot)$ is a continuous piecewise affine map. In view of this, $\bar{\boldsymbol{\kappa}}(\cdot)$ is also globally Lipschitz, i.e.

$$\exists\ L > 0\ :\ ||\bar{\boldsymbol{\kappa}}(\mathbf{x} - \hat{\mathbf{x}}) - \bar{\boldsymbol{\kappa}}(\mathbf{x})|| \leq L||\hat{\mathbf{x}}|| \tag{6.28}$$

for all $(\mathbf{x}, \hat{\mathbf{x}})$ such that $\mathbf{x} \in \mathbb{X}$ and $\mathbf{x} - \hat{\mathbf{x}} \in \mathbb{Z}$. Using (6.28) one can show that, for all $\epsilon > 0$, setting $\delta = \frac{\epsilon}{||B||(1+L)}$ the following implication holds

$$||\hat{\mathbf{x}}(0|t)|| \leq \delta \text{ and } ||\mathbf{v}(0|t)|| \leq \delta \Rightarrow ||\bar{\boldsymbol{\eta}}(t)|| \leq \epsilon,\ \forall x(t) \in \mathbb{X}.$$

Therefore, from (6.27),

$$\forall \epsilon > 0,\ \exists T_1 > 0 :\ ||\bar{\boldsymbol{\eta}}(t)|| \leq \epsilon,\ \forall t \geq T_1. \tag{6.29}$$

Since $||\hat{\mathbf{x}}(0|t)|| \to 0$, as $t \to \infty$, and $\mathbb{Z}$ contains $\prod_{i=1}^M B_{\omega_i}(\mathbf{0}_{n_i})$, then

$$\forall \delta_z > 0,\ \exists T_2 > 0 :\ \hat{\mathbf{x}}(0|t) \in \delta_z\mathbb{Z},\ \forall t \geq T_2 \tag{6.30}$$

and hence, from (6.7),

$$\mathbf{x}(t) = \hat{\mathbf{x}}(0|t) + (\mathbf{x}(t) - \hat{\mathbf{x}}(0|t)) \in (1 + \delta_z)\mathbb{Z},\ \forall t \geq T_2. \tag{6.31}$$

From (6.25) we have, for all $i \in \mathcal{M}$,

$$x_{[i]}(t+1) = A_{ii}x_{[i]}(t) + B_i\bar{\kappa}_i(x_{[i]}(t)) + \tilde{w}_{[i]}(t) \tag{6.32}$$

where $\tilde{w}_{[i]} = \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]} + \bar{\eta}_{[i]}$, $\forall i \in \mathcal{M}$. Setting $\bar{T} = \max\{T_1, T_2\}$ and using (6.29) and (6.31), one has, $\forall t \geq \bar{T}$

$$\tilde{w}_{[i]} \in (1+\delta_z)\bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{Z}_j \oplus B_\epsilon(\mathbf{0}_{n_i}). \tag{6.33}$$

From Assumption 6.2 we have

$$\bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{Z}_j \subseteq \bigoplus_{j\in\mathcal{N}_i} A_{ij}\left[\mathbb{X}_j \ominus B_{\rho_{j,1}}(\mathbf{0}_{n_j})\right] \tag{6.34a}$$

$$\subseteq \bigoplus_{j\in\mathcal{N}_i}\left[(A_{ij}\mathbb{X}_j) \ominus \left(A_{ij}B_{\rho_{j,1}}(\mathbf{0}_{n_j})\right)\right] \tag{6.34b}$$

$$\subseteq \left(\bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{X}_j\right) \ominus \left(\bigoplus_{j\in\mathcal{N}_i} A_{ij}B_{\rho_{j,1}}(\mathbf{0}_{n_j})\right) \tag{6.34c}$$

$$\subseteq \mathbb{W}_i \ominus \left(\bigoplus_{j\in\mathcal{N}_i} A_{ij}B_{\rho_{j,1}}(\mathbf{0}_{n_j})\right). \tag{6.34d}$$

Manipulations (6.34b) and (6.34c) are justified as follows. Let U_1, U_2, V_1, V_2 be convex bodies in $\mathbb{R}^n$, i.e. nonempty convex and compact sets. Then, from [Sch93], (6.34b) follows from $G(U \ominus V) \subseteq GU \ominus GV$, where $G \in \mathbb{R}^{n\times n}$. Furthermore (6.34c) follows from $(U_1 \ominus V_1)\oplus(U_2\ominus V_2) \subseteq (U_1\oplus U_2)\ominus(V_1\oplus V_2)$. Therefore, there is $\xi_i \in [0,1)$ (that does not depend on ϵ and δ_z) such that

$$\bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{Z}_j \subseteq \xi_i\mathbb{W}_i, \tag{6.35}$$

and then, from (6.33),

$$\tilde{w}_{[i]} \in (1+\delta_z)\xi_i\mathbb{W}_i \oplus B_\epsilon(\mathbf{0}_{n_i}),\ \forall t \geq \bar{T}.$$

Note that in (6.29) the parameter $\epsilon > 0$ can be chosen arbitrarily small. Assume that it verifies $\epsilon < (1+\delta_z)\xi_i\omega_i$, $\forall i \in \mathcal{M}$ where ω_i are the radii of the balls in Assumption 6.3. Then, using Assumption 6.3 we get for $t \geq \bar{T}$

$$\tilde{w}_{[i]}(t) \in (1+\delta_z)\xi_i(\mathbb{W}_i \oplus B_{\omega_i}(\mathbf{0}_{n_i})) \subseteq (1+\delta_z)\xi_i\bar{\mathbb{Z}}_i^0. \tag{6.36}$$

In view of (6.31) and (6.36), Lemma 6.2 guarantees that

$$x_{[i]}^+ \in (1+\delta_z)(\mathbb{Z}_i \ominus (1-\xi_i)\bar{\mathbb{Z}}_i^0) \tag{6.37}$$

From Assumption 6.3, one has $\mathbb{Z}_i \ominus (1-\xi_i)\bar{\mathbb{Z}}_i^0 \subset \mathbb{Z}_i \ominus B_{(1-\xi_i)\omega_i}(\mathbf{0}_{n_i})$ and hence, since $\mathbb{Z}_i$ contains the origin in its interior, there is $\mu_i \in [0,1)$ such that $\mathbb{Z}_i \ominus (1-\xi_i)\mathbb{Z}_i^0 \subset \mu_i\mathbb{Z}_i$. From (6.37) we get $x_{[i]}^+ \in (1+\delta_z)\mu_i\mathbb{Z}_i$. If in (6.30) we set δ_z such that $(1+\delta_z)\mu_i < 1$, we have shown that for $t = \bar{T}$ it holds $x_{[i]}(\bar{T}+1) \in \mathbb{Z}_i$ and the proof of Step 1 is concluded setting $\tilde{T} = \bar{T}+1$.

Step 2 *Prove that if* $\mathbf{x}(0) \in \mathbb{X}^N$*, then* $\mathbf{x}(t) \to \mathbf{0}_n$ *as* $t \to +\infty$.
Set $t = \tilde{T}$. Since from Step 1 (that holds if $\mathbf{x}(0) \in \mathbb{X}^N$), one has $x_{[i]}(t) \in \mathbb{Z}_i$, under Assumption 5.3, the optimizers to $\mathbb{P}_i^N(x_{[i]}(t))$ are $v_{[i]}(0|t) = \mathbf{0}_{m_i}$ and $\hat{x}_{[i]}(0|t) = \mathbf{0}_{n_i}$. Hence, from (6.26) one has $\bar{\boldsymbol{\eta}}(t) = \mathbf{0}_n$ and (6.32) holds with $\tilde{w}_{[i]}(t) = \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]}(t)$. Furthermore, since $u_{[i]} = \bar{\kappa}_i(x_{[i]})$ is the control law that makes the set $\mathbb{Z}_i$ RCI with respect to $\tilde{w}_{[i]}$, one has $x_{[i]}(t+1) \in \mathbb{Z}_i$. The previous arguments can be applied in a recursive fashion showing that, $\forall t \geq \tilde{T}$

$$x_{[i]}(t) \in \mathbb{Z}_i \tag{6.38}$$

$$\tilde{w}_{[i]}(t) = \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]}(t). \tag{6.39}$$

From (6.35), (6.38), (6.39) and Assumption 6.3

$$\tilde{w}_{[i]}(t) \in \bigoplus_{j\in\mathcal{N}_i} A_{ij}\mathbb{Z}_j \subseteq \xi_i\mathbb{W}_i \subseteq \xi_i\bar{\mathbb{Z}}_i^0. \tag{6.40}$$

Set $t = \tilde{T}$ and $\lambda_i(t) = 1$, $\forall i \in \mathcal{M}$. From (6.38) and (6.40) it holds

$$x_{[i]}(t) \in \lambda_i(t)\mathbb{Z}_i \tag{6.41}$$

$$\tilde{w}_{[i]}(t) \in \lambda_i(t)\xi_i\bar{\mathbb{Z}}_i^0. \tag{6.42}$$

From Lemma 6.4 we have

$$x_{[i]}(t+1) \in \lambda_i(t)\mathbb{Z}_i \ominus (1-\xi_i)\lambda_i(t)\bar{\mathbb{Z}}_i^0.$$

From Assumption 6.3, it holds $B_{\omega_i}(\mathbf{0}_{n_i}) \subseteq \bar{\mathbb{Z}}_i^0$. Then

$$x_{[i]}(t+1) \in \lambda_i(t)\mathbb{Z}_i \ominus B_{(1-\xi_i)\lambda_i(t)\omega_i}(\mathbf{0}_{n_i}). \tag{6.43}$$

The next goal is to compute $\tilde{\lambda}_i(t+1) < \lambda_i(t)$ such that

$$x_{[i]}(t+1) \in \tilde{\lambda}_i(t+1)\mathbb{Z}_i. \tag{6.44}$$

This is achieved using Lemma 6.2 with $A = \mathbb{Z}_i$ and $\beta = (1-\xi_i)\omega_i$. We have to verify that $A \ominus B_\beta(\mathbf{0}_{n_i})$ contains the origin in its interior and since $B_{\omega_i}(\mathbf{0}_{n_i}) \in \bar{\mathbb{Z}}_i^0$, it is enough to show that the smaller set $\mathbb{Z}_i \ominus (1-\xi_i)\bar{\mathbb{Z}}_i^0$ contains the origin in its interior.
From (6.10) one has (6.45)

$$\begin{aligned}\mathbb{Z}_i \ominus (1-\xi_i)\bar{\mathbb{Z}}_i^0 &= \left(\left((1-\alpha_i)^{-1}\bigoplus_{s=1}^{k_i-1}\bar{\mathbb{Z}}_i^s\right) \oplus (1-\alpha_i)^{-1}\bar{\mathbb{Z}}_i^0\right) \ominus (1-\xi_i)\bar{\mathbb{Z}}_i^0\\ &\supseteq \left((1-\alpha_i)^{-1}\bar{\mathbb{Z}}_i^0 \ominus (1-\xi_i)\bar{\mathbb{Z}}_i^0\right) \oplus \left((1-\alpha_i)^{-1}\bigoplus_{s=1}^{k_i-1}\mathbb{Z}_i^s\right)\end{aligned} \tag{6.45}$$

where the last inclusion follows from the fact that for generic sets C, D and E in $\mathbb{R}^n$, it holds that $(C \oplus D) \ominus E \supseteq (D \ominus E) \oplus C$ [Mar98]. Note that the origin is strictly contained in $\bar{\mathbb{Z}}_i^0$ (from Assumption 6.3) and also in sets $\bar{\mathbb{Z}}_i^s$, $s \in 1 : k_i - 1$, (by construction). Since in (6.10) $\alpha_i \in [0, 1)$ and we have chosen $\xi_i \in [0, 1)$, one has $(1-\alpha_i)^{-1} > (1-\xi_i)$ and therefore $0 \in (1-\alpha_i)^{-1}\bar{\mathbb{Z}}_i^0 \ominus (1-\xi_i)\bar{\mathbb{Z}}_i^0$. Since the origin is strictly contained in all summands appearing in (6.45), it is also strictly contained in $A \ominus B_\beta(\mathbf{0}_{n_i})$. Letting $\bar{\mathbb{Z}}_i^0 = \{z \in \mathbb{R}^{n_i} : \bar{\mathcal{Z}}_i^0 z \leq \mathbf{1}_{q_i}\}$ and $\bar{\mathcal{Z}}_i^0 = [\bar{z}_{i,1}^0, \ldots, \bar{z}_{i,q_i}^0]^T$, from point c of Lemma 6.2 we get $\mathbb{Z}_i^0 \ominus B_{(1-\xi_i)\omega_i}(\mathbf{0}_{n_i}) \subseteq (1-(1-\xi_i)\omega_i\psi_i)\mathbb{Z}_i$, $\psi_i = \min_{j \in 1:q_i} ||\bar{\mathcal{Z}}_{i,j}^0||$. From (6.43), one obtains that (6.44) is fulfilled with

$$\begin{cases}\tilde{\lambda}_i(t+1) = a_i\lambda_i(t)\\ \quad a_i = 1-(1-\xi_i)\omega_i\psi_i\end{cases} \tag{6.46}$$

and from point b of Lemma 6.2, it holds $|a_i| < 1$.
From (6.39) and (6.44) we have $\tilde{w}_{[i]}(t+1) \in \bigoplus A_{ij}\tilde{\lambda}_j(t+1)\mathbb{Z}_j$ and setting

$$\lambda_i(t+1) = \max_{j \in \mathcal{N}_i \cup \{i\}} \tilde{\lambda}_j(t+1) \tag{6.47}$$

it holds

$$\tilde{w}_{[i]}(t+1) \in \lambda_i(t+1)(\bigoplus_{j \in \mathcal{N}_i} A_{ij}\mathbb{Z}_j) \subseteq \lambda_i(t+1)\xi_i\bar{\mathbb{Z}}_i^0 \tag{6.48}$$

where we used (6.35) and Assumption 6.3. Note that (6.48) corresponds to (6.42) at time $t+1$. Similarly, from (6.44) and (6.47) we have that (6.41) holds at time $t+1$. Furthermore $0 \leq \tilde{\lambda}_i(t+1) < 1$ and $0 \leq \lambda_i(t+1) < 1$. Therefore, the previous arguments can be applied in a recursive fashion to prove that (6.41), (6.42), (6.46) and (6.47) hold for all $t \geq \tilde{T}$ and $i \in \mathcal{M}$.
In order to conclude the proof of Step 2, we show that $\lambda_i(t)$ given by (6.47) converge to zero as $t \to +\infty$. Indeed, from (6.41) this implies that $\mathbf{x}(t) \to \mathbf{0}_n$ as $t \to \infty$. Let $\bar{\lambda}(t) = \max_{i\in\mathcal{M}} \lambda_i(t)$ and $\bar{a} = \max_{i\in\mathcal{M}} a_i$. From (6.46) and (6.47) one has

$$\begin{aligned}
\bar{\lambda}(t+1) &= \max_{i\in\mathcal{M}} \lambda_i(t+1) \\
&= \max_{i\in\mathcal{M}} \max_{j\in\mathcal{N}_i\cup\{i\}} a_i\lambda_i(t) \\
&\leq \bar{a} \max_{i\in\mathcal{M}} \max_{j\in\mathcal{N}_i\cup\{i\}} \lambda_i(t) \\
&= \bar{a}\bar{\lambda}(t).
\end{aligned} \tag{6.49}$$

Being $\bar{a} \in [0,1)$ and $\bar{\lambda}(\tilde{T}) = 1$, (6.49) implies that $\bar{\lambda}(t) \to 0$ as $t \to \infty$.

Step 3 *Prove stability of the origin of the closed-loop system* (6.16).
Note that $\mathbb{Z}_i \subseteq \mathbb{X}_i^N$. Moreover, from Assumption 6.3 and (6.10) (where, from (6.9), $\alpha_i \in [0,1)$) $B_{\omega_i}(\mathbf{0}_{n_i}) \subseteq \mathbb{Z}_i$ and then $\mathbb{Z}$ is a neighborhood of the origin of $\mathbb{R}^n$. For a given $\epsilon > 0$ let $\rho > 0$ be such that $\rho < 1$ and $\rho\mathbb{Z} \subseteq B_\epsilon(\mathbf{0}_n)$. As shown at the beginning of Step 2, if $\mathbf{x} \in \mathbb{Z}$ then problems $\mathbb{P}_i^N(x_{[i]})$, $i \in \mathcal{M}$ are feasible, the closed-loop dynamics (6.16) reduces to

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\bar{\boldsymbol{\kappa}}(\mathbf{x}) \tag{6.50}$$

and $\mathbb{Z}$ is an invariant set for (6.50). This implies that if $\mathbf{x}(0) \in \mathbb{Z}$, then $\mathbf{x}(t)$ is well defined $\forall t \geq 0$. In order to conclude the proof we have to show that also $\rho\mathbb{Z}$ is invariant for (6.50). If $\tilde{\mathbf{x}} \in \rho\mathbb{Z}$ then there is $\mathbf{x} \in \mathbb{Z}$ such that $\tilde{\mathbf{x}} = \rho\mathbf{x}$. From Lemma 6.3, (6.50) and for the fact that $z_{[i]} \in \mathbb{Z}_i$ and $\rho \geq 0$ one has $\bar{\kappa}_i^s(\rho z_{[i]}) = \rho\bar{\kappa}_i^s(z_{[i]})$ hence $\tilde{\mathbf{x}}^+ = \rho\mathbf{A}\mathbf{x} + \rho\mathbf{B}\bar{\boldsymbol{\kappa}}(\mathbf{x}) = \rho\mathbf{x}^+$ and since $\mathbf{x}^+ \in \mathbb{Z}$ then $\tilde{\mathbf{x}}^+ \in \rho\mathbb{Z}$. □

Chapter 7

Plug-and-play MPC: robustness, coupling attenuation and matched nonlinearities

Contents

7.1 Introduction

This chapter is devoted to generalizations of PnP-DeMPC schemes proposed in Chapters 5 and 6 in order to achieve different aims: we will show how to design robust PnP-DeMPC controllers for perturbed LTI systems (Section 7.2) and how to design local controllers for subsystems comprising "matched nonlinearities" (Section 7.4). Moreover, we propose a method for designing PnP-DiMPC regulators. In particular, we show how the on-line exchange of information can be used for designing distributed PnPMPC schemes that are less conservative than those presented in Chapters 5 and 6.

7.2 Robust plug-and-play MPC

We consider a large-scale discrete-time LTI system

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{D}\mathbf{d} \tag{7.1}$$

composed of M subsystems, as in Section 1.5. We assume that each subsystem is equipped with state constraints $\mathbb{X}_i$, input constraints $\mathbb{U}_i$ and model disturbance constraints $\mathbb{D}_i$.

Assumption 7.1. *Sets $\mathbb{D}_i$, $\forall i \in \mathcal{M}$ are C-sets.*

7.2.1 Robust tube-based decentralized MPC

In Chapters 5 and 6, we proposed decentralized tube-based MPC controllers with PnP capabilities. In this section, we propose an algorithm to synthesize a robust PnP controller.
In Chapters 5 and 6, in the spirit of tube-based MPC, we have treated $w_{[i]}$ in (1.3) as a disturbance and defined the nominal subsystem

$$\hat{\Sigma}_{[i]}: \quad \hat{x}^+_{[i]} = A_{ii}\hat{x}_{[i]} + B_i v_{[i]} \tag{7.2}$$

where $v_{[i]}$ is the input. Then we proved asymptotic stability of the closed-loop system and constraints satisfaction by guaranteeing that $\hat{x}_{[i]} \in \hat{\mathbb{X}}_i$ and $v_{[i]} \in \mathbb{V}_i$ where $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$ are suitable sets representing "tightened" constraints (see Sections 5.2.1 and 6.2.1). In order to account for the bounded disturbance $d_{[i]}$ affecting $\Sigma_{[i]}$, we define the nominal subsystem $\hat{\Sigma}_{[i]}$ as in (7.2), the error as $z_{[i]} = x_{[i]} - \hat{x}_{[i]}$ and the control law for subsystem (1.2) as $u_{[i]} = v_{[i]} + \bar{\kappa}_i(z_{[i]})$. Using (1.2), we obtain the dynamics

$$z^+_{[i]} = A_{ii}z_{[i]} + B_i\bar{\kappa}_i(z_{[i]}) + \bar{w}_{[i]} \tag{7.3}$$

where $\bar{w}_{[i]} = w_{[i]} + D_i d_{[i]}$. In other words the error now comprises both the coupling terms $w_{[i]}$ and the disturbance $D_i d_{[i]}$. Therefore, similarly to Section 6.2.1, we can compute an RCI set $\mathbb{Z}_i$ treating $\bar{w}_{[i]}$ as a disturbance in (7.3). Summarizing, in order to design a robust PnP controller, we can execute Algorithm 6.1 substituting $\mathbb{W}_i$ with $\bar{\mathbb{W}}_i = \mathbb{W}_i \oplus D_i \mathbb{D}_i$.
Defining the collective variables as in Section 6.3.4, from (1.3) and (6.4) one obtains the collective model

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{v} + \mathbf{B}\bar{\boldsymbol{\kappa}}(\mathbf{x} - \bar{\mathbf{x}}) + \mathbf{D}\mathbf{d}. \tag{7.4}$$

The next theorem summarizes the key properties of the closed-loop system (7.4).

Theorem 7.1. *Let Assumptions 1.1, 6.1 and 7.1 hold. Assume controllers $\mathcal{C}_{[i]}$ in (6.4) are computed using Algorithm 6.1 by substituting $\mathbb{W}_i$ with $\bar{\mathbb{W}}_i$ and let the function $\bar{\kappa}_i$ be given by (6.15). Then, $\mathbb{Z} = \prod_{i \in \mathcal{M}} \mathbb{Z}_i$ is robustly attractive for the closed-loop system and $\mathbb{X}^N$ defined in (6.17) is a region of attraction of $\mathbb{Z}$. Furthermore, $\mathbf{x}(0) \in \mathbb{X}^N$ guarantees constraints $\mathbf{x}(t) \in \mathbb{X}$ and $\mathbf{u}(t) \in \mathbb{U}$ are fulfilled at all time instants, for all $d_{[i]} \in \mathbb{D}_i$.*

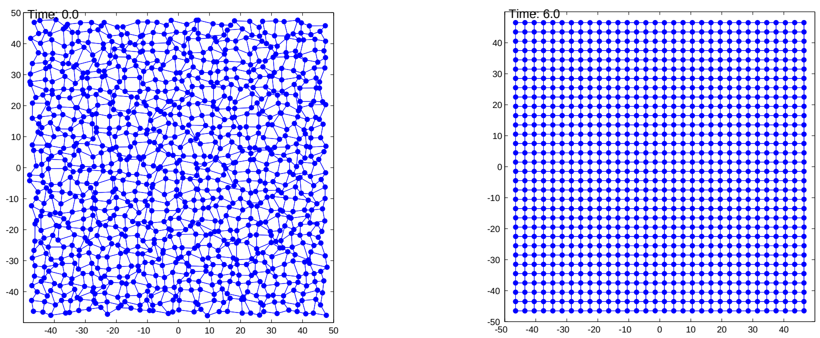
Proof. The proof of Theorem 7.1 is given in Appendix 7.6.1. □

7.2.2 Example: large-scale system

We consider a LSS composed of 1024 masses coupled as in Figure 7.1b through springs and dampers arranged as in Figure 6.9. Each mass $i \in \mathcal{M} = 1 : 1024$, is a subsystem as described in Section 6.6.3. All subsystems are characterized by the same values: $m_i = 8$, $k_i = 0.5$ and $h_i = 0.5$. Subsystems are equipped with the state constraints $||x_{[i,j]}||_\infty \leq 1.5$, $j = 1, 3$, $||x_{[i,l]}||_\infty \leq 0.8$, $i \in \mathcal{M}$, $l = 2, 4$, with the input constraints $||u_{[i]}||_\infty \leq 1$ and with model disturbance $||d_{[i]}||_\infty \leq 0.12$, $D_i = \mathbb{I}_4$. We obtain models $\Sigma_{[i]}$ by discretizing continuous-time models with 0.2 sec sampling time, using zero-order hold discretization for the local dynamics and treating $x_{[j]}$, $j \in \mathcal{N}_i$ as exogenous signals [FCS13]. We synthesized controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ using Algorithm 6.1 replacing $\mathbb{W}_i$ with $\bar{\mathbb{W}}_i$. In Figure 7.1 we show a simulation[1] where, at time $t = 0$, the masses are still and placed as in Figure 7.1a. At all time steps t, the control action $u_{[i]}(t)$ computed by the controllers $\mathcal{C}_{[i]}$, for all $i \in \mathcal{M}$, is kept constant during the sampling

[1] All simulations have been done using a Linux distribution (Kubuntu 12.04), with processor Intel Core i7-2600, 3.4 GHz, MatLab r2011b, solver CPLEX [IBM11], YALMIP [LÖ4] and MPT [KGB04].

interval and applied to the continuous-time system. Convergence can not be obtained for all masses to their equilibrium position since a bounded disturbance acts on each subsystem. However all subsystems fulfill input and state constraints. State and input variables are depicted in Figure 7.2.



(a) Position of the masses at initial time. (b) Position of the masses at time 6 sec.

Figure 7.1: Position of the 1024 masses on the plane.

7.3 Distributed plug-and-play MPC

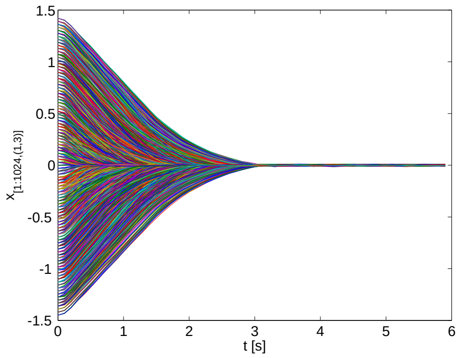
We consider a large-scale discrete-time LTI system

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{7.5}$$

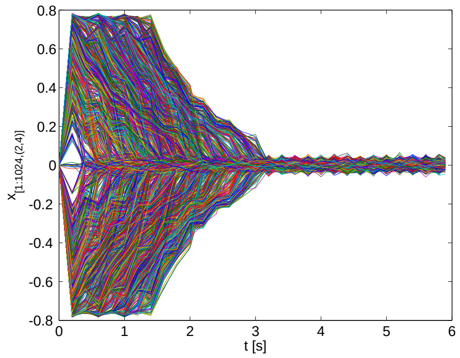
composed of M subsystems, as described in Section 1.5. We assume that each subsystem is equipped with state constraints $\mathbb{X}_i$ and input constraints $\mathbb{U}_i$, defined as in (6.2).

7.3.1 Tube-base distributed MPC

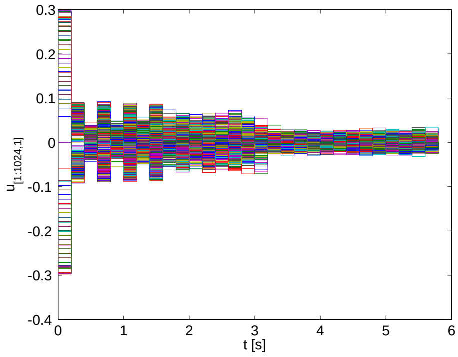
In this section, we propose a distributed architecture generalizing decentralized regulators described in Chapters 5 and 6, i.e. DeMPC regulators with PnP capabilities. We already considered distributed controllers in Section 6.5. However, in that case local controllers $\mathcal{C}^{dis}_{[i]}$ where designed assuming a decentralized implementation and modified *a posteriori* for tacking advantage of the states $x_{[j]}$, $j \in \mathcal{N}_i$ transmitted by parent subsystems. Next we propose a different approach where we exploit knowledge of parents' states in the design phase. More precisely, similarly to Section 4.2.1, we



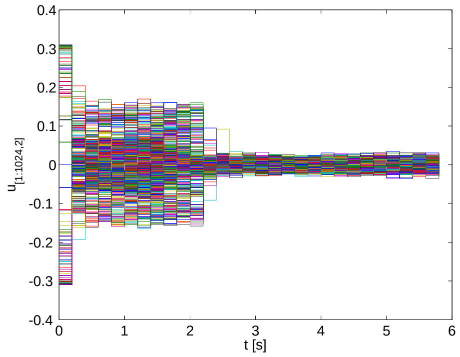
(a) Displacements of the masses with respect their equilibrium positions.



(b) Velocities, i.e. states $x_{[i,2]}$ and $x_{[i,4]}$, $i \in \mathcal{M}$.



(c) Inputs $u_{[i,1]}$, $i \in \mathcal{M}$.



(d) Inputs $u_{[i,2]}$, $i \in \mathcal{M}$.

Figure 7.2: State and input trajectories of the 1024 masses with $\mathbf{x}(0)$ as in Figure 7.1a.

will use these additional pieces of information for attenuating coupling between subsystems.
Consider controller

$$\mathcal{C}_{[i]}^{dis}: \qquad u_{[i]} = u_{[i]}^{dec} + \sum_{j\in\mathcal{N}_i} \delta_{ij} K_{ij} x_{[j]} \tag{7.6}$$

where $u_{[i]}^{dec} \in \mathbb{R}^{m_i}$ collects the terms of the decentralized controller (5.4) or (6.4), $K_{ij} \in \mathbb{R}^{m_i \times n_j}$ and $\delta_{ij} \in \{0,1\}$, $i,j \in \mathcal{M}$. Note that, if $\delta_{ij} = 0$, $\forall i \in \mathcal{M}, \forall j \in \mathcal{N}_i$, the control scheme is completely decentralized, since $u_{[i]} = u_{[i]}^{dec}$, hence controller $\mathcal{C}_{[i]}^{dis}$ is defined as in (5.4) or (6.4). In the spirt of tube-based MPC, we define the nominal model $\hat{\Sigma}_{[i]}$ as in (5.3) and (6.3). Using controller (7.6) in (1.3), the coupling term for subsystem $\Sigma_{[i]}$ can be written as

$$\hat{w}_{[i]} = w_{[i]} + B_i \sum_{j\in\mathcal{N}_i} \delta_{ij} K_{ij} x_{[j]} = \sum_{j\in\mathcal{N}_i} (A_{ij} + \delta_{ij} B_i K_{ij}) x_{[j]} \tag{7.7}$$

and verifies the constraints

$$\hat{w}_{[i]} \in \hat{\mathbb{W}}_i = \bigoplus_{j\in\mathcal{N}_i} (A_{ij} + \delta_{ij} B_i K_{ij}) \mathbb{X}_j. \tag{7.8}$$

Moreover, tightened state constraints $\hat{\mathbb{X}}_i$ defined for the nominal subsystems $\hat{\Sigma}_{[i]}$ (see (5.3) and (6.3)) do not depend on the choice of matrices K_{ij}. They only impact on the size of tightened input constraints $\mathbb{V}_i$, as, from (7.6), $\mathbb{V}_i$ must be a nonempty set verifying

$$\mathbb{V}_i \subseteq \mathbb{U}_i \ominus \left(\bar{\mathbb{U}}_i \oplus \bigoplus_{j\in\mathcal{N}_i} \delta_{ij} K_{ij} \mathbb{X}_j \right) \tag{7.9}$$

where, from (5.11b), $\bar{\mathbb{U}}_i = K_i \mathbb{Z}_i$ and, from (6.5b), $\bar{\mathbb{U}}_i = \mathbb{U}_{z_i}$. In Algorithm 7.1 we summarize the steps needed to design a distributed PnP controller.

Algorithm 7.1 Design of the distributed controller $\mathcal{C}^{dis}_{[i]}$ defined in (7.6)

Input: A_{ii}, B_i, $\mathbb{X}_i$, $\mathbb{U}_i$, $\mathcal{N}_i$, $\{\delta_{ij}\}_{j\in\mathcal{N}_i}$, $\{A_{ij}\}_{j\in\mathcal{N}_i}$, $\{\mathbb{X}_j\}_{j\in\mathcal{N}_i}$
Output: controller $\mathcal{C}^{dis}_{[i]}$ in (7.6)

(I) $\forall j \in \mathcal{N}_i$, if $\delta_{ij} = 1$, compute the matrix K_{ij} solving

$$\min_{K_{ij}} ||\mathcal{C}_{x_i} (A_{ij} + B_i K_{ij}) \mathcal{C}^{\flat}_{x_j}||_p \tag{7.10a}$$

$$||K_{ij} \mathcal{C}^{\flat}_{x_j}||_p \leq \xi_{ij} \tag{7.10b}$$

where p is a generic norm and $\mathcal{C}_{x_i} = (c^T_{x_{i,1}}, \dots, c^T_{x_{i,\tau^x_i}}) \in \mathbb{R}^{\tau^x_i \times n_i}$.

(II) Execute Algorithm 5.1 or 6.1, where $\mathbb{V}_i$ must fulfill (7.9) and the disturbance $w_{[i]}$ is defined as in (7.7).

In Step (I) of Algorithm 7.1, if $\delta_{ij} = 1$ the computation of matrix K_{ij} is required. We propose to compute K_{ij} in order to reduce as much as possible the magnitude of coupling terms $A_{ij} + B_i K_{ij}$, hence to decrease

the size of set $\hat{\mathbb{W}}_i$. However, if K_{ij} is "too aggressive" we could not find a nonempty set $\mathbb{V}_i$ in Step (II). Therefore we propose to bound set $K_{ij}\mathbb{X}_j$ using scalars ξ_{ij} in (7.10b). We highlight that the minimization of $||\mathcal{C}_{x_i}(A_{ij}+B_iK_{ij})\mathcal{C}_{x_j}^{\flat}||_1$ in (7.10) amounts to an LP problem and the minimization of $||\mathcal{C}_{x_i}(A_{ij}+B_iK_{ij})\mathcal{C}_{x_j}^{\flat}||_F$ can be recast into a QP problem. So far, the parameters δ_{ij} have been considered fixed. However, if in Step (II) one obtains $K_{ij}=\mathbf{0}_{m_i\times n_j}$ for some $j\in\mathcal{N}_i$, it is impossible to reduce the magnitude of the coupling term $A_{ij}+B_iK_{ij}$ and the knowledge of $x_{[j]}$ is useless for controller $\mathcal{C}_{[i]}^{dis}$. This suggests to revise the choice of δ_{ij} and set $\delta_{ij}=0$.
It is possible to show that if Algorithm 7.1 does not stop in Step (II) for all subsystems, then, for the closed-loop system, state and input constraints satisfaction and asymptotic stability of the origin are guaranteed. The main ideas for proving stability are the following ones. We can always rewrite system (7.5) equipped with distributed controllers (7.6) as a system equipped with decentralized controllers proposed in (5.4) or (6.4). Indeed we can redefine the coupling terms among subsystems as in (7.7) and input constraints as in (7.9) and then we can prove asymptotic stability of the origin and constraint satisfaction using Theorems 5.1 and 6.1. For plug-in and unplugging operations, one can use procedures at all similar to those described in Sections 5.4 and 6.4.

7.3.2 Example: power network system

In this section, we apply the proposed DiMPC scheme to the PNS described in Appendix B. In particular we will show advantages brought about by the proposed PnP-DiMPC to respect to PnP-DeMPC proposed in Chapter 5. A similar comparison can be carried out to respect to PnP-DeMPC proposed in Chapter 6. We rewrite matrices A_{ij}, $\forall i,j\in\mathcal{M}$ in (B.2) as

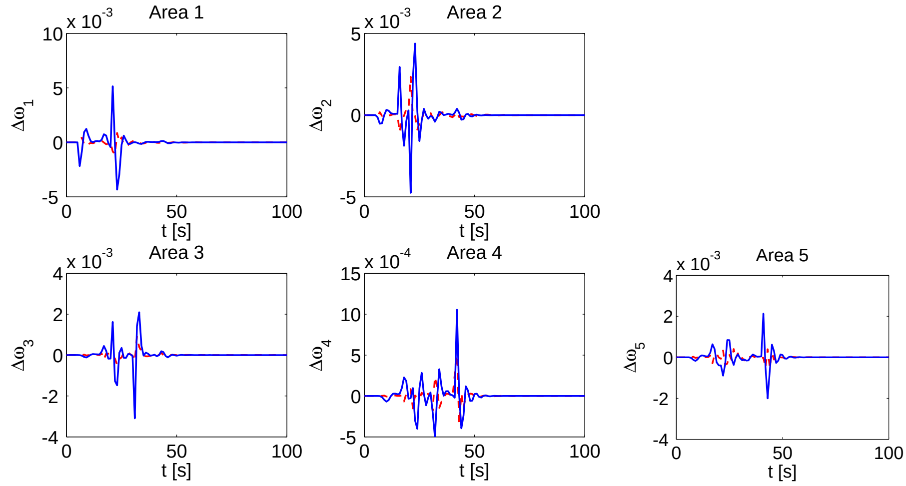
$$\begin{aligned}
A_{ii}(\{\gamma P_{ij}\}_{j\in\mathcal{N}_i}) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{\sum_{j\in\mathcal{N}_i}\gamma P_{ij}}{2H_i} & -\frac{D_i}{2H_i} & \frac{1}{2H_i} & 0 \\ 0 & 0 & -\frac{1}{T_{t_i}} & \frac{1}{T_{t_i}} \\ 0 & -\frac{1}{R_iT_{g_i}} & 0 & -\frac{1}{T_{g_i}} \end{bmatrix} \\
A_{ij} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{\gamma P_{ij}}{2H_i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.
\end{aligned} \tag{7.11}$$

Differently from (B.2), in (7.11) the parameter $\gamma > 0$ allows us to increase or decrease coupling among areas. We consider the PNS and local power loads used in Scenario 2 in Section B.1.2. In order to apply the PnP-DeMPC method in Chapter 5, we assume that the input of each area is not constrained.

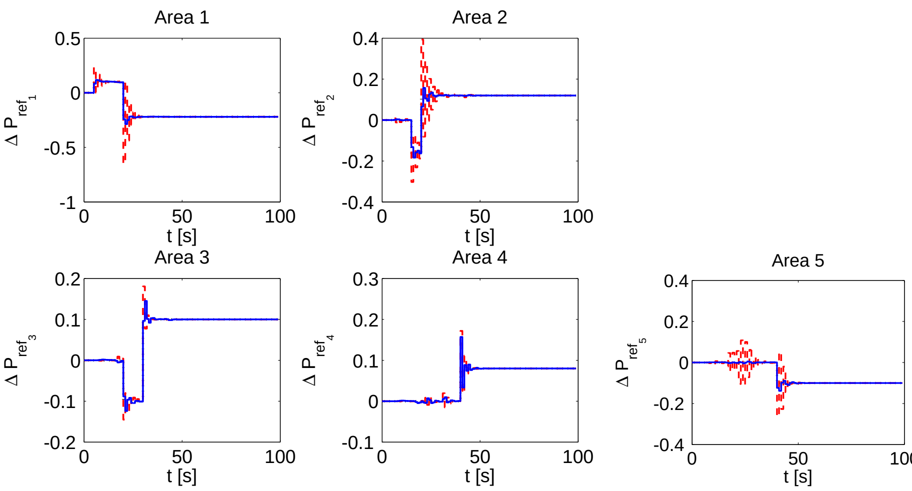
In order to design PnP controllers we need to execute Step (I) of Algorithm 7.1. First, we consider $\delta_{ij} = 0$, $\forall i, j \in \mathcal{M}$, and therefore controllers $\mathcal{C}_{[i]}$ are completely decentralized. In Step (II) of Algorithm 7.1 we execute Algorithm 5.1 in order to design a DeMPC regulator for each area. Increasing the parameter γ also increases the magnitude of coupling terms A_{ij} and, for $\gamma \geq 2.4$, Algorithm 7.1 stops in Step (II). By setting $\delta_{ij} = 1$ and $\xi_{ij} = \infty$, $\forall i \in 1:5, \forall j \in \mathcal{N}_i$, we can compute matrices K_{ij} for reducing coupling (in (7.10) we use the norm $p = \infty$). The design of distributed controllers $\mathcal{C}_{[i]}^{dis}$ in (7.6) can be successfully performed through Algorithm 7.1 for $\gamma \leq 4$, and this shows the benefits of a distributed architecture.

In the following, we compare the performance of PnP-DeMPC in (5.4) and PnP-DiMPC in (7.6) for $\gamma = 1.4$. In order to guarantee stability and feasibility of local MPC regulators, we design MPC-i, $i \in \mathcal{M}$, using a zero terminal constraint (see Section B.2). Figures 7.3 and 7.4 show the results of control experiments[2] with $T_{sim} = 100$ and load steps as in Table B.4. The performance indices for PnP-DeMPC are $\eta_{de} = 0.0405$ and $\Phi_{de} = 0.0061$. PnP-DiMPC gives $\eta_{di} = 0.1953$ and $\Phi_{di} = 0.0026$. In terms of the index η, decentralized control outperforms distributed control. As shown in Figure 7.3, this is mainly due to the fact that when an area is affected by a load step, parent areas do not change their ΔP_{ref}. However, this also causes bigger variations in the frequency deviation $\Delta\omega$. With distributed controllers, each area can compensate in advance power loads of parent areas by modifying local power production and this leads to smaller oscillations in frequency deviations. For instance, at time 25, area 5 helps area 2 by varying ΔP_{ref_5} in order to counteract the load step ΔP_{L_2} occurred at time $t = 20$. In terms of power transfers, decentralized control is worse than distributed control since $\Phi_{de} > \Phi_{di}$. This can also be noticed in Figure 7.4 showing power transfer. If, as in nation-wide power networks, the cost of exchanging power is higher than the cost of producing power, then distributed control achieves better economic performance.

[2] All simulations have been done using a MacOS 10.7.5, with processor Intel Core i5, 1.7 GHz, MatLab r2013a, solver CPLEX [IBM11], YALMIP [Lö04] and MPT [KGB04].



(a) Frequency deviation in each area.



(b) Load reference set-point in each area.

Figure 7.3: Control experiments with decentralized (continuous lines) and distributed (dashed lines) PnPMPC: 7.3a frequency deviation and 7.3b inputs for each area.

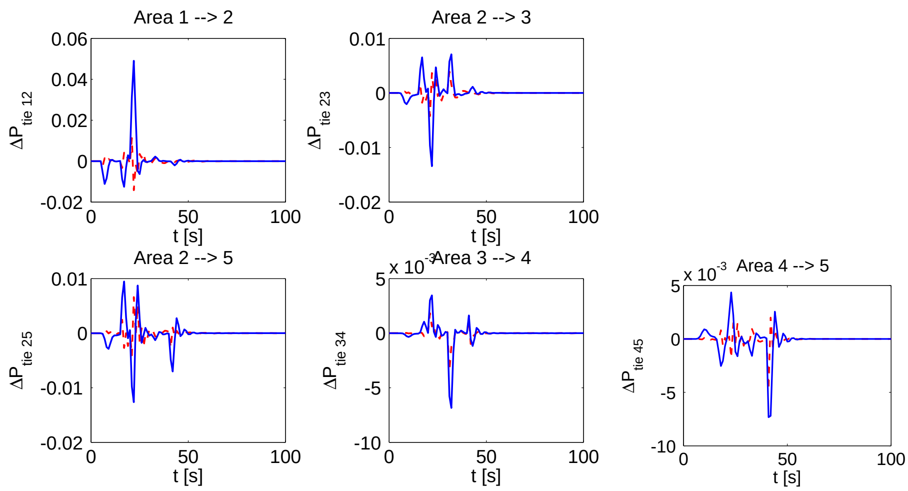


Figure 7.4: Tie-line power between connected areas when PnP-DeMPC (continuous line) or PnP-DiMPC (dashed line) are used.

7.4 Distributed plug-and-play MPC for a class of nonlinear systems

We consider a LSS partitioned in M subsystems, where the $i-th$ subsystem is described by the state equation

$$\begin{aligned}\Sigma_{[i]}: \quad x_{[i]}^{+} =& A_{ii}x_{[i]} + B_i[g_i(x_{[i]}, \{x_{[j]}\}_{j\in\mathcal{N}_i})u_{[i]}+\\ &+ h_i(x_{[i]}, \{x_{[j]}\}_{j\in\mathcal{N}_i})] + \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]}\end{aligned} \tag{7.12}$$

where, differently from (1.2), we define the set of parents to subsystem i as $\mathcal{N}_i = \{j \in \mathcal{M} : \frac{\partial x_{[i]}^{+}}{\partial x_{[j]}} \neq \mathbf{0}_{n_i}, i \neq j\}$. Furthermore, $h_i(\cdot) : \mathbb{R}^{n_i+\sum_{j\in\mathcal{N}_i} n_j} \rightarrow \mathbb{R}^{m_i}$, and we assume $g_i(\cdot) : \mathbb{R}^{n_i+\sum_{j\in\mathcal{N}_i} n_j} \rightarrow \mathbb{R}$ is invertible.

7.4.1 Nonlinear tube-based distributed MPC

In the spirit of tube-based MPC, as in [RTMA06], we define a nominal model for each subsystem as

$$\hat{\Sigma}_{[i]}: \quad \hat{x}_{[i]}^{+} = A_{ii}\hat{x}_{[i]} + B_i v_{[i]} \tag{7.13}$$

where $v_{[i]}$ is the input. As in [RTMA06] our goal is to relate inputs $v_{[i]}$ in (7.13) to $u_{[i]}$ in (7.12) and compute sets $\mathbb{Z}_i \subseteq \mathbb{R}^{n_i}$, $i \in \mathcal{M}$ such that

$$x_{[i]}(0) \in \hat{x}_{[i]}(0) \oplus \mathbb{Z}_i \Rightarrow x_{[i]}(t) \in \hat{x}_{[i]}(t) \oplus \mathbb{Z}_i,\ \forall t \geq 0.$$

In other words, as in Chapters 5 and 6, we want to confine $x_{[i]}(t)$ in a tube around $\hat{x}_{[i]}(t)$ of section $\mathbb{Z}_i$. Therefore, if $x_{[i]} \in \mathbb{Z}_i$ there exists $u_{[i]} = \bar{\kappa}_i(x_{[i]}) : \mathbb{Z}_i \to \mathbb{U}_{z_i}$ such that $x^+_{[i]} \in \mathbb{Z}_i$, $\forall x_{[j]} \in \mathbb{X}_j$, $j \neq i$. Moreover if $x_{[i]} \in \hat{x}_{[i]} \oplus \mathbb{Z}_i$ and one uses the controller

$$\begin{aligned}\mathcal{C}^{NL}_{[i]}: \quad u_{[i]} = g_i(x_{[i]}, \{x_{[j]}\}_{j\in\mathcal{N}_i})^{-1}[-h_i(x_{[i]}, \{x_{[j]}\}_{j\in\mathcal{N}_i})+ \\ +v_{[i]} + \bar{\kappa}_i(x_{[i]} - \bar{x}_{[i]}) + \sum_{j\in\mathcal{N}_i} K_{ij}x_{[j]}]\end{aligned} \tag{7.14}$$

where $K_{ij} \in \mathbb{R}^{m_i \times n_j}$, then, for all $v_{[i]}$, one has $x^+_{[i]} \in \hat{x}^+_{[i]} \oplus \mathbb{Z}_i$. Controller $\mathcal{C}^{NL}_{[i]}$ is based on the well-known idea of canceling nonlinearities in the state equations. This is possible because in (7.12) nonlinear terms are “matched”, i.e. they can be directly modified through the control input $u_{[i]}$. Following [RTMA06], the next goal is to compute tightened constraints $\hat{\mathbb{X}}_i \subseteq \mathbb{X}_i$ and $\mathbb{V}_i \subseteq \mathbb{U}_i$ for guaranteeing $\hat{x}_{[i]} \in \hat{\mathbb{X}}_i$ and $v_{[i]} \in \mathbb{V}_i$, at all time instants. Tightened state constraints must satisfy the following inclusions

$$\begin{aligned}&\hat{\mathbb{X}}_i \oplus \mathbb{Z}_i \subseteq \mathbb{X}_i \\ &\mathbb{G}_i \left(\mathbb{V}_i \oplus \mathbb{U}_{z_i} \oplus \bigoplus \delta_{ij} K_{ij} \mathbb{X}_j \oplus \mathbb{H}_i\right) \subseteq \mathbb{U}_i\end{aligned}$$

where $\mathbb{G}_i = g_i(\mathbb{X}_i, \{\mathbb{X}_j\}_{j\in\mathcal{N}_i})^{-1}$ and $\mathbb{H}_i = h_i(\mathbb{X}_i, \{\mathbb{X}_j\}_{j\in\mathcal{N}_i})$. Obviously, as in nonlinear tube-based MPC theory, the most difficult step is the evaluation of sets $\mathbb{G}_i$ and $\mathbb{H}_i$. Estimates of these sets can be obtained using methods of reachability analysis for nonlinear systems, as those discussed in [RRS+12]. Since we need to stabilize the nominal subsystems (7.13) and to guarantee satisfaction of tightened state constraints, we need to solve a suitable local MPC problem. In summary, similar to Chapter 6, the controller $\mathcal{C}^{NL}_{[i]}$ is given by (7.14), (6.6) and (5.13) and it is distributed since it depends upon quantities of subsystem $\Sigma_{[i]}$ and parents' subsystems only. Therefore, using (7.12) and (7.14), and defining $\hat{\mathbf{x}} = (\hat{x}_{[1]}, \dots, \hat{x}_{[M]}) \in \mathbb{R}^n$, $\mathbf{v} = (v_{[1]}, \dots, v_{[M]}) \in \mathbb{R}^m$, $\bar{\boldsymbol{\kappa}}(\mathbf{x}) = (\bar{\kappa}_1(x_{[1]}), \dots, \bar{\kappa}_M(x_{[M]})) : \mathbb{R}^n \to \mathbb{R}^m$ and $\mathbf{K}$ that collects matrices K_{ij}, we obtain the closed-loop system

$$\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{v} + \mathbf{B}\bar{\boldsymbol{\kappa}}(\mathbf{x} - \hat{\mathbf{x}}) + \mathbf{B}\mathbf{K}\mathbf{x}. \tag{7.15}$$

We note that, thanks the choice of nonlinear controllers $\mathcal{C}_{[i]}^{NL}$, we obtained a closed-loop system easier to analyze: if $\delta_{ij} = 0$, for all $i \in \mathcal{M}$ and $j \in \mathcal{N}_i$, (7.15) is equivalent to (6.16), if at least one $\delta_{ij} \neq 0$, (7.15) is equivalent to the closed-loop system analyzed in Section 7.3. Therefore, asymptotic stability of the origin of the closed-loop system and constraint satisfaction can be guaranteed. Moreover if each subsystem is affected by bounded model disturbances, we can adopt the method proposed in Section 7.2 in order to design a robust version of local controllers. Since we can design controller (7.14) through Algorithm 7.1, distributed nonlinear controllers $\mathcal{C}_{[i]}^{NL}$ can be synthesized in a PnP fashion. We will illustrate PnP operations in the next section through an example.

7.4.2 Examples

We consider the nonlinear discrete-time system (7.16) composed by the 4 subsystems shown in Figure 7.5.
We assume that each subsystem is equipped with the following constraints $\mathbb{X}_i = \{x_{[i]} \in \mathbb{R}^{n_i} : ||x||_\infty \leq 1\}$, $\forall i = 1:4$ and $\mathbb{U}_i = \{u_{[i]} \in \mathbb{R}^{m_i} : ||u||_\infty \leq 2\}$, $\forall i = 1:3$. Moreover subsystem 4 is affected by disturbances $d_{[4]} \in \mathbb{R}^2$ bounded in the set $\mathbb{D}_4 = \{d_{[4]} \in \mathbb{R}^2 : ||d||_\infty \leq 0.1\}$. We note that (7.16) is in the form of (7.12). In the following we consider two scenarios.

Scenario 1

In this scenario we consider that system (7.16) is composed of subsystems $\Sigma_{[1]}$, $\Sigma_{[2]}$ and $\Sigma_{[3]}$ only. We design local controllers $\mathcal{C}_{[i]}^{NL}$, $i = 1:3$ setting $\delta_{ij} = 0$, $i \neq j$, therefore local controllers $\mathcal{C}_{[i]}^{NL}$ are decentralized, since they depends on local quantities only. In Figure 7.6 we show a control experiment for system proposed in Scenario 1. We highlight that the state $\mathbf{x}$ and the input $\mathbf{u}$ go to zero asymptotically.

Scenario 2

In this scenario we consider the system of Scenario 1 where subsystem $\Sigma_{[4]}$ wants to plug in. Since the state of subsystem $\Sigma_{[4]}$ acts on subsystems $\Sigma_{[1]}$ and $\Sigma_{[3]}$, local controllers $\mathcal{C}_{[1]}^{NL}$ and $\mathcal{C}_{[3]}^{NL}$ must be retuned. We set $\delta_{s,4} = 0$, $s = \{1,3\}$, $\delta_{4,j} = 1$, $j = \{1,3\}$ and $\xi_{4j} = \infty$. In Step (I) of Algorithm 7.1, we obtained matrices K_{41} and K_{43} such that $A_{41} + B_4K_{41}$ and $A_{43} + B_4K_{43}$ are zero. Note that, differently from local controllers $\mathcal{C}_{[i]}^{NL}$, $i = 1:3$, local controller $\mathcal{C}_{[4]}^{NL}$ is distributed since $u_{[4]}$ depends on

$$x_{[1]}^{+} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{[1]} + \begin{bmatrix} 0.1 & 0.02 & 0 & 0.08 & 0 \\ -0.2 & 0 & 0.1 & -0.08 & 0.02 \end{bmatrix} \begin{bmatrix} x_{[2]} \\ x_{[4]} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tilde{x}_{[1]} \tag{7.16a}$$

$$\text{where } \tilde{x}_{[1]} = |\sum_{\substack{j \in \mathcal{N}_1 \\ k \in 1:n_j}} x_{[j,k]} - 5| u_{[1]} + x_{[2,1]}^3$$

$$x_{[2]}^{+} = \begin{bmatrix} 0.1 & 1 & 0.3 \\ -0.1 & -1 & 0.5 \\ 0.6 & 0.7 & 0.8 \end{bmatrix} x_{[2]} + \begin{bmatrix} 0.1 & -0.2 & 0.05 & -0.07 \\ 0.02 & 0 & 0.02 & 0 \\ 0 & 0.1 & 0 & 0.06 \end{bmatrix} \begin{bmatrix} x_{[1]} \\ x_{[3]} \end{bmatrix} + \begin{bmatrix} 1 \\ 0.5 \\ 1 \end{bmatrix} \tilde{x}_{[2]} \tag{7.16b}$$

$$\text{where } \tilde{x}_{[2]} = \left(\prod_{\substack{j \in \mathcal{N}_2 \\ k \in 1:n_j}} x_{[j,k]} - 5 \right) u_{[2]} + (x_{[1,2]} x_{[3,2]})^5$$

$$x_{[3]}^{+} = \begin{bmatrix} 1 & 0.1 \\ 1 & 1 \end{bmatrix} x_{[3]} + \left[\begin{array}{ccc|cc} 0.05 & 0.02 & 0 & 0.01 & 0 \\ -0.07 & 0 & 0.06 & 0.01 & 0.02 \end{array}\right] \begin{bmatrix} x_{[2]} \\ x_{[4]} \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} \tilde{x}_{[3]} \tag{7.16c}$$

$$\text{where } \tilde{x}_{[3]} = \left(\sqrt{\sum_{\substack{j \in \mathcal{N}_3 \\ k \in 1:n_j}} x_{[j,k]}^2} - 3 \right) u_{[3]} + \sin(x_{[2,3]}) \cos(x_{[2,1]})$$

$$x_{[4]}^{+} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} x_{[4]} + \left[\begin{array}{cc|cc} 0.08 & 0 & 0.01 & 0.02 \\ -0.08 & 0.02 & 0.01 & 0.02 \end{array}\right] \begin{bmatrix} x_{[1]} \\ x_{[3]} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tilde{x}_{[4]} + d_{[4]} \tag{7.16d}$$

$$\text{where } \tilde{x}_{[4]} = \left(\sum_{\substack{j \in \mathcal{N}_4 \\ k \in 1:n_j}} x_{[j,k]} - 5 \right) u_{[4]} + \begin{bmatrix} 0.5 e^{x_{[1,1]}} \sin(x_{[3,1]}) \\ 0.5 e^{x_{[1,2]}} \sin(x_{[3,2]}) \end{bmatrix}$$

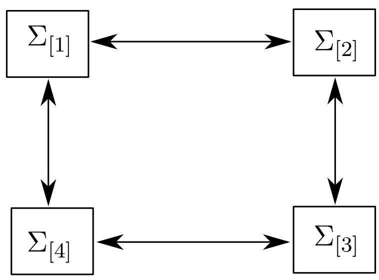
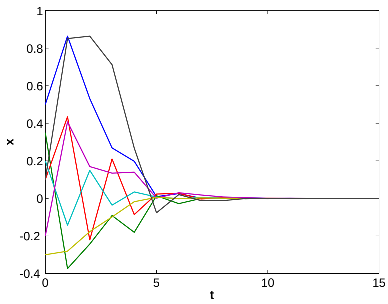
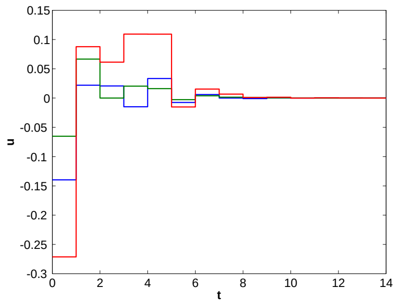


Figure 7.5: Nonlinear system composed of 4 subsystems.

states of parents' subsystems. In Figure 7.7 we show a control experiment for system proposed in Scenario 2. We highlight that the state $\mathbf{x}$ and the input $\mathbf{u}$ cannot go to zero asymptotically since subsystem 4 is affected by

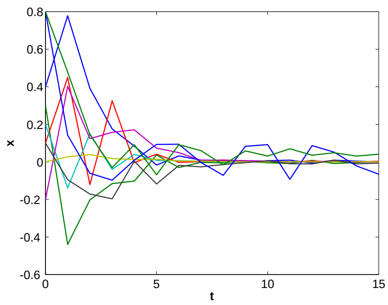


(a) State **x** of system considered in Scenario 1.

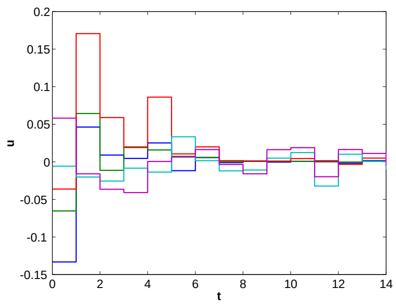


(b) Input **u** of system considered in Scenario 1.

Figure 7.6: Control experiments for Scenario 1.



(a) State **x** of system considered in Scenario 2.



(b) Input **u** of system considered in Scenario 2.

Figure 7.7: Control experiments for Scenario 2.

bounded disturbances, therefore the state of each subsystem 4 is kept in a tube as highlighted in Section 7.2.

7.5 Final comments

In this chapter we proposed various improvements for the PnP-DeMPC regulators introduced in previous chapters. We introduced methods to design robust controllers and to design controllers for subsystems described by dynamics with matched nonlinearities. Most importantly, we proposed a distributed PnPMPC controller that is less conservative than the PnP-DeMPC regulators studied in previous chapters, as the exchanged online information can be used for reducing the magnitude of the coupling among subsystems.

7.6 Appendix

7.6.1 Proof of Theorem 7.1

Proof. The proof is a straightforward adaptation of the Step 1 of the proof of Theorem 6.1 (see the Appendix 6.8). We report it below for the sake of completeness. The first part of the proof uses arguments similar to the ones adopted for proving Theorem 1 both in [FS12] and Chapter 5. Indeed we can easily prove that optimizers $\hat{x}_{[i]}(0|t)$ and $v_{[i]}(0|t)$ of the MPC-i optimization problem defined in Step (IIIiii) of Algorithm 6.1 verify $\hat{x}_{[i]}(0|t) \to \mathbf{0}_{n_i}$ and $v_{[i]}(0|t) \to \mathbf{0}_{m_i}$ as $t \to \infty$. We highlight that constraint (5.13b) in the MPC-i problem is replaced by (6.7).
Next, we prove that $\mathrm{dist}(\mathbb{Z}, \mathbf{x}(t)) \to 0$, as $t \to \infty$. To this purpose, we prove that if $\mathbf{x}(0) \in \mathbb{X}^N$ there is $\tilde{T} > 0$ such that $\mathbf{x}(\tilde{T}) \in \mathbb{Z}$ and hence $\mathrm{dist}(\mathbb{Z}, \mathbf{x}(\tilde{T})) = 0$.
Recalling that the state $\mathbf{x}(t)$ evolves according to the equation (7.4), we can write

$$\mathbf{x}(t+1) = \mathbf{A_D}\mathbf{x}(t) + \mathbf{B}\bar{\kappa}(\mathbf{x}(t)) + \mathbf{A_c}\mathbf{x}(t) + \bar{\boldsymbol{\eta}}(t) + \mathbf{Dd} \tag{7.16}$$

where $\mathbf{A_D} = \mathrm{diag}(A_{11}, \dots, A_{MM})$, $\mathbf{A_C} = \mathbf{A} - \mathbf{A_D}$,

$$\bar{\boldsymbol{\eta}}(t) = \mathbf{B}(\mathbf{v}(t) + \bar{\boldsymbol{\kappa}}(\mathbf{z}(t)) - \bar{\boldsymbol{\kappa}}(\mathbf{x}(t))) \tag{7.17}$$

and $\mathbf{z}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(0|t)$. In particular, if $\mathbf{x}(0) \in \mathbb{X}^N$, recursive feasibility shown above implies that (7.16) holds for all $t \geq 0$.
Note that in view of Assumption 6.3, the LP problem (6.14) is feasible for all $z_{[i]} \in \mathbb{R}^{n_i}$. Indeed (6.14c) and (6.14e) require that there are $\mu > 0$ and $\bar{z}^s_{[i]} \in \mu\bar{\mathbb{Z}}^s_i$, $s \in 0 : k_i - 1$ such that $z_{[i]} = \sum_{i=0}^{k_i-1} \bar{z}^s_{[i]}$ and since $\bar{\mathbb{Z}}^0_i \supset B_{\omega_i}(0)$ (i.e. $\bar{\mathbb{Z}}^0_i$ is full dimensional), these quantities always exist. This implies that the function $\bar{\boldsymbol{\kappa}}(\mathbf{x}(t))$ in (7.16) is always well defined.
We already proved the asymptotic convergence to zero of the nominal state $\hat{\mathbf{x}}(0|t)$ and the input signal $\mathbf{v}(0|t)$ and hence it holds

$$\forall \delta > 0,\ \exists T_1 > 0 :\ ||\hat{\mathbf{x}}(0|t)|| \leq \delta \text{ and } ||\mathbf{v}(0|t)|| \leq \delta,\ \forall t \geq T_1. \tag{7.18}$$

Moreover, according to [Gal95], we can assume without loss of generality that $\bar{\kappa}_i(\cdot)$ is a continuous piecewise affine map. In view of this, $\bar{\kappa}(\cdot)$ is also globally Lipschitz, i.e.

$$\exists\ L > 0\ :\ ||\bar{\boldsymbol{\kappa}}(\mathbf{x} - \hat{\mathbf{x}}) - \bar{\boldsymbol{\kappa}}(\mathbf{x})|| \leq L||\hat{\mathbf{x}}|| \tag{7.19}$$

for all $(\mathbf{x}, \hat{\mathbf{x}})$ such that $\mathbf{x} \in \mathbb{X}$ and $\mathbf{x} - \hat{\mathbf{x}} \in \mathbb{Z}$. Using (7.19) one can show that, for all $\epsilon > 0$, setting $\delta = \frac{\epsilon}{||B||(1+L)}$ the following implication holds

$$||\hat{\mathbf{x}}(0|t)|| \leq \delta \text{ and } ||\mathbf{v}(0|t)|| \leq \delta \Rightarrow ||\bar{\boldsymbol{\eta}}(t)|| \leq \epsilon, \ \forall x(t) \in \mathbb{X}.$$

Therefore, from (7.18),

$$\forall \epsilon > 0, \ \exists T_1 > 0 : \ ||\bar{\boldsymbol{\eta}}(t)|| \leq \epsilon, \ \forall t \geq T_1. \tag{7.20}$$

Since $||\hat{\mathbf{x}}(0|t)|| \to \mathbf{0}_n$, as $t \to \infty$, and $\mathbb{Z}$ contains $\prod_{i=1}^{M} B_{\omega_i}(\mathbf{0}_{n_i})$, then

$$\forall \delta_z > 0, \ \exists T_2 > 0 : \ \hat{\mathbf{x}}(0|t) \in \delta_z \mathbb{Z}, \ \forall t \geq T_2 \tag{7.21}$$

and hence, from (6.7),

$$\mathbf{x}(t) = \hat{\mathbf{x}}(0|t) + (\mathbf{x}(t) - \hat{\mathbf{x}}(0|t)) \in (1 + \delta_z)\mathbb{Z}, \ \forall t \geq T_2. \tag{7.22}$$

From (7.16) we have, for all $i \in \mathcal{M}$,

$$x_{[i]}(t+1) = A_{ii} x_{[i]}(t) + B_i \bar{\kappa}_i(x_{[i]}(t)) + \tilde{w}_{[i]}(t) \tag{7.23}$$

where $\tilde{w}_{[i]} = \sum_{j \in \mathcal{N}_i} A_{ij} x_{[j]} + \bar{\eta}_{[i]} + D_i d_{[i]}, \forall i \in \mathcal{M}$. Setting $\bar{T} = \max\{T_1, T_2\}$ and using (7.20) and (7.22), one has, $\forall t \geq \bar{T}$

$$\tilde{w}_{[i]} \in (1 + \delta_z) \bigoplus_{j \in \mathcal{N}_i} A_{ij} \mathbb{Z}_j \oplus B_\epsilon(\mathbf{0}_{n_i}) \oplus D_i \mathbb{D}_i. \tag{7.24}$$

From Assumption 6.2 we have

$$D_i \mathbb{D}_i \oplus \bigoplus_{j \in \mathcal{N}_i} A_{ij} \mathbb{Z}_j \subseteq D_i \mathbb{D}_i \oplus \bigoplus_{j \in \mathcal{N}_i} A_{ij} \left[\mathbb{X}_j \ominus B_{\rho_{j,1}}(\mathbf{0}_{n_j}) \right] \tag{7.25a}$$

$$\subseteq D_i \mathbb{D}_i \oplus \bigoplus_{j \in \mathcal{N}_i} \left[(A_{ij} \mathbb{X}_j) \ominus \left(A_{ij} B_{\rho_{j,1}}(\mathbf{0}_{n_j}) \right) \right] \tag{7.25b}$$

$$\subseteq D_i \mathbb{D}_i \oplus \left(\bigoplus_{j \in \mathcal{N}_i} A_{ij} \mathbb{X}_j \right) \ominus \left(\bigoplus_{j \in \mathcal{N}_i} A_{ij} B_{\rho_{j,1}}(\mathbf{0}_{n_j}) \right) \tag{7.25c}$$

$$\subseteq D_i \mathbb{D}_i \oplus \mathbb{W}_i \ominus \left(\bigoplus_{j \in \mathcal{N}_i} A_{ij} B_{\rho_{j,1}}(\mathbf{0}_{n_j}) \right) \tag{7.25d}$$

$$\subseteq \bar{\mathbb{W}}_i \ominus \left(\bigoplus_{j \in \mathcal{N}_i} A_{ij} B_{\rho_{j,1}}(\mathbf{0}_{n_j}) \right) \tag{7.25e}$$

Manipulations (7.25b) and (7.25c) similar to (6.34b) and (6.34c) and they are justified in the remark after (6.34).
Therefore, there is $\xi_i \in [0, 1)$ (that does not depend on ϵ and δ_z) such that

$$D_i \mathbb{D}_i \oplus \bigoplus_{j \in \mathcal{N}_i} A_{ij} \mathbb{Z}_j \subseteq \xi_i \bar{\mathbb{W}}_i, \tag{7.26}$$

and then, from (7.24),

$$\tilde{w}_{[i]} \in (1 + \delta_z)\xi_i \bar{\mathbb{W}}_i \oplus B_\epsilon(\mathbf{0}_{n_i}), \ \forall t \geq \bar{T}.$$

Note that in (7.20) the parameter $\epsilon > 0$ can be chosen arbitrarily small. Assume that it verifies $\epsilon < (1 + \delta_z)\xi_i \omega_i$, $\forall i \in \mathcal{M}$ where ω_i are the radii of the balls in Assumption 6.3. Then, using Assumption 6.3 we get for $t \geq \bar{T}$

$$\tilde{w}_{[i]}(t) \in (1 + \delta_z)\xi_i(\bar{\mathbb{W}}_i \oplus B_{\omega_i}(\mathbf{0}_{n_i})) \subseteq (1 + \delta_z)\xi_i \bar{\mathbb{Z}}_i^0. \tag{7.27}$$

In view of (7.22) and (7.27), Lemma 6.2 guarantees that

$$x_{[i]}^+ \in (1 + \delta_z)(\mathbb{Z}_i \ominus (1 - \xi_i)\bar{\mathbb{Z}}_i^0) \tag{7.28}$$

From Assumption 6.3, one has $\mathbb{Z}_i \ominus (1 - \xi_i)\bar{\mathbb{Z}}_i^0 \subset \mathbb{Z}_i \ominus B_{(1-\xi_i)\omega_i}(\mathbf{0}_{n_i})$ and hence, since $\mathbb{Z}_i$ contains the origin in its interior, there is $\mu_i \in [0, 1)$ such that $\mathbb{Z}_i \ominus (1 - \xi_i)\mathbb{Z}_i^0 \subset \mu_i \mathbb{Z}_i$. From (7.28) we get $x_{[i]}^+ \in (1 + \delta_z)\mu_i \mathbb{Z}_i$. If in (7.21) we set δ_z such that $(1 + \delta_z)\mu_i < 1$, we have shown that for $t = \bar{T}$ it holds $x_{[i]}(\bar{T} + 1) \in \mathbb{Z}_i$ and the proof is concluded setting $\tilde{T} = \bar{T} + 1$. $\square$

Chapter 8

Plug-and-play distributed state estimation

Contents

8.1 Introduction

As in [FS11a] and Chapter 3, in this chapter we consider discrete-time linear time-invariant subsystems affected by bounded disturbances and propose

a DSE composed by LSEs with a Luenberger-like structure and connected through a network with parent-child topology. We provide conditions for guaranteeing estimation errors fulfill prescribed zonotopic constraints at all times and converge to zero when there are no disturbances. A key feature of our approach is that, differently from [FS11a] and Chapter 3, checking these conditions amounts to numerical tests that are associated with individual LSEs and that can be conducted in parallel using hardware collocated with subsystems. Furthermore, each test requires data from parent subsystems only. These properties enable PnP design of LSEs.
The chapter is structured as follows. The DSE is introduced in Section 8.2. In Section 8.3, the main results allowing design decentralization are presented together with the optimization-based synthesis of LSEs. PnP operations are discussed in 8.4. In Section 8.5 we illustrate the use of the DSE for reconstructing the states of a 2D array of masses connected by springs and dampers. Finally, Section 8.6 is devoted to conclusions.

8.2 Distributed state estimator

We consider a large-scale discrete-time LTI system

$$\begin{aligned} \mathbf{x}^+ &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{D}\mathbf{d} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \boldsymbol{\varrho} \end{aligned} \tag{8.1}$$

composed of M subsystems, in accordance with the notation introduced in Section 1.5. In this chapter we will focus our attention on the problem of bounded-error state estimation, therefore we consider constraints on model and output disturbances on each subsystem. More specifically, we assume that the sets $\mathbb{D}_i \subset \mathbb{R}^{r_i}$ and $\mathbb{O}_i \subset \mathbb{R}^{p_i}$ are zonotopes centered at the origin. Without loss of generality, $\mathbb{D}_i$ can be written as

$$\begin{aligned} \mathbb{D}_i &= \{d_{[i]} \in \mathbb{R}^{r_i} : \mathcal{F}_i w_{[i]} \leq \mathbf{1}_{\bar{\upsilon}_i}\} \\ &= \{d_{[i]} \in \mathbb{R}^{r_i} : d_{[i]} = \Delta_i \digamma_i^d,\ ||\digamma_i^d||_\infty \leq 1\} \end{aligned} \tag{8.2}$$

where $\mathcal{F}_i = (f_{i,1}^T, \dots, f_{i,\bar{\upsilon}_i}^T) \in \mathbb{R}^{\bar{\upsilon}_i \times r_i}$, $\text{rank}(\mathcal{F}_i) = r_i$, $\Delta_i \in \mathbb{R}^{r_i \times \bar{r}_i}$ and $\digamma_i^d \in \mathbb{R}^{\bar{r}_i}$, and $\mathbb{O}_i$ can be written as

$$\begin{aligned} \mathbb{O}_i &= \{\varrho_{[i]} \in \mathbb{R}^{p_i} : \mathcal{G}_i \varrho_{[i]} \leq \mathbf{1}_{\bar{\nu}_i}\} \\ &= \{\varrho_{[i]} \in \mathbb{R}^{p_i} : \varrho_{[i]} = \Upsilon_i \digamma_i^\varrho,\ ||\digamma_i^\varrho||_\infty \leq 1\} \end{aligned} \tag{8.3}$$

where $\mathcal{G}_i = (g_{i,1}^T, \dots, g_{i,\bar{\nu}_i}^T) \in \mathbb{R}^{\bar{\nu}_i \times p_i}$, $\text{rank}(\mathcal{G}_i) = p_i$, $\Upsilon_i \in \mathbb{R}^{p_i \times \bar{p}_i}$ and $\digamma_i^\varrho \in \mathbb{R}^{\bar{p}_i}$.

8.2.1 Distributed state estimator definition

In this section we propose a DSE for (8.1). As in Chapter 3, we define for $i \in \mathcal{M}$ the LSE

$$\tilde{\Sigma}_{[i]}: \quad \tilde{x}^+_{[i]} = A_{ii}\tilde{x}_{[i]} + B_i u_{[i]} - L_{ii}(y_{[i]} - C_i\tilde{x}_{[i]}) + \sum_{j\in\mathcal{N}_i} A_{ij}\tilde{x}_{[j]} - \sum_{j\in\mathcal{N}_i} \tilde{\delta}_{ij}L_{ij}(y_{[j]} - C_j\tilde{x}_{[j]}) \tag{8.4}$$

where $\tilde{x}_{[i]} \in \mathbb{R}^{n_i}$ is the state estimate, $L_{ij} \in \mathbb{R}^{n_i\times p_j}$ are gain matrices and $\tilde{\delta}_{ij} \in \{0,1\}$. This implies that $\tilde{\Sigma}_{[i]}$ depends only on local variables ($\tilde{x}_{[i]}$, $u_{[i]}$ and $y_{[i]}$) and parents' variables ($\tilde{x}_{[j]}$ and $y_{[j]}$, $j \in \mathcal{N}_i$). Binary parameters $\tilde{\delta}_{ij}$, $j \in \mathcal{N}_i$ can be chosen equal to one for exploiting the knowledge of parents' outputs, or equal to zero for reducing the number of transmitted output samples.
Defining the state estimation error as

$$e_{[i]} = x_{[i]} - \tilde{x}_{[i]}, \tag{8.5}$$

from (1.2), (8.4) and (8.5), we obtain the local error dynamics

$$\Theta_{[i]}: \quad e^+_{[i]} = \bar{A}_{ii}e_{[i]} + \sum_{j\in\mathcal{N}_i}\bar{A}_{ij}e_{[j]} + D_i d_{[i]} + L_{ii}\varrho_{[i]} + \sum_{j\in\mathcal{N}_i}\tilde{\delta}_{ij}L_{ij}\varrho_{[j]} \tag{8.6}$$

where $\bar{A}_{ii} = A_{ii} + L_{ii}C_i$ and $\bar{A}_{ij} = A_{ij} + \tilde{\delta}_{ij}L_{ij}C_j$, $i \neq j$. Our main goal is to solve the following problem.

Problem 8.1. Design in a decentralized fashion LSEs $\tilde{\Sigma}_{[i]}$, $i \in \mathcal{M}$ that

(a) are nominally convergent, i.e. when $\mathbb{D}_i = \{\mathbf{0}_{r_i}\}$ and $\mathbb{O}_i = \{\mathbf{0}_{p_i}\}$ it holds

$$||e_{[i]}(t)|| \rightarrow \mathbf{0}_{n_i} \text{ as } t \rightarrow \infty \tag{8.7}$$

(b) guarantee, for suitable initial conditions

$$e_{[i]}(t) \in \mathbb{E}_i, \ \forall t \geq 0 \tag{8.8}$$

where $\mathbb{E}_i \subseteq \mathbb{R}^{n_i}$ are zonotopes centered at the origin given by

$$\begin{aligned}\mathbb{E}_i &= \{e_{[i]} \in \mathbb{R}^{n_i} : \mathcal{H}_i e_{[i]} \leq \mathbf{1}_{\bar{r}_i}\} \\ &= \{e_{[i]} \in \mathbb{R}^{n_i} : e_{[i]} = \Xi_i F^e_i, \ ||F^e_i||_\infty \leq 1\}.\end{aligned} \tag{8.9}$$

In (8.9), $\mathcal{H}_i = (h^T_{i,1}, \dots, h^T_{i,\bar{r}_i}) \in \mathbb{R}^{\bar{r}_i\times n_i}$, $\text{rank}(\mathcal{H}_i) = n_i$, $\Xi_i \in \mathbb{R}^{n_i\times\bar{n}_i}$ and $F^e_i \in \mathbb{R}^{\bar{n}_i}$.

Defining the variable $\mathbf{e} = (e_{[1]}, \dots, e_{[M]}) \in \mathbb{R}^n$, from (8.6) one obtains the collective dynamics of the estimation error

$$\mathbf{e}^+ = \bar{\mathbf{A}}\mathbf{e} + \mathbf{D}\mathbf{d} + \mathbf{L}\boldsymbol{\varrho} \tag{8.10}$$

where the matrices $\bar{\mathbf{A}}$ and $\mathbf{L}$ are composed by blocks $\bar{A}_{ij}$, $i, j \in \mathcal{M}$.
We equip system (8.10) with constraints $\mathbf{e} \in \mathbb{E} = \prod_{i\in\mathcal{M}} \mathbb{E}_i$, $\mathbf{d} \in \mathbb{D}$ and $\boldsymbol{\varrho} \in \mathbb{O}$. From (8.10), if $\mathbf{L}$ is such that $\bar{\mathbf{A}}$ is Schur, then property (8.7) holds. Moreover, if there exists an RPI set $\mathbb{S} \subseteq \mathbb{E}$ for the constrained system (8.10), then $\mathbf{e}(0) \in \mathbb{S}$ guarantees property (8.8). As in Chapter 3, we highlight that methods based on LP for computing $\mathbb{S}$ exist [RKKM05] and [RB10]. However the resulting LP problems require the knowledge of the collective model (8.1) and therefore they become prohibitive for LSSs. In the next section, we propose a method to design LSEs using only computational resources collocated with subsystems. Differently from method propose in Chapter 3 and based on practical invariance, in this chapter we do not need any centralized operation for the design of a LSE.

8.3 Decentralization of LSE design

In the following, we first solve Problem 8.1 in the case of $\mathbb{D} = \{\mathbf{0}_r\}$ and $\mathbb{O} = \{\mathbf{0}_p\}$, i.e. no disturbances act on subsystems (8.1), and then show how to take disturbances into account.
When $\mathbb{D} = \{\mathbf{0}_r\}$ and $\mathbb{O} = \{\mathbf{0}_p\}$, we need to find matrices L_{ij} $i, j \in \mathcal{M}$ such that system (8.10) is asymptotically stable. To achieve this aim in a decentralized fashion, we treat the coupling term $v_{[i]} = \sum_{j\in\mathcal{N}_i} \bar{A}_{ij} e_{[j]}$ as a disturbance for the error dynamics

$$e_{[i]}^+ = \bar{A}_{ii} e_{[i]} + v_{[i]} \tag{8.11}$$

and then confine the error into an RPI set $\mathbb{S}_i \subseteq \mathbb{E}_i$ for (8.11) and $v_{[i]} \in \mathbb{V}_i = \bigoplus_{j\in\mathcal{N}_i} \bar{A}_{ij}\mathbb{E}_j$. The main result, that will also enable PnP design of LSEs, is given in the next proposition.

Proposition 8.1. *Let $\mathbb{D} = \{\mathbf{0}_r\}$ and $\mathbb{O} = \{\mathbf{0}_p\}$. If, for given matrices L_{ij} and parameters $\tilde{\delta}_{ij}$, $i, j \in \mathcal{M}$, the following conditions are fulfilled*

$$\bar{A}_{ii} \text{ is Schur}, \ \forall i \in \mathcal{M} \tag{8.12a}$$

$$\beta_i = \sum_{j\in\mathcal{N}_i} \sum_{k=0}^{\infty} ||\mathcal{H}_i \bar{A}_{ii}^k \bar{A}_{ij} \mathcal{H}_j^\flat||_\infty < 1, \ \forall i \in \mathcal{M} \tag{8.12b}$$

then

(I) $\bar{\mathbf{A}}$ *is Schur;*

(II) $\forall i \in \mathcal{M}$ *there exists an RPI* $\mathbb{S}_i \subseteq \mathbb{E}_i$ *for dynamics* (8.11)*, such that* $\mathbb{S} = \prod_{i \in \mathcal{M}} \mathbb{S}_i$ *is a positively invariant set for system* (8.10)*.*

Proof. The proof is given in the Appendix 8.7.1. □

Some comments are in order. The conditions in Proposition 8.1 guarantee that if $e_{[i]}(0) \in \mathbb{S}_i$, $\forall i \in \mathcal{M}$, then (8.7) and (8.8) hold. Condition (8.12b), that stems from the small gain theorem for networks [DRW07], implies that the coupling between subsystems must be sufficiently small. In particular, if subsystems are decoupled, (8.12b) is always fulfilled and nominal convergence of the state estimator is guaranteed by condition (8.12a) only.

Remark 8.1. We highlight that, for a given $i \in \mathcal{M}$, the quantity β_i in (8.12) depends only upon local fixed parameters $\{A_{ii}, C_i, \mathcal{H}_i\}$, parents' fixed parameters $\{A_{ij}, C_j, \mathcal{H}_j\}_{j \in \mathcal{N}_i}$ and local tunable parameters $\{L_{ii}, \{L_{ij}, \tilde{\delta}_{ij}\}_{j \in \mathcal{N}_i}\}$ but not on parents' tunable parameters. This implies that the choice of $\{L_{ii}, \{L_{ij}, \tilde{\delta}_{ij}\}_{j \in \mathcal{N}_i}\}$ does not influence the choice of $\{L_{jj}, \{L_{jk}, \tilde{\delta}_{jk}\}_{k \in \mathcal{N}_j}\}$, for $i \neq j$.

When system (8.1) is affected by disturbances, i.e. $\mathbb{D} \neq \{\mathbf{0}_r\}$ or $\mathbb{O} \neq \{\mathbf{0}_p\}$, we can still use (8.12) for guaranteeing the stability of matrix $\bar{\mathbf{A}}$, but we need an additional condition in order to guarantee the existence of an RPI set $\mathbb{S}_i \subseteq \mathbb{E}_i$ for the error dynamics

$$e^+_{[i]} = \bar{A}_{ii} e_{[i]} + \tilde{v}_{[i]} \tag{8.13}$$

where the disturbance $\tilde{v}_{[i]}$ verifies

$$\tilde{v}_{[i]} = v_{[i]} + D_i d_{[i]} + L_{ii} \varrho_{[i]} + \sum_{j \in \mathcal{N}_i} \tilde{\delta}_{ij} L_{ij} \varrho_{[j]} \in \tilde{\mathbb{V}}_i \tag{8.14}$$

where

$$\mathbb{V}_i = \bigoplus_{j \in \mathcal{N}_i} \bar{A}_{ij} \mathbb{E}_j \oplus D_i \mathbb{D}_i \oplus L_{ii} \mathbb{O}_i \oplus \bigoplus_{j \in \mathcal{N}_i} \tilde{\delta}_{ij} L_{ij} \mathbb{O}_j.$$

Since $\tilde{\mathbb{V}}_i$ is a zonotope, it can be written as $\tilde{\mathbb{V}}_i = \{\tilde{v}_{[i]} \in \mathbb{R}^{\tilde{n}_i} : \tilde{v}_{[i]} = \Psi_i \tilde{F}_i, \ ||\tilde{F}_i||_\infty \leq 1\}$ where Ψ_i collects its generators.

Proposition 8.2. *For given matrices* L_{ij} *and parameters* $\tilde{\delta}_{ij}$*,* $i, j \in \mathcal{M}$*, if conditions* (8.12) *hold and*

$$\gamma_i = \sum_{k=0}^{\infty} ||\mathcal{H}_i \bar{A}^k_{ii} \Psi_i||_\infty < 1, \ \forall i \in \mathcal{M} \tag{8.15}$$

then, there exists an RPI set $\mathbb{S}_i \subseteq \mathbb{E}_i$ for (8.13)*, such that $\mathbb{S} = \prod_{i\in\mathcal{M}} \mathbb{S}_i$ is an RPI set for system* (8.10)*.*

Proof. The proof is given in the Appendix 8.7.2. □

Remark 8.2. We note that if the subsystems are decoupled, then condition (8.15) implies that there exists an mRPI $\underline{\mathbb{S}}_i \subseteq \mathbb{E}_i$ for the local error dynamics (8.13). Moreover, when subsystems are coupled and $\mathbb{D}_i = \{\mathbf{0}_{r_i}\}$ and $\mathbb{O}_i = \{\mathbf{0}_{p_i}\}$, if $\beta_i < 1$ then $\gamma_i < 1$. Indeed, $\mathbb{D}_i = \{\mathbf{0}_{r_i}\}$ and $\mathbb{O} = \{\mathbf{0}_{p_i}\}$ imply that $\Delta_i = \mathbf{0}_{r_i\times\bar{r}_i}$ and $\Upsilon_i = \mathbf{0}_{\underline{p}_i\times\bar{p}_i}$, as shown in the proof of Proposition 8.1, it holds $\sum_{k=0}^{\infty}\sum_{j\in\mathcal{N}_i}||\mathcal{H}_i\bar{A}_{ii}^k\bar{A}_{ij}\Xi_j^\flat||_\infty \leq \sum_{k=0}^{\infty}\sum_{j\in\mathcal{N}_i}||\mathcal{H}_i\bar{A}_{ii}^k\bar{A}_{ij}\mathcal{H}_j^\flat||_\infty$. Finally, the pieces of information needed for computing scalars γ_i are the same needed for computing scalars β_i (see Remark 8.1).

From results in Proposition 8.1 and 8.2, Problem 8.1 can be decomposed into the following independent design problems for $i \in \mathcal{M}$.

Problem 8.2 (Problem $\mathcal{P}_i$)**.** Check if there exist L_{ii} and $\{L_{ij}\}_{j\in\mathcal{N}_i}$ such that $\bar{A}_{ii}$ is Schur, $\beta_i < 1$ and $\gamma_i < 1$.

Remark 8.3. As shown in [KG98], a necessary condition for the existence of RPI sets $\mathbb{S}_i$ for (8.13) is that

$$\tilde{\mathbb{V}}_i \subseteq \mathbb{E}_i,\ \forall i \in \mathcal{M} \tag{8.16}$$

where $\tilde{\mathbb{V}}_i$ depend upon sets $\mathbb{E}_j$, $j \in \mathcal{N}_i$, see (8.14). In our approach, sets $\mathbb{E}_i$ are assigned *a priori* on the basis, e.g. of application-dependent constraints. Therefore we implicitly assume conditions (8.16) are verified. However, if subsystems are added sequentially to an existing plant and LSEs are designed with the PnP procedure described in Section 8.4, conditions (8.16) are automatically checked and, if violated, they prevent from plug in subsystem $\Sigma_{[i]}$. We also highlight that when sets $\mathbb{E}_i$ can be arbitrarily chosen, centralized methods for fulfilling conditions (8.16) exist [FS11a].

8.3.1 Optimization-based synthesis of LSEs

The procedure for solving problems $\mathcal{P}_i$, $i \in \mathcal{M}$ is summarized in Algorithm 8.1 that can be executed in parallel by each subsystem using local hardware.

In Step (I), if $\tilde{\delta}_{ij} = 1$, the computation of matrices L_{ij}, $j \in \mathcal{N}_i$ is required. Since the choice of L_{ij} affects the coupling term $\bar{A}_{ij} = A_{ij} + \tilde{\delta}_{ij}L_{ij}C_j$, and hence the possibility of verifying inequalities (8.12) and (8.15), as in Section 7.3.1 in the case of distributed controllers, we propose to reduce

Algorithm 8.1 Design of the LSE $\tilde{\Sigma}_{[i]}$ for subsystem $\Sigma_{[i]}$

Input: zonotopes $\mathbb{E}_i$, $\mathbb{D}_i$, $\mathbb{O}_i$ and scalars $\tilde{\delta}_{ij}, \forall j \in \mathcal{N}_i$.
Output: set $\mathbb{S}_i$ and state estimator $\tilde{\Sigma}_{[i]}$.

(I) if $\delta_{ij} = 1$, compute the matrix L_{ij}, $\forall j \in \mathcal{N}_i$ solving

$$\min_{L_{ij}} ||\mathcal{H}_i \bar{A}_{ij} \mathcal{H}_j^\flat||_p \tag{8.17}$$

where p is a generic norm.

(II) compute a matrix L_{ii} such that $\beta_i < 1$ and $\gamma_i < 1$. If it does not exist **stop**;

(III) compute the set $\mathbb{S}_i$.

the magnitude of coupling by minimizing the magnitude of $\bar{A}_{ij}$ in (8.17), where $\mathcal{H}_i$ and $\mathcal{H}_j^\flat$ allow us to take into account the size of sets $\mathbb{E}_i$ and $\mathbb{E}_j$, respectively. More precisely, it can be shown that the term $||\mathcal{H}_i \bar{A}_{ij} \mathcal{H}_j^\flat||_p$ is a measure of how much the coupling term $\bar{A}_{ij} e_{[j]}$, $j \in \mathcal{N}_i$ affects the fulfillment of the constraint $e_{[i]} \in \mathbb{E}_i$ (see Appendix 8.7.3). We highlight that the minimization of $\|\mathcal{H}_i \bar{A}_{ij} \mathcal{H}_j^\flat\|_1$ in (8.17) amounts to an LP problem and the minimization of $\|\mathcal{H}_i \bar{A}_{ij} \mathcal{H}_j^\flat\|_F$ can be recast into a QP problem. So far, the parameters δ_{ij} have been considered fixed. However, if in Step (I) one obtains $L_{ij} = \mathbf{0}_{n_i \times p_j}$ for some $j \in \mathcal{N}_i$, it is impossible to reduce the magnitude of the coupling term $\bar{A}_{ij}$ and the knowledge of $y_{[j]}$ is useless for estimator $\tilde{\Sigma}_{[i]}$. This suggests to revise the choice of $\tilde{\delta}_{ij}$ and set $\tilde{\delta}_{ij} = 0$.

In Step (II), for the computation of matrix L_{ii} we propose an automatic method in order to guarantee satisfaction of inequalities (8.12) and (8.15). This procedure parallels the method proposed in Section 5.3 for control design. Moreover the proposed procedure is implemented in the *PnPMPC-toolbox* for MatLab [RBFT12] (see Appendix C). We solve the following nonlinear optimization problem

$$\min_{L_{ii}} \mu_i \tag{8.18a}$$
$$\bar{\rho}(A_{ii} + L_{ii}C_i) < 1 \tag{8.18b}$$
$$\beta_i < 1 \tag{8.18c}$$
$$\gamma_i < 1 \tag{8.18d}$$

where $\mu_i = \max(\beta_i, \gamma_i, \bar{\rho}(A_{ii}+L_{ii}C_i))$ and constraint (8.18d) is needed only if $\mathbb{D}_i \neq \{\mathbf{0}_{r_i}\}$ or $\mathbb{O}_i \neq \{\mathbf{0}_{p_i}\}$. Since (8.18) is a nonlinear optimization problem, a suitable initialization of L_{ii} is needed, hence we initialize L_{ii} as the dual LQR gain associated to matrices $\tilde{Q}_i \geq \mathbf{0}_{n_i \times n_i}$ and $\tilde{R}_i > \mathbf{0}_{m_i \times m_i}$, that are inputs for the nonlinear optimization. The feasibility of problem (8.18) guarantees that the estimator $\tilde{\Sigma}_{[i]}$ can be successfully designed. Note that if all matrices L_{ij}, $j \in \mathcal{N}_i$ are such that $\bar{A}_{ij} = \mathbf{0}_{n_i \times n_j}$, the inequality (8.18c) is always fulfilled and, when $\mathbb{D} = \{\mathbf{0}_{r_i}\}$ and $\mathbb{O} = \{\mathbf{0}_{p_i}\}$, the optimization problem (8.18) is reduced to the problem of design a standard dual LQR gain.
In Step (III) of Algorithm 8.1 we need to compute a nonempty RPI set $\mathbb{S}_i \subseteq \mathbb{E}_i$ that, in view of Propositions 8.1 and 8.2, exists if the optimization problem (8.18) is feasible. To this purpose, several algorithms can be used. For instance, [RKKM05] discusses the computation of ϵ-outer approximation of the mRPI $\underline{\mathbb{S}}_i$. The MRPI set $\bar{\mathbb{S}}_i$ can be obtained using methods in [GT91]. More recently, efficient procedures have been also proposed for computing polytopic [RB10] or zonotopic [Rak05] RPI sets.

8.4 Plug-and-play operations

Consider a plant composed by subsystems $\Sigma_{[i]}$, $i \in \mathcal{M}$ equipped with local state estimators $\tilde{\Sigma}_{[i]}$, $i \in \mathcal{M}$ produced by Algorithm 8.1. In case subsystems are added or removed, we show how to preserve properties (8.7) and (8.8) by updating a limited number of existing LSEs. Note that plug in and plug out of subsystems are here considered as offline operations, i.e. they do not lead to switching between different dynamics in real time.

8.4.1 Plugging in operation

We start considering plug in of subsystem $\Sigma_{[M+1]}$, characterized by parameters $A_{M+1,M+1}$, C_{M+1}, $\mathbb{E}_{M+1}$, $\mathbb{D}_{M+1}$, $\mathbb{O}_{M+1}$, $\mathcal{N}_{M+1}$ and coupling terms $\{A_{M+1,j}\}_{j\in\mathcal{N}_{M+1}}$. In particular, $\mathcal{N}_{M+1}$ identifies the subsystems that will influence $\Sigma_{[M+1]}$ through matrices $\{A_{M+1,j}\}_{j\in\mathcal{N}_{M+1}}$. Subsystems that will be influenced by $\Sigma_{[M+1]}$ are given by $\mathcal{S}_{M+1}$ where

$$\mathcal{S}_i = \{j : i \in \mathcal{N}_j\}$$

is the set of children of subsystem $\Sigma_{[i]}$. For designing the LSE $\tilde{\Sigma}_{[M+1]}$ we execute Algorithm 8.1 that needs information only from subsystems $\Sigma_{[j]}$, $j \in \mathcal{N}_{M+1}$. If Algorithm 8.1 stops before the last step, we declare that $\Sigma_{[M+1]}$ cannot be plugged in. Since sets $\mathcal{N}_j$, $j \in \mathcal{S}_{M+1}$ have now one more element, previously obtained matrices L_{jj}, $j \in \mathcal{S}_{M+1}$ might give $\beta_i \geq 1$ or $\gamma_i \geq 1$. Indeed, quantities β_i and γ_i in (8.12) and (8.15) can only increase. Furthermore, the size of the set $\mathbb{S}_j$ increases and therefore the condition $\mathbb{S}_j \subseteq \mathbb{E}_j$ could be violated. This means that for each $j \in \mathcal{S}_{M+1}$ the LSE $\tilde{\Sigma}_{[j]}$ must be redesigned by running Algorithm 8.1. Again, if Algorithm 8.1 stops before completion for some $j \in \mathcal{S}_{M+1}$, we declare that $\Sigma_{[M+1]}$ cannot be plugged in.
Note that LSE redesign does not propagate further in the network, i.e. even without changing state estimators $\tilde{\Sigma}_{[i]}$, $i \notin \{M+1\} \bigcup \mathcal{S}_{M+1}$, properties (8.7) and (8.8) are guaranteed for the new DSE.

8.4.2 Unplugging operation

We consider plug out of subsystem $\Sigma_{[k]}$, $k \in \mathcal{M}$. Since for each $i \in \mathcal{S}_k$ the set $\mathcal{N}_i$ contains one element less, one has that β_i in (8.12) and γ_i in (8.15) cannot increase. Furthermore, the set $\mathbb{S}_i$, chosen before the removal of system $\Sigma_{[k]}$, still verifies $\mathbb{S}_i \supseteq \tilde{\mathbb{V}}_i$ and therefore previously obtained optimizers for problem (8.17) can still be used. This means that for each $i \in \mathcal{S}_k$ the LSE $\tilde{\Sigma}_{[i]}$ does not have to be redesigned. Moreover, since for each system $\Sigma_{[j]}$, $j \notin \{k\} \bigcup \mathcal{S}_k$, the set $\mathcal{N}_j$ does not change, the redesign of the LSE $\tilde{\Sigma}_{[j]}$ is not required.
In conclusion, the removal of system $\Sigma_{[k]}$ does not require the redesign of any LSE in order to guarantee (8.7) and (8.8). However systems $\Sigma_{[i]}$ $i \in \mathcal{S}_k$ have one parent less and the redesign of LSEs $\tilde{\Sigma}_{[i]}$ through Algorithm 8.1 could improve the performance.

8.5 Example

We consider a system composed by 16 masses coupled as in Figure 8.1 where the four edges connected to a point correspond to springs and dampers arranged as in Figure 6.9 in Section 6.6.3. Each mass $f \in 1 : 16$ is an

LTI system with state variables $x_{[f]} = (x_{[f,1]}, x_{[f,2]}, x_{[f,3]}, x_{[f,4]})$ and input $u_{[f]} = (u_{[f,1]}, u_{[f,2]})$, where $x_{[f,1]}$ and $x_{[f,3]}$ are the displacements of mass f with respect to a given equilibrium position in the plane (equilibria lie on a regular grid), $x_{[f,2]}$ and $x_{[f,4]}$ are the horizontal and vertical velocity of the mass f, respectively, and $100u_{[f,1]}$ (respectively $100u_{[f,2]}$) is the force applied to mass f in the horizontal (respectively, vertical) direction. The values of m_f have been extracted randomly in the interval $[5, 10]$ while spring constants and damping coefficients are identical and equal to 0.5. Each mass is equipped with local state estimation error constraints $||e_{[f,j]}||_\infty \leq 1$, $j = 1, 3$ and $||e_{[f,l]}||_\infty \leq 1.5$, $l = 2, 4$.
A subsystem $\Sigma_{[i]}$, $i \in \mathcal{M} = 1 : 4$ is a group of four masses as in Figure 8.1. Therefore each subsystem has order 16 and two parents. For each subsystem $\Sigma_{[i]}$ we have 8 outputs that are the displacements of two masses and the velocities of the other two masses.

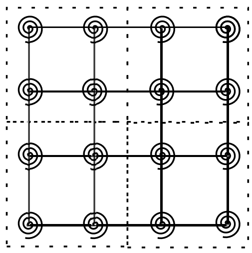
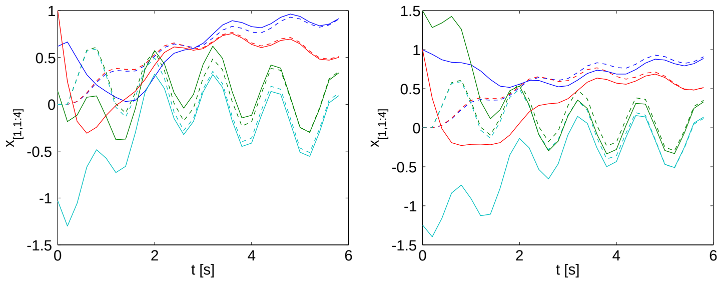
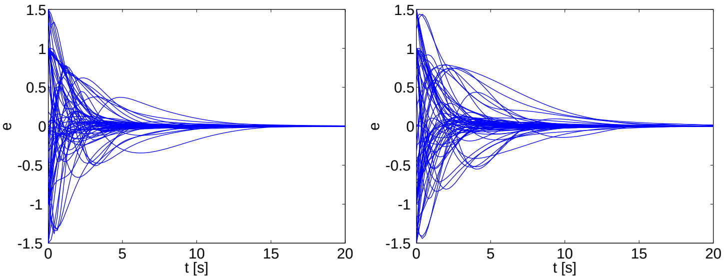


Figure 8.1: Position of the 16 masses on the plane. Dashed lines define subsystems $\Sigma_{[i]}$, $i \in \mathcal{M} = 1 : 4$.

We obtain models $\Sigma_{[i]}$ by discretizing continuous-time models with 0.2 sec sampling time, using zero-order hold discretization for the local dynamics and treating $x_{[j]}$, $j \in \mathcal{N}_i$ as exogenous signals [FCS13]. We design an LSE $\tilde{\Sigma}_{[i]}$, $i \in \mathcal{M}$ using Algorithm 8.1 and assuming matrices $\tilde{Q}_i = 0.01\mathbb{I}_{16}$ and $R_i = 100\mathbb{I}_8$ as inputs for nonlinear optimization problem (8.18). The modeling of the LSS, the design of LSEs and the simulations have been performed using [RBFT12]. In Figure 8.2 we show a simulation where the initial state of each mass is $x_{[f]}(0) = 0$, $f \in 1 : 16$ and the control inputs $u_{[f,l]}(k) = 0.1 \sin(k)$, $l \in 1 : 2$, have been used. We initialize each LSE in order to have $e_{[i]} \in \mathbb{S}_i$. Estimation results produced by LSEs that have been designed with $\tilde{\delta}_{ij} = 0$, $j \in \mathcal{N}_i$ are represented in the left panels of Figures 8.2a and 8.2b. Results obtained by setting $\tilde{\delta}_{ij} = 1$, $j \in \mathcal{N}_i$ and choosing Frobenius norm in (8.17), i.e. $p =$ "F", are shown in the right panels of Figures 8.2a and 8.2b. One can notice that in both cases, state estimation errors converge to zero and they are bounded at all times.



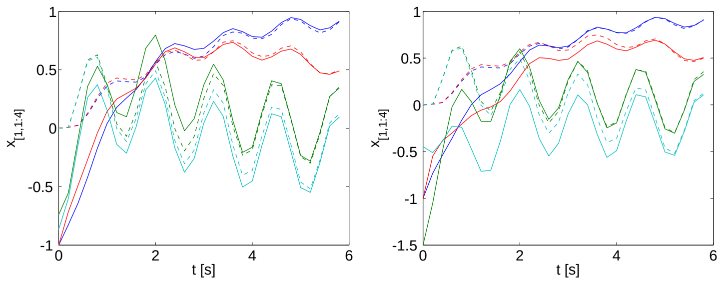
(a) State (dashed lines) and state estimation (continuous line) of the upper left mass in Figure 8.1 at time instants $t = 0 : 6$ sec.



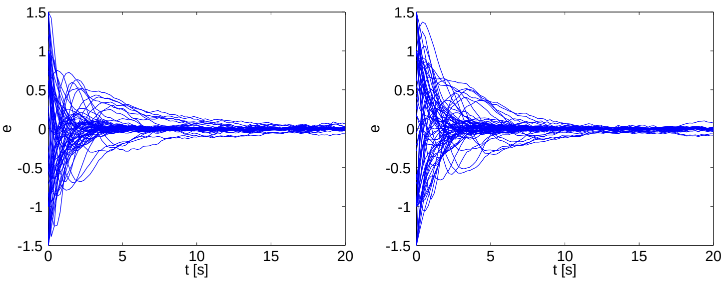
(b) Estimation errors for all states at times $t = 0 : 20$ sec.

Figure 8.2: State estimation results for LSEs designed setting $\tilde{\delta}_{ij} = 0$, $j \in \mathcal{N}_i$ (left panels) and $\tilde{\delta}_{ij} = 1$, $j \in \mathcal{N}_i$ (right panels). In panels 8.2a the same color has been used for a state and its estimate: cyan and green lines denote velocities while blue and red lines denote positions.

In Figure 8.3 we show a simulation where each state of subsystem $\Sigma_{[i]}$, $i \in 1 : 4$ is affected by a disturbance $d_{[i]}$ sampled from the uniform distribution in the set $\mathbb{D}_i = \{d_{[i]} \in \mathbb{R} : |d_{[i]}| \leq 0.015\}$. This has been obtained setting $D_i = \mathbf{1}_{16}$ and $\mathbb{O}_i = \{\mathbf{0}_{p_i}\}$. The left panels of Figures 8.3a and 8.3b show results produced by LSEs designed with $\tilde{\delta}_{ij} = 0$, $j \in \mathcal{N}_i$ while the right panels of Figures 8.3a and 8.3b show the results obtained for $\tilde{\delta}_{ij} = 1$, $j \in \mathcal{N}_i$ and choosing norm $p = F$. In both cases, errors fulfill the prescribed bounds but do not converge to zero because of the persistent disturbances $d_{[i]}$, $i \in 1 : 4$.



(a) State (dashed lines) and state estimation (continuous line) of the upper left mass in Figure 8.1 at time instants $t = 0 : 6$ sec.



(b) Estimation errors for all states at times $t = 0 : 20$ sec.

Figure 8.3: State estimation results for LSEs designed setting $\tilde{\delta}_{ij} = 0$, $j \in \mathcal{N}_i$ (left panels) and $\tilde{\delta}_{ij} = 1$, $j \in \mathcal{N}_i$ (right panels). In panels 8.3a the same color has been used for a state and its estimate: cyan and green lines denote velocities while blue and red lines denote positions.

8.6 Final comments

In this chapter, we have proposed a novel DSE for large-scale linear perturbed systems, which guarantees that the estimation errors are bounded into prescribed sets and converge to zero in absence of disturbances. The algorithm is based on the partition of the overall system into subsystems with non-overlapping states. In particular, the design of LSEs can be carried out in a decentralized fashion by solving a suitable optimization problem where just information by parent nodes is required. This allows one to efficiently update the overall DSE when subsystems are plugged in and out.

In Chapter 9, we will propose the design of output-feedback PnP controllers combining the proposed state estimator and the state-feedback PnP controllers presented in previous chapters.

8.7 Appendix

8.7.1 Proof of Proposition 8.1

The proof uses arguments that are similar to the ones adopted in Section 5.7.1.

Proof of (I)

Proof. Define a matrix $\mathbb{M}$ such that its ij-th entry μ_{ij} is

$$\begin{array}{ll} \mu_{ij} = -1 & \text{if} \quad i = j \\ \mu_{ij} = \sum_{k=0}^{\infty} ||\mathcal{H}_i \bar{A}_{ii}^k \bar{A}_{ij} \mathcal{H}_j^{\flat}||_{\infty} & \text{if} \quad i \neq j. \end{array}$$

Note that all the off-diagonal entries of matrix $\mathbb{M}$ are non-negative, i.e., $\mathbb{M}$ is Metzler (see Section A.2 in Appendix A).
Inequalities (8.12b) are equivalent to $\mathbb{M}\nu < \mathbf{0}_M$ where $\nu = \mathbf{1}_M$. Then, from Lemma A.1, $\mathbb{M}$ is Hurwitz. From Lemma A.2, (8.12b) implies that matrix $\Gamma = \mathbb{M} + \mathbb{I}_M$ is Schur.
For dynamics (8.11), we have

$$e_{[i]}(t) = \bar{A}_{ii}^t e_{[i]}(0) + \sum_{k=0}^{t-1} \bar{A}_{ii}^k \sum_{j \in \mathcal{N}_i} \bar{A}_{ij} e_{[j]}(t-k-1). \tag{8.19}$$

In view of (8.19) we can write

$$||\mathcal{H}_i e_{[i]}(t)||_{\infty} \leq ||\mathcal{H}_i \bar{A}_{ii}^t \mathcal{H}_i^{\flat}||_{\infty} ||\mathcal{H}_i e_{[i]}(0)||_{\infty} + \sum_{j \in \mathcal{N}_i} \gamma_{ij} \max_{k \leq t} ||\mathcal{H}_j e_{[j]}(k)||_{\infty}.$$

where γ_{ij} are the entries of Γ. Denoting $\tilde{e}_{[i]} = \mathcal{H}_i e_{[i]}$, we can collectively define $\tilde{\mathbf{e}} = \tilde{\mathcal{H}}\mathbf{e}$, where $\tilde{\mathcal{H}} = \text{diag}(\mathcal{H}_1, \ldots, \mathcal{H}_M)$. From the definition of sets $\mathbb{E}_i$, we have $\text{rank}(\tilde{\mathcal{H}}) = n$. We define the system

$$\tilde{\mathbf{e}}^+ = \tilde{\bar{A}}\tilde{\mathbf{e}} \tag{8.20}$$

where $\tilde{\bar{A}} = \tilde{\mathcal{H}}\bar{\mathbf{A}}\tilde{\mathcal{H}}^{\flat}$. In order to analyze the stability of the origin of (8.20), we use the small gain theorem for networks in [DRW07]. In view of Corollary 16 in [DRW07], the overall system (8.20) is asymptotically stable if the gain matrix Γ is Schur and, as shown above, this property is implied by (8.12). Moreover, system (8.20) is an expansion of the original system (see Chapter 3.4 in [Lun92]). In view of the inclusion principle [Sta04], the asymptotic stability of (8.20) implies the asymptotic stability of the original system. ☐

Proof of (II)

Proof. First note that, for $i \in \mathcal{M}$, since $\mathbb{E}_i$ is a zonotope, $||h_{i,\tau}^T \Xi_i||_\infty = 1$ for all $\tau \in 1 : \bar{\tau}_i$ and therefore $||\mathcal{H}_i \Xi_i||_\infty = 1$. This implies that

$$
\begin{aligned}
||h_{i,\tau}^T \bar{A}_{ii}^k \bar{A}_{ij} \Xi_j||_\infty &\leq ||h_{i,\tau}^T \bar{A}_{ii}^k \bar{A}_{ij} \mathcal{H}_j^\flat||_\infty ||\mathcal{H}_j \Xi_j||_\infty \\
&= ||h_{i,\tau}^T \bar{A}_{ii}^k \bar{A}_{ij} \mathcal{H}_j^\flat||_\infty \\
&\leq ||\mathcal{H}_i \bar{A}_{ii}^k \bar{A}_{ij} \mathcal{H}_j^\flat||_\infty .
\end{aligned}
$$

Therefore, from (8.12b), for all $\tau \in 1 : \bar{\tau}_i$ it holds

$$
\sum_{k=0}^{\infty} \sum_{j \in \mathcal{N}_i} ||h_{i,\tau}^T \bar{A}_{ii}^k \bar{A}_{ij} \Xi_j||_\infty \leq \sum_{k=0}^{\infty} \sum_{j \in \mathcal{N}_i} ||\mathcal{H}_i \bar{A}_{ii}^k \bar{A}_{ij} \mathcal{H}_j^\flat||_\infty < 1. \tag{8.21}
$$

The next aim is to prove that there exists an RPI $\mathbb{S}_i \subseteq \mathbb{E}_i$ for the dynamics (8.11), in particular we define $\mathbb{S}_i$ as an outer approximation of the mRPI $\underline{\mathbb{S}}_i$ and we prove that the outer approximation always exists.
The mRPI for (8.11) is given by [RKKM05]

$$
\underline{\mathbb{S}}_i = \bigoplus_{k=0}^{\infty} \bar{A}_{ii}^k \bigoplus_{j \in \mathcal{N}_i} \bar{A}_{ij} \mathbb{E}_j . \tag{8.22}
$$

From [RKKM05], for given $\epsilon_i > 0$ there exist $\alpha_i \in [0, 1)$ and $s_i \in \mathbb{N}_+$ such that the set

$$
\mathbb{S}_i(\epsilon_i) = (1 - \alpha_i)^{-1} \bigoplus_{k=0}^{s_i - 1} \bar{A}_{ii}^k \bigoplus_{j \in \mathcal{N}_i} \bar{A}_{ij} \mathbb{E}_j
$$

is an ϵ_i-outer approximation of the mRPI $\underline{\mathbb{S}}_i$.
Using arguments from Section 3 of [KG98], we can then guarantee that $\mathbb{S}_i(\epsilon_i) \subseteq \mathbb{E}_i$. In fact for all $\tau \in 1 : \bar{\tau}_i$

$$
\sup_{s_{[i]} \in \mathbb{S}_i(\epsilon_i)} h_{i,\tau}^T s_{[i]} \leq 1. \tag{8.23}
$$

Using (8.22), the inequalities (8.23) are verified if

$$
\sup_{\substack{\{e_{[j]}(k) \in \mathbb{E}_j\}_{j \in \mathcal{N}_i}^{k=0,\dots,\infty} \\ \sigma_i \in B_{\epsilon_i}(\mathbf{0}_{n_i})}} z_{i,\tau}^x(\{e_{[j]}(k)\}_{j \in \mathcal{N}_i}^{k=0,\dots,\infty}) + ||h_{i,\tau}^T \sigma_i||_\infty \leq 1 \tag{8.24}
$$

where $z^x_{i,\tau}(\cdot) = h^T_{i,\tau} \sum_{k=0}^{\infty} \bar{A}^k_{ii} \sum_{j\in\mathcal{N}_i} \bar{A}_{ij} e_{[j]}(k)$.
Since $||h^T_{i,\tau}\sigma_i||_\infty \leq ||h^T_{i,\tau}||_\infty \epsilon_i$, conditions (8.24) are satisfied if

$$\sup_{\{e_{[j]}(k)\in\mathbb{S}_j\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty}} z^x_{i,\tau}(\{e_{[j]}(k)\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty}) \leq 1 - ||h^T_{i,\tau}||_\infty \epsilon_i. \tag{8.25}$$

Using (8.9), we can rewrite (8.25) as

$$\sup_{\{||\digamma^e_j(k)||_\infty \leq 1\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty}} z^d_{i,\tau}(\{\digamma^e_j(k)\}_{j\in\mathcal{N}_i}^{k=0,\dots,\infty}) \leq 1 - ||h^T_{i,\tau}||_\infty \epsilon_i \tag{8.26}$$

where $z^d_{i,\tau}(\cdot) = h^T_{i,\tau}(\sum_{k=0}^{\infty} \bar{A}^k_{ii} \sum_{j\in\mathcal{N}_i} \bar{A}_{ij}\Xi_j \digamma^e_j(k))$.
The inequalities (8.26) are satisfied if

$$\sum_{k=0}^{\infty} \sum_{j\in\mathcal{N}_i} ||h^T_{i,\tau}\bar{A}^k_{ii}\bar{A}_{ij}\Xi_j||_\infty \leq 1 - ||h^T_{i,\tau}||_\infty \epsilon_i \tag{8.27}$$

for all $\tau \in 1 : \bar{\tau}_i$.
In view of (8.21), there exists a sufficiently small $\epsilon_i > 0$ satisfying (8.27). Hence we proved that $\forall i \in \mathcal{M}$ there exists an RPI $\mathbb{S}_i \subseteq \mathbb{E}_i$ for dynamics (8.11). Moreover if we define $\mathbb{S} = \prod_{i\in\mathcal{M}} \mathbb{S}_i$, the set $\mathbb{S}$ is an invariant set for system (8.10) equipped with constraints $\mathbb{E}$, $\mathbb{D} = \{\mathbf{0}_r\}$ and $\mathbb{O} = \{\mathbf{0}_p\}$. □

8.7.2 Proof of Proposition 8.2

Proof. In the following we use similar arguments of Proof of Proposition 8.1 (see Section 8.7.1) to prove that there exists an RPI $\mathbb{S}_i \subseteq \mathbb{E}_i$ for the dynamics (8.13), in particular we define $\mathbb{S}_i$ as an outer approximation of the mRPI $\underline{\mathbb{S}}_i$ and we prove that the outer approximation always exists.
The mRPI for (8.13) is given by [RKKM05]

$$\begin{aligned}\underline{\mathbb{S}}_i &= \bigoplus_{k=0}^{\infty} \bar{A}^k_{ii} \left(\bigoplus_{j\in\mathcal{N}_i} \bar{A}_{ij}\mathbb{E}_j \oplus D_i\mathbb{D}_i \oplus L_{ii}\mathbb{O}_i \oplus \bigoplus_{j\in\mathcal{N}_i} \tilde{\delta}_{ij} L_{ij}\mathbb{O}_j \right) \\ &= \bigoplus_{k=0}^{\infty} \bar{A}^k_{ii}\tilde{\mathbb{V}}_i.\end{aligned} \tag{8.28}$$

From [RKKM05], for given $\epsilon_i > 0$ there exist $\alpha_i \in [0, 1)$ and $s_i \in \mathbb{N}_+$ such that the set

$$\mathbb{S}_i(\epsilon_i) = (1 - \alpha_i)^{-1} \bigoplus_{k=0}^{s_i-1} \bar{A}^k_{ii}\tilde{\mathbb{V}}_i$$

is an ϵ_i-outer approximation of the mRPI $\underline{\mathbb{S}}_i$.
Using arguments from Section 3 of [KG98], we can then guarantee that $\mathbb{S}_i(\epsilon_i) \subseteq \mathbb{E}_i$. In fact for all $\tau \in 1 : \bar{\tau}_i$

$$\sup_{s_{[i]} \in \mathbb{S}_i(\epsilon_i)} h_{i,\tau}^T s_{[i]} \leq 1. \tag{8.29}$$

Using (8.28), the inequalities (8.29) are verified if

$$\sup_{\substack{\sigma_i \in B_{\epsilon_i}(\mathbf{0}_{n_i}) \\ \tilde{v}_{[i]} \in \tilde{\mathbb{V}}_i}} z_{i,\tau}^x(\{\tilde{v}_{[i]}(k)\}^{k=0,\dots,\infty}) + ||h_{i,\tau}^T \sigma_i||_\infty \leq 1 \tag{8.30}$$

where $z_{i,\tau}^x(\cdot) = h_{i,\tau}^T \sum_{k=0}^\infty \bar{A}_{ii}^k \tilde{v}_{[i]}$.
Since $||h_{i,\tau}^T \sigma_i||_\infty \leq ||h_{i,\tau}^T||_\infty \epsilon_i$, conditions (8.30) are satisfied if

$$\sup_{\tilde{v}_{[i]} \in \tilde{\mathbb{V}}_i} z_{i,\tau}^x(\{\tilde{v}_{[i]}(k)\}^{k=0,\dots,\infty}) \leq 1 - ||h_{i,\tau}^T||_\infty \epsilon_i. \tag{8.31}$$

Using (8.2), (8.3) and (8.9) we can rewrite (8.31) as

$$\sup_{\{||\tilde{F}_i(k)||_\infty \leq 1\}^{k=0,\dots,\infty}} z_{i,\tau}^d(\{\tilde{F}_i(k)\}^{k=0,\dots,\infty}) \leq 1 - ||h_{i,\tau}^T||_\infty \epsilon_i \tag{8.32}$$

where $z_{i,\tau}^d(\cdot) = h_{i,\tau}^T (\sum_{k=0}^\infty \bar{A}_{ii}^k \Psi_i \tilde{F}_i(k))$.
The inequalities (8.32) are satisfied if

$$\sum_{k=0}^\infty ||h_{i,\tau}^T \bar{A}_{ii}^k \Psi_i||_\infty \leq 1 - ||h_{i,\tau}^T||_\infty \epsilon_i$$

for all $\tau \in 1 : \bar{\tau}_i$.
We proved that $\forall i \in \mathcal{M}$ there exists an RPI $\mathbb{S}_i \subseteq \mathbb{E}_i$ for dynamics (8.13). Moreover if we define $\mathbb{S} = \prod_{i \in \mathcal{M}} \mathbb{S}_i$, the set $\mathbb{S}$ is an RPI set for system (8.10) equipped with constraints $\mathbb{E}$ and disturbances $\mathbb{D} \neq \{\mathbf{0}_r\}$ or $\mathbb{O} \neq \{\mathbf{0}_p\}$. □

8.7.3 Notes on the optimization problem (8.17)

In order to fulfill condition (8.12b), we need to guarantee at least that

$$\bar{A}_{ij} \mathbb{E}_j \subseteq \mathbb{E}_i$$

hence

$$\mathcal{H}_i \bar{A}_{ij} e_{[j]} \leq 1, \forall e_{[j]} \in \mathbb{E}_j$$

In order to minimize the effect of coupling terms $\bar{A}_{ij}$, from (8.9) we can solve the following optimization problem.

$$\eta_{ij} = \min_{L_{ij}} \max_{\substack{e_{[j]}=\Xi_j F_j^e \\ ||F_j^e||_\infty \leq 1}} ||\mathcal{H}_i \bar{A}_{ij} e_{[j]}||_p. \tag{8.33}$$

where p is a generic norm. Using arguments similar to the ones adopted in the proof of Proposition 8.1, from (8.33) we obtain

$$\begin{aligned}
\eta_{ij} &\leq \min_{L_{ij}} \max_{||F_j^e||_\infty \leq 1} ||\mathcal{H}_i \bar{A}_{ij} \Xi_j F_j^e||_p \\
&\leq \min_{L_{ij}} \max_{||F_j^e||_\infty \leq 1} ||\mathcal{H}_i \bar{A}_{ij} \mathcal{H}_j^\flat||_p ||\mathcal{H}_j \Xi_j F_j^e||_p \\
&\leq \min_{L_{ij}} \max_{||F_j^e||_\infty \leq 1} ||\mathcal{H}_i \bar{A}_{ij} \mathcal{H}_j^\flat||_p ||\mathcal{H}_j \Xi_j||_p ||F_j^e||_p.
\end{aligned}$$

Irrespectively of p, there exist constants $c_{1,p} > 0$ and $c_{2,p} > 0$ such that

$$||\mathcal{H}_j \Xi_j||_p \leq c_{1,p} ||\mathcal{H}_j \Xi_j||_\infty = c_{1,p}$$

$$\max_{||F_j^e||_\infty \leq 1} ||F_j^e||_p \leq \max_{||F_j^e||_\infty \leq 1} c_{2,p} ||F_j^e||_\infty = c_{2,p}.$$

Therefore, we can conclude that

$$\eta_{ij} \leq c_{1,p} c_{2,p} \min_{L_{ij}} ||\mathcal{H}_i \bar{A}_{ij} \mathcal{H}_j^\flat||_p$$

and this motivates the optimization problem (8.17).

Chapter 9

Robust output-feedback plug-and-play MPC

Contents

9.1 Introduction

In this chapter we propose an algorithm for PnP design of output-feedback controllers. In particular, we propose methods combining PnP local state estimators described in Chapter 8 and PnP-DiMPC, described in Chapter 6 and 7. The main challenge is that, because of the presence of constraints, we need

- to guarantee bounded-error state estimation
- to design local controllers guaranteeing constraint satisfaction in presence of estimation errors.

Therefore in Section 9.2 we introduce an output-feedback DiMPC regulator based on tube MPC and in Section 9.3 we show how the control scheme can be designed in a PnP fashion. In Section 9.4 we propose a numerical example and Section 9.5 is devoted to some conclusions.

9.2 Output-feedback distributed tube-based MPC

We consider a large-scale discrete-time LTI system

$$\begin{aligned} \mathbf{x}^+ &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{D}\mathbf{d} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \boldsymbol{\varrho} \end{aligned} \tag{9.1}$$

composed of M subsystems, in accordance with the notation introduced in Section 1.5. In this chapter we consider that each subsystem is equipped with state, input and disturbance constraints. In the following sections we will make use of Assumption 6.1, hence constraints $\mathbb{X}_i$ and $\mathbb{U}_i$ can be defined as in (6.2), and we will use the definitions (8.2) and (8.3) for sets $\mathbb{D}_i$ and $\mathbb{O}_i$, respectively.

9.2.1 Adopted LSEs

We define an output feedback control strategy similar to the one proposed in [FS11a], based on the approach described in [MRFA06] for output-feedback robust tube-based MPC. The first step towards this goal consists in defining a suitable DSE for (9.1). This is done exactly as in Section 8.2.1 that is summarized next for the reader convenience.
We define for $i \in \mathcal{M}$ the LSE

$$\begin{aligned} \tilde{\Sigma}_{[i]}: \quad \tilde{x}^+_{[i]} =& A_{ii}\tilde{x}_{[i]} + B_i u_{[i]} - L_{ii}(y_{[i]} - C_i\tilde{x}_{[i]}) + \sum_{j\in\mathcal{N}_i} A_{ij}\tilde{x}_{[j]} - \\ &- \sum_{j\in\mathcal{N}_i} \tilde{\delta}_{ij} L_{ij}(y_{[j]} - C_j\tilde{x}_{[j]}) \end{aligned} \tag{9.2}$$

where $\tilde{x}_{[i]} \in \mathbb{R}^{n_i}$ is the state estimate, $L_{ij} \in \mathbb{R}^{n_i \times p_j}$ are gain matrices and $\tilde{\delta}_{ij} \in \{0, 1\}$. This implies that $\tilde{\Sigma}_{[i]}$ depends only on local variables ($\tilde{x}_{[i]}$, $u_{[i]}$ and $y_{[i]}$) and parents' variables ($\tilde{x}_{[j]}$ and $y_{[j]}$, $j \in \mathcal{N}_i$). Binary parameters $\tilde{\delta}_{ij}$, $j \in \mathcal{N}_i$ can be equal to one for exploiting the knowledge of parents' outputs, or equal to zero for reducing the number of transmitted output samples. Defining the state estimation error as

$$e_{[i]} = x_{[i]} - \tilde{x}_{[i]}, \tag{9.3}$$

from (1.2), (9.2) and (9.3), we obtain the local error dynamics

$$e^+_{[i]} = \bar{A}_{ii} e_{[i]} + \sum_{j \in \mathcal{N}_i} \bar{A}_{ij} e_{[j]} + D_i d_{[i]} + L_{ii} \varrho_{[i]} + \sum_{j \in \mathcal{N}_i} \tilde{\delta}_{ij} L_{ij} \varrho_{[j]} \tag{9.4}$$

where $\bar{A}_{ii} = A_{ii} + L_{ii} C_i$ and $\bar{A}_{ij} = A_{ij} + \tilde{\delta}_{ij} L_{ij} C_j$, $i \neq j$. Our first goal is to solve the following problem.

Problem 9.1. Design in a distributed fashion LSEs $\tilde{\Sigma}_{[i]}$, $i \in \mathcal{M}$ that

(a) are nominally convergent, i.e. when $\mathbb{D}_i = \{\mathbf{0}_{r_i}\}$ and $\mathbb{O}_i = \{\mathbf{0}_{p_i}\}$ it holds

$$||e_{[i]}(t)|| \to \mathbf{0}_{n_i} \text{ as } t \to \infty \tag{9.5}$$

(b) guarantee, for suitable initial conditions,

$$e_{[i]}(t) \in \mathbb{E}_i, \ \forall t \geq 0 \tag{9.6}$$

where $\mathbb{E}_i \subseteq \mathbb{R}^{n_i}$ are zonotopes centered at the origin given by

$$\begin{aligned} \mathbb{E}_i &= \{e_{[i]} \in \mathbb{R}^{n_i} : \mathcal{H}_i e_{[i]} \leq \mathbf{1}_{\bar{r}_i}\} \\ &= \{e_{[i]} \in \mathbb{R}^{n_i} : e_{[i]} = \Xi_i F^e_i, \ ||F^e_i||_\infty \leq 1\}. \end{aligned} \tag{9.7}$$

In (9.7), $\mathcal{H}_i = (h^T_{i,1}, \ldots, h^T_{i,\bar{r}_i}) \in \mathbb{R}^{\bar{r}_i \times n_i}$, $\text{rank}(\mathcal{H}_i) = n_i$, $\Xi_i \in \mathbb{R}^{n_i \times \bar{n}_i}$ and $F^e_i \in \mathbb{R}^{\bar{n}_i}$.

Defining the variable $\mathbf{e} = (e_{[1]}, \ldots, e_{[M]}) \in \mathbb{R}^n$, from (9.4) one obtains the collective dynamics of the estimation error

$$\mathbf{e}^+ = \bar{\mathbf{A}}\mathbf{e} + \mathbf{D}\mathbf{d} + \mathbf{L}\boldsymbol{\varrho} \tag{9.8}$$

where the matrices $\bar{\mathbf{A}}$ and $\mathbf{L}$ are composed by blocks $\bar{A}_{ij}$ and L_{ij}, $i, j \in \mathcal{M}$, respectively. We equip system (9.8) with constraints $\mathbf{e} \in \mathbb{E} = \prod_{i \in \mathcal{M}} \mathbb{E}_i$, $\mathbf{d} \in \mathbb{D}$ and $\boldsymbol{\varrho} \in \mathbb{O}$. From (9.8), if $\mathbf{L}$ is such that $\bar{\mathbf{A}}$ is Schur, then property (9.5) holds. Moreover, if there exists an RPI set $\mathbb{S} \subseteq \mathbb{E}$ for (9.8), then $\mathbf{e}(0) \in \mathbb{S}$ guarantees property (9.6). More specifically, we aim at defining a suitable "rectangular" invariant set $\mathbb{S} = \prod_{i \in \mathcal{M}} \mathbb{S}_i$ such that

(a) $\mathbb{S}_i \subseteq \mathbb{E}_i$ for all $i \in \mathcal{M}$

(b) if $e_{[i]}(0) \in \mathbb{S}_i$ for all $i \in \mathcal{M}$, then

$$e_{[i]}(t) \in \mathbb{S}_i,\ t \geq 0. \tag{9.9}$$

The design of LSEs will be presented later on in Section 9.3.1.

9.2.2 Definition of local controllers

The next step towards the synthesis of output-feedback local controllers $\tilde{\mathcal{C}}_{[i]}$, $i \in \mathcal{M}$ is to design a distributed controller robust to state estimation errors and coupling among subsystems. They will be jointly used with LSEs $\tilde{\Sigma}_{[i]}$. The main idea is to adopt robust state-feedback PnP-DeMPC controllers in Section 7.2 and

- replace states $x_{[i]}$ with their estimates $\tilde{x}_{[i]}$ (as remarked in [MRFA06], the goal of local MPC will be to control the state estimator (9.2));
- provide robustness to state estimation errors (instead of generic disturbances as in Section 7.2);
- add to the control law coupling attenuation terms (as in Section 7.3) for reducing conservativity of the approach. This also leads to a distributed control architectures.

For the sake of completeness, we detail below the construction of controllers $\tilde{\mathcal{C}}_{[i]}$.
We assume the controller has access to state estimation only and, therefore, if $e_{[i]}(t) = x_{[i]}(t) - \tilde{x}_{[i]}(t) \in \mathbb{S}_i$ for all $i \in \mathcal{M}$ and $t \geq 0$, then the following constraints must be guaranteed to ensure $x_{[i]} \in \mathbb{X}_i$

$$\tilde{x}_{[i]} \in \tilde{\mathbb{X}}_i = \mathbb{X}_i \ominus \mathbb{S}_i. \tag{9.10}$$

Since $\tilde{\mathbb{X}}_i$ contains the origin, without loss of generality, we can define $\tilde{\mathbb{X}}_i$ as

$$\tilde{\mathbb{X}}_i = \{\tilde{x}_{[i]} \in \mathbb{R}^{n_i} : \tilde{\mathcal{C}}_{x_i}\tilde{x}_{[i]} \leq \mathbf{1}_{\tilde{r}_i^x}\} \tag{9.11}$$

where $\tilde{\mathcal{C}}_{x_i} = (\tilde{c}_{x_{i,1}}^T, \dots, \tilde{c}_{x_{i,\tilde{r}_i^x}}^T) \in \mathbb{R}^{\tilde{r}_i^x \times n_i}$.
For the design of distributed regulators, we define a local nominal subsystem $\hat{\Sigma}_{[i]}$

$$\hat{\Sigma}_{[i]}: \quad \hat{x}_{[i]}^+ = A_{ii}\hat{x}_{[i]} + B_i v_{[i]} \tag{9.12}$$

where $v_{[i]} \in \mathbb{R}^{m_i}$ is the input. We want to confine $\tilde{x}_{[i]}$, the state estimates, in a tube of section $\mathbb{Z}_i$ (that will be defined in the sequel) centered in $\hat{x}_{[i]}$, i.e. to obtain that

$$\tilde{x}_{[i]}(0) \in \hat{x}_{[i]}(0) \oplus \mathbb{Z}_i \Rightarrow \tilde{x}_{[i]}(t) \in \hat{x}_{[i]}(t) \oplus \mathbb{Z}_i,\ \forall t \geq 0. \tag{9.13}$$

In order to achieve our aim we propose the following distributed controller (see also Sections 6.2.1 and 7.3)

$$\tilde{\mathcal{C}}_{[i]}: \quad u_{[i]} = v_{[i]} + \bar{\kappa}_i(\tilde{x}_{[i]} - \bar{x}_{[i]}) + \sum_{j\in\mathcal{N}_i} \delta_{ij} K_{ij} \tilde{x}_{[j]} \tag{9.14}$$

where $\bar{\kappa}_i(\cdot) : \mathbb{R}^{n_i} \to \mathbb{R}^{m_i}$ is a feedback control law that will allow us to guarantee (9.13), $K_{ij} \in \mathbb{R}^{m_i\times n_j}$ and $\delta_{ij} \in \{0, 1\}$, $i, j \in \mathcal{M}$. Note that, if $\delta_{ij} = 0$, $\forall i \in \mathcal{M}, \forall j \in \mathcal{N}_i$, the control scheme is completely decentralized, since each input $u_{[i]}$ depends upon state estimates of subsystem $\Sigma_{[i]}$ only. Defining $z_{[i]} = \tilde{x}_{[i]} - \hat{x}_{[i]}$, from (9.2), (9.12) and (9.14) (where we assume $\bar{x}_{[i]} = \hat{x}_{[i]}$), we obtain

$$z^+_{[i]} = A_{ii} z_{[i]} + B_i \bar{\kappa}_i(z_{[i]}) + \tilde{w}_{[i]} \tag{9.15}$$

$$\begin{aligned}\tilde{w}_{[i]} = &\sum_{j\in\mathcal{N}_i} (A_{ij} + \delta_{ij} B_i K_{ij})\tilde{x}_{[j]} - L_{ii} C_i e_{[i]} - \sum_{j\in\mathcal{N}_i} \delta_{ij} L_{ij} C_j e_{[j]} - \\ &- L_{ii}\varrho_{[i]} - \sum_{j\in\mathcal{N}_i} \delta_{ij} L_{ij} \varrho_{[j]}.\end{aligned} \tag{9.16}$$

Hence, in view of (6.2a), (9.9) and (9.10), we obtain that $\tilde{w}_{[i]}$ can be constrained as

$$\begin{aligned}\tilde{w}_{[i]} \in \tilde{\mathbb{W}}_i = &\bigoplus_{j\in\mathcal{N}_i} (A_{ij} + \delta_{ij} B_i K_{ij})\tilde{\mathbb{X}}_j \oplus (-L_{ii} C_i \mathbb{S}_i) \oplus \\ &\oplus \bigoplus_{j\in\mathcal{N}_i} \tilde{\delta}_{ij}(-L_{ij} C_j \mathbb{S}_j) \oplus (-L_{ii}\mathbb{O}_i) \oplus \bigoplus_{j\in\mathcal{N}_i} \tilde{\delta}_{ij}(-L_{ij}\mathbb{O}_j).\end{aligned} \tag{9.17}$$

As in [RM05], similarly to Sections 6.2.1 and 6.3, in order to guarantee (9.13), we look for a function $\bar{\kappa}_i(\cdot)$ and a set $\mathbb{Z}_i$, such that $\mathbb{Z}_i$ is an RPI set for the dynamics (9.15) with respect to the disturbance $\tilde{w}_{[i]}$.
Furthermore, following [RM05], in (9.14) we set

$$v_{[i]}(t) = v_{[i]}(0|t), \qquad \bar{x}_{[i]}(t) = \hat{x}_{[i]}(0|t) \tag{9.18}$$

where $v_{[i]}(0|t)$ and $\hat{x}_{[i]}(0|t)$ are optimal values of the variables $v_{[i]}(0)$ and $\hat{x}_{[i]}(0)$, respectively, appearing in the MPC-i problem

$$\mathbb{P}^N_i(\tilde{x}_{[i]}(t)):$$

$$\min_{\substack{\hat{x}_{[i]}(0) \\ v_{[i]}(0:\tilde{N}_i-1)}} \sum_{k=0}^{\tilde{N}_i-1} \tilde{\ell}_i(\hat{x}_{[i]}(k), v_{[i]}(k)) + \tilde{V}_{f_i}(\hat{x}_{[i]}(\tilde{N}_i)) \tag{9.19a}$$

$$\tilde{x}_{[i]}(t) - \hat{x}_{[i]}(0) \in \mathbb{Z}_i \tag{9.19b}$$

$$\hat{x}_{[i]}(k+1) = A_{ii}\hat{x}_{[i]}(k) + B_i v_{[i]}(k) \qquad k \in 0 : \tilde{N}_i - 1 \tag{9.19c}$$

$$\hat{x}_{[i]}(k) \in \hat{\mathbb{X}}_i,\ v_{[i]}(k) \in \mathbb{V}_i \qquad k \in 0 : \tilde{N}_i - 1 \tag{9.19d}$$

$$\hat{x}_{[i]}(\tilde{N}_i) \in \hat{\tilde{\mathbb{X}}}_{f_i} \tag{9.19e}$$

In (9.19), $\tilde{N}_i \in \mathbb{N}$ is the control horizon, $\tilde{\ell}_i : \mathbb{R}^{n_i \times m_i} \rightarrow \mathbb{R}_{0+}$ is the stage cost, $\tilde{V}_{f_i} : \mathbb{R}^{n_i} \rightarrow \mathbb{R}_{0+}$ is the final cost and $\hat{\tilde{\mathbb{X}}}$ is the terminal set. Note that in (9.18) we redefined the variables $\bar{x}_{[i]}$ in (9.14) that was previously set equal to $\hat{x}_{[i]}(t)$. The reason is that, from (9.19c), the nominal system $\hat{\Sigma}_{[i]}$, equipped with suitable constraints $\hat{x}_{[i]} \in \hat{\mathbb{X}}_i$ and $v_{[i]} \in \mathbb{V}_i$, is now used for obtaining the state predictions over the control horizon. Note also that the re-definition of $\bar{x}_{[i]}$ is at the core of the tube-MPC scheme proposed in [RM05].

As shown in [RM05], constraints (6.2) can be fulfilled using (9.14)-(9.19) if there exist sets $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$, $i \in \mathcal{M}$ verifying

$$\hat{\mathbb{X}}_i \oplus \mathbb{Z}_i \subseteq \tilde{\mathbb{X}}_i, \qquad \mathbb{V}_i \oplus \mathbb{U}_{z_i} \oplus \bigoplus_{j \in \mathcal{N}_i} \delta_{ij} K_{ij} \mathbb{X}_j \subseteq \mathbb{U}_i. \tag{9.20}$$

where $\mathbb{U}_{z_i} = \bar{\kappa}_i(\mathbb{Z}_i)$. The existence of such sets is guaranteed by the following assumptions.

Assumption 9.1. *Let δ_{ij} and K_{ij}, $i, j \in \mathcal{M}$. The set*

$$\bar{\mathbb{U}}_i = \mathbb{U}_i \ominus \bigoplus_{j \in \mathcal{N}_i} \delta_{ij} K_{ij} \mathbb{X}_j \tag{9.21}$$

is not empty.

Assumption 9.2. *There exist $\rho_{i,1} > 0$, $\rho_{i,2} > 0$ such that $\mathbb{Z}_i \oplus B_{\rho_{i,1}}(\mathbf{0}_{n_i}) \subseteq \tilde{\mathbb{X}}_i$ and $\mathbb{U}_{z_i} \oplus B_{\rho_{i,2}}(\mathbf{0}_{m_i}) \subseteq \bar{\mathbb{U}}_i$, where $B_{\rho_{i,1}}(\mathbf{0}_{n_i}) \subset \mathbb{R}^{n_i}$ and $B_{\rho_{i,2}}(\mathbf{0}_{m_i}) \subset \mathbb{R}^{m_i}$.*

Note that Assumption 9.2 implies that the coupling of subsystems connected in a cyclic fashion must be sufficiently small (see the discussion after Assumption 6.2).

In order to stabilize the origin of the closed-loop system, we introduce a customary assumption in MPC [RM09].

Assumption 9.3. *For all $i \in \mathcal{M}$, there exist an auxiliary control law $\tilde{\kappa}_i^{aux}(\hat{x}_{[i]})$ and a $\mathcal{K}_\infty$ function $\mathcal{B}_i$ such that:*

(i) $\tilde{\ell}_i(\hat{x}_{[i]}, v_{[i]}) \geq \mathcal{B}_i(||(\hat{x}_{[i]}, v_{[i]})||)$, *for all* $\hat{x}_{[i]} \in \mathbb{R}^{n_i}$, $v_{[i]} \in \mathbb{R}^{m_i}$ *and* $\tilde{\ell}_i(\boldsymbol{0}_{n_i}, \boldsymbol{0}_{m_i}) = 0$*;*

(ii) $\tilde{\hat{\mathbb{X}}}_{f_i} \subseteq \hat{\mathbb{X}}_i$ *is an invariant set for* $\hat{x}_{[i]}^+ = A_{ii}\hat{x}_{[i]} + B_i\tilde{\kappa}_i^{aux}(\hat{x}_{[i]})$*;*

(iii) $\forall \hat{x}_{[i]} \in \tilde{\hat{\mathbb{X}}}_{f_i}$, $\tilde{\kappa}_i^{aux}(\hat{x}_{[i]}) \in \mathbb{V}_i$*;*

(iv) $\forall \hat{x}_{[i]} \in \tilde{\hat{\mathbb{X}}}_{f_i}$, $\tilde{V}_{f_i}(\hat{x}_{[i]}^+) - \tilde{V}_{f_i}(\hat{x}_{[i]}) \leq -\tilde{\ell}_i(\hat{x}_{[i]}, \tilde{\kappa}_i^{aux}(\hat{x}_{[i]}))$.

We defer the reader to [RM09] for a review of methods for computing $\tilde{\ell}_i(\cdot)$, $\tilde{V}_{f_i}(\cdot)$ and $\tilde{\hat{\mathbb{X}}}_{f_i}$ verifying Assumption 9.3.

In summary, the controller $\tilde{\mathcal{C}}_{[i]}$ is given by (9.14), (9.18) and (9.19). Moreover the collective controller for (9.2) is decentralized or distributed depending on the choice of parameters δ_{ij}. The main problem that still has to be solved in the design of local controllers is the following one.

Problem 9.2. Compute nonempty RCI sets $\mathbb{Z}_i$, $i \in \mathcal{M}_i$, if they exist, verifying Assumption 9.2.

In the next section we show how to solve Problems 9.1 and 9.2, through distributed and computationally efficient algorithms. Moreover we study properties on system (9.1) equipped with LSE $\tilde{\Sigma}_{[i]}$ and distributed controllers $\tilde{\mathcal{C}}_{[i]}$.

9.3 Decentralization of output-feedback controllers design

In the following, we first solve Problem 9.1, then we design controllers $\mathcal{C}_{[i]}$ that allow one to solve Problem 9.2.

9.3.1 Design of LSE

In order to decentralize the synthesis of the state estimator, we can execute Algorithm 8.1 for designing in parallel all LSE $\tilde{\Sigma}_{[i]}$. We highlight that as sets $\mathbb{S}_i$, $i \in \mathcal{M}$, computed in Step (III) of Algorithm 8.1 increase, sets $\tilde{\mathbb{W}}_i$ in (9.17) increase as well. This in turn reduces the chances of successfully terminating Algorithm 9.1 (given in the next section) for the design of controllers $\tilde{\mathcal{C}}_{[i]}$. Therefore it is reasonable to compute $\mathbb{S}_i$ as an outer-approximation of the mRPI set for (9.4) (see methods proposed in [RKKM05]).

9.3.2 Design of $\tilde{\mathcal{C}}_{[i]}$

The aim of this section is to decentralize the design of distributed controllers $\tilde{\mathcal{C}}_{[i]}$ and solve Problem 9.2 in parallel for each subsystem. To this purpose, using the procedure proposed in Section VI of [RB10], we compute an RCI set $\mathbb{Z}_i \subset \tilde{\mathbb{X}}_i$ using an appropriate parametrization, similarly to Section 6.3. We define the set of variables θ_i as

$$\theta_i = \{\bar{z}_{[i]}^{(s,f)} \in \mathbb{R}^{n_i},\ \forall s \in 1:k_i,\ \forall f \in 1:q_i; \tag{9.22a}$$

$$\bar{u}_{[i]}^{(s,f)} \in \mathbb{R}^{m_i},\ \forall s \in 0:k_i-1,\ \forall f \in 1:q_i; \tag{9.22b}$$

$$\alpha_i \in \mathbb{R}\} \tag{9.22c}$$

where k_i, $q_i \in \mathbb{N}$ are parameters that can be chosen by the user as well as the set

$$\bar{\mathbb{Z}}_i^0 = \mathrm{convh}(\{\bar{z}_{[i]}^{(0,f)} \in \mathbb{R}^{n_i},\ \forall f \in 1:q_i\}) \tag{9.23}$$

where $\bar{z}_{[i]}^{(0,1)} = \mathbf{0}_{n_i}$. The next assumption on the choice of $\bar{\mathbb{Z}}_i^0$ is needed for the computation of sets $\mathbb{Z}_i$.

Assumption 9.4. *The set $\bar{\mathbb{Z}}_i^0$ is such that there is $\omega_i > 0$ verifying $\tilde{\mathbb{W}}_i \oplus B_{\omega_i}(\mathbf{0}_{n_i}) \subseteq \bar{\mathbb{Z}}_i^0$.*

Note that, since $\mathbf{0}_{n_i} \in \tilde{\mathbb{W}}_i$, Assumption 9.4 can be fulfilled only if $\mathbf{0}_{n_i} \in \bar{\mathbb{Z}}_i^0$, that is guaranteed by the use of $\bar{z}_{[i]}^{(0,1)} = \mathbf{0}_{n_i}$ in (9.23).
Let us define the sets

$$\bar{\mathbb{Z}}_i^s = \mathrm{convh}(\{\bar{z}_{[i]}^{(s,f)} \in \mathbb{R}^{n_i},\ \forall f \in 1:q_i\}), \forall s \in 1:k_i$$
$$\bar{\mathbb{U}}_{z_i}^p = \mathrm{convh}(\{\bar{u}_{[i]}^{(p,f)} \in \mathbb{R}^{m_i},\ \forall f \in 1:q_i\}), \forall p \in 0:k_i-1$$

where $\bar{z}_{[i]}^{(s,1)} = \mathbf{0}_{n_i}$, $\bar{u}_{[i]}^{(p,1)} = \mathbf{0}_{m_i}$, and consider the following optimization problem

$$\min_{\theta_i} \quad \alpha_i$$

$$\alpha_i < 1, \; -\alpha_i \leq 0 \tag{9.24a}$$

$$\bar{z}_{[i]}^{(s+1,f)} = A_{ii}\bar{z}_{[i]}^{(s,f)} + B_i\bar{u}_{[i]}^{(s,f)} \qquad \forall s \in 0 : k_i - 1, \; \forall f \in 1 : q_i \tag{9.24b}$$

$$\bar{z}_{[i]}^{(s,f)} \in \bar{\mathbb{Z}}_i^s \qquad \forall f \in 1 : q_i, \; \forall s \in 1 : k_i \tag{9.24c}$$

$$\bar{\mathbb{Z}}_i^{k_i} \subseteq \alpha_i \bar{\mathbb{Z}}_i^0 \tag{9.24d}$$

$$\bigoplus_{s=0}^{k_i-1} \bar{\mathbb{Z}}_i^s \subset (1-\alpha_i)\tilde{\mathbb{X}}_i \tag{9.24e}$$

$$\bar{u}_{[i]}^{(p,f)} \in \bar{\mathbb{U}}_i^p \qquad \forall f \in 1 : q_i, \; \forall p \in 0 : k_i - 1 \tag{9.24f}$$

$$\bigoplus_{p=0}^{k_i-1} \bar{\mathbb{U}}_{z_i}^p \subset \bar{\mathbb{U}}_i \tag{9.24g}$$

The relation between optimization problem (9.24) and the RCI sets $\mathbb{Z}_i$ is established in the next proposition.

Proposition 9.1. *Let Assumptions 1.1 and 9.4 hold and sets $\tilde{\mathbb{X}}_i$ and $\bar{\mathbb{U}}_i$ be defined as in (9.10) and (9.21) respectively. Let $k_i > 0$. If there exists an admissible θ_i for optimization problem (9.24), then*

$$\mathbb{Z}_i = (1-\alpha_i)^{-1} \bigoplus_{s=0}^{k_i-1} \bar{\mathbb{Z}}_i^s \tag{9.25}$$

is an RCI set and the corresponding set $\mathbb{U}_{z_i}$ is given by

$$\mathbb{U}_{z_i} = (1-\alpha_i)^{-1} \bigoplus_{p=0}^{k_i-1} \bar{\mathbb{U}}_{z_i}^p. \tag{9.26}$$

Proof. The proof directly follows from Section 6-a and Theorem 4.3 of [RB10]. □

We highlight that the set of constraints in the LP problem (9.24) depends only upon local fixed parameters $\{A_{ii}, B_i, \tilde{\mathbb{X}}_i, \bar{\mathbb{U}}_i, \mathbb{S}_i, \mathbb{O}_i\}$, fixed parameters $\{A_{ij}, \tilde{\mathbb{X}}_j, \mathbb{S}_j, \mathbb{O}_j\}_{j\in\mathcal{N}_i}$ of parents of $\Sigma_{[i]}$ and local tunable parameters θ_i (the decision variables (9.22)). Moreover, Θ_i does not depend on tunable parameters of parents controllers. This implies that the computation of sets $\mathbb{Z}_i$ and $\mathbb{U}_{z_i}$ in (9.25) and (9.26) does not influence the choice of $\mathbb{Z}_j$ and $\mathbb{U}_{z_j}$, $j \neq i$. Therefore Problem 9.2 is decomposed in the following independent LP problems for $i \in \mathcal{M}$.

Problem 9.3 ($\mathcal{PC}_i$)**.** Solve the LP problem (9.24).

If Problem $\mathcal{PC}_i$ is solved, then we can compute sets $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$ in (9.20) as

$$\hat{\mathbb{X}}_i = \tilde{\mathbb{X}}_i \ominus \mathbb{Z}_i, \qquad \mathbb{V}_i = \bar{\mathbb{U}}_i \ominus \mathbb{U}_{z_i}. \tag{9.27}$$

The overall procedure for the decentralized synthesis of local controllers $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ is given in Algorithm 9.1 whose properties are summarized in the next proposition.

Proposition 9.2. *Under Assumptions 1.1, 6.1 and definitions* (8.2) *and* (8.3)*, if, for all $i \in \mathcal{M}$, controllers $\tilde{\mathcal{C}}_{[i]}$ are designed according to Algorithm 9.1, then also Assumptions 9.1, 9.2, 9.3 and 9.4 are verified.*

Proof. Assumption 9.1 is enforced in Steps (I) and (II). Assumptions 9.3 and 9.4 are enforced in Steps (Viii) and (III) of Algorithm 9.1, respectively. Assumption 9.2 holds because of constraints (9.24e) and (9.24g) in the LP problem solved in step (IV) of Algorithm 9.1. □

In Step (I) of Algorithm 9.1, if $\delta_{ij} = 1$ the computation of matrix K_{ij} is required. For more details on the optimization problem (9.28), we refer the interested reader to Section 7.3.1. If the Algorithm stops in Step (II) a possible remedy is to restart it by increasing the number of parameters δ_{ij} that are equal to one or making parameters ξ_{ij} smaller. If in Step (IV) of Algorithm 9.1 the LP problem is infeasible, we can restart it with a different k_i, although there is no guarantee that the LP problem is feasible for some values of k_i [RB10]. Steps (Vi) and (Vii) of Algorithm 9.1, that provide constraints in (9.19), are the most computationally expensive because involve Minkowski sums and differences of polytopic sets. The interested reader is referred to Sections 6.3.1 and 6.3.2, where we show how to avoid burdensome computations exploiting results from [RB10] and how to compute a suitable function $\bar{\kappa}_i$ (appearing in (9.14)) through LP (see Section 6.14).

9.3.3 Analysis of the closed-loop system

In this section we give the main results on stability and constraints satisfaction for the overall system (9.1). First, we summarize how to design an output-feedback distributed controller in a parent-based decentralized fashion. The complete procedure is proposed in Algorithm 9.2.
In Algorithm 9.2 we also detail the flow of information from parent subsystems to their children in the design phase. We note that Algorithm 9.1,

Algorithm 9.1 Design of a state-feedback controller $\tilde{\mathcal{C}}_{[i]}$ for subsystem $\tilde{\Sigma}_{[i]}$

Input: A_{ii}, B_i, $\tilde{\mathbb{X}}_i$, $\mathbb{U}_i$, $\mathbb{S}_i$, $\mathbb{O}_i$, $\mathcal{N}_i$, $\{A_{ij}\}_{j\in\mathcal{N}_i}$, $\{\mathbb{O}_j\}_{j\in\mathcal{N}_i}$, $\{\mathbb{S}_j\}_{j\in\mathcal{N}_i}$, $\{\tilde{\mathbb{D}}_j\}_{j\in\mathcal{N}_i}$, $\{\xi_{ij}\}_{j\in\mathcal{N}_i}$, $\{\delta_{ij}\}_{j\in\mathcal{N}_i}$, $k_i > 0$

Output: controller $\tilde{\mathcal{C}}_{[i]}$ given by (9.14), (9.18) and (9.19)

(I) if $\delta_{ij} = 1$, compute the matrix K_{ij}, $\forall j \in \mathcal{N}_i$ solving

$$\min_{K_{ij}} ||\tilde{\mathcal{C}}_{x_i}(A_{ij} + \delta_{ij}B_iK_{ij})\tilde{\mathcal{C}}^{\flat}_{x_j}||_p \tag{9.28a}$$

$$||K_{ij}\tilde{\mathcal{C}}^{\flat}_{x_j}||_p \leq \xi_{ij} \tag{9.28b}$$

where p is a generic norm and scalars $\xi_{ij} > 0$ are given.

(II) Compute the set $\bar{\mathbb{U}}_i$ as in (9.21). If it is $\bar{\mathbb{U}}_i$ is empty **stop** (the controller $\tilde{\mathcal{C}}_{[i]}$ cannot be designed)

(III) Compute the set $\tilde{\mathbb{W}}_i$ as in (9.17) and choose $\bar{\mathbb{Z}}^0_i$ such that $\tilde{\mathbb{X}}_i \supseteq \bar{\mathbb{Z}}^0_i \supseteq \tilde{\mathbb{W}}_i \oplus B_{\omega_i}(\mathbf{0}_{n_i})$ for a sufficiently small $\omega_i > 0$. If $\bar{\mathbb{Z}}^0_i$ does not exist **stop** (the controller $\tilde{\mathcal{C}}_{[i]}$ cannot be designed)

(IV) Solve the LP problem (9.24). If it is unfeasible **stop** (the controller $\tilde{\mathcal{C}}_{[i]}$ cannot be designed).

(V) Design controller MPC-i by

 (i) Computing $\mathbb{Z}_i$ as in (9.25) and $\mathbb{U}_{z_i}$ as in (9.26).

 (ii) Computing $\hat{\mathbb{X}}_i$ and $\mathbb{V}_i$ as in (9.27).

 (iii) Choosing $\tilde{\ell}_i(\cdot)$, $\tilde{V}_{f_i}(\cdot)$ and $\tilde{\hat{\mathbb{X}}}_{f_i}$ verifying Assumption 9.3.

(VI) Choose the function $\bar{\kappa}_i$ in (9.14).

---

i.e. the Algorithm executed to compute controllers $\tilde{\mathcal{C}}_{[i]}$, must be executed after to Algorithm 8.1 since the computation of sets $\tilde{\mathbb{W}}_i$ in Algorithm 9.1 is based on the availability of set $\mathbb{S}_i$ that are computed in Algorithm 8.1. Defining the collective variables and matrices

$$\hat{\mathbf{x}} = (\hat{x}_{[1]}, \dots, \hat{x}_{[M]}) \in \mathbb{R}^n,\ \mathbf{v} = (v_{[1]}, \dots, v_{[M]}) \in \mathbb{R}^m,$$

$$\tilde{\mathbf{x}} = (\tilde{x}_{[1]}, \dots, \tilde{x}_{[M]}) \in \mathbb{R}^n,\ \mathbf{K} \in \mathbb{R}^{m\times n}$$

Algorithm 9.2 Design of an output-feedback controller for subsystem $\Sigma_{[i]}$

Input: A_{ii}, B_i, $\mathbb{X}_i$, $\mathbb{U}_i$, $\mathbb{D}_i$, $\mathbb{O}_i$, $\mathbb{E}_i$, $\mathcal{N}_i$, $\{A_{ij}\}_{j\in\mathcal{N}_i}$, $\{\xi_{ij}\}_{j\in\mathcal{N}_i}$, $\{\delta_{ij}\}_{j\in\mathcal{N}_i}$, $\{\tilde{\delta}_{ij}\}_{j\in\mathcal{N}_i}$, $k_i > 0$

Output: output-feedback controller composed by state estimator $\tilde{\Sigma}_{[i]}$ and state-feedback controller $\tilde{\mathcal{C}}_{[i]}$

(I) Send sets $\mathbb{E}_i$ to child subsystems $i \in \mathcal{N}_c$, $c \in \mathcal{M}$ and sets $\mathbb{O}_i$ if $\tilde{\delta}_{ci} = 1$

(II) Receive sets $\mathbb{E}_j$ from parent subsystems $j \in \mathcal{N}_i$ and sets $\mathbb{O}_j$ if $\tilde{\delta}_{ij} = 1$

(III) Execute Algorithm 8.1

(IV) Compute set $\tilde{\mathbb{X}}_i = \mathbb{X}_i \ominus \mathbb{S}_i$

(V) Send sets $\tilde{\mathbb{X}}_i$ to child subsystems $i \in \mathcal{N}_c$, $c \in \mathcal{M}$ and sets $\mathbb{S}_i$ if $\tilde{\delta}_{ci} = 1$

(VI) Receive sets $\tilde{\mathbb{X}}_j$ from parent subsystems $j \in \mathcal{N}_i$ and sets $\mathbb{S}_j$ if $\tilde{\delta}_{ij} = 1$

(VII) Execute Algorithm 9.1

---

where $\mathbf{K}$ collects matrices K_{ij} and the function

$$\bar{\boldsymbol{\kappa}}(x) = (\bar{\kappa}_1(x_{[1]}), \ldots, \bar{\kappa}_M(x_{[M]})) : \mathbb{R}^n \to \mathbb{R}^m,$$

from (9.2) one obtains the collective state estimator

$$\tilde{\mathbf{x}}^+ = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\mathbf{v} + \mathbf{B}\bar{\boldsymbol{\kappa}}(\tilde{\mathbf{x}} - \hat{\mathbf{x}}) + \mathbf{B}\mathbf{K}\tilde{\mathbf{x}} - \mathbf{L}\mathbf{C}\mathbf{e} - \mathbf{L}\boldsymbol{\varrho}.$$

Definition 9.1. The feasibility region for the MPC-i problem is

$$\tilde{\mathbb{X}}_i^N = \{s_{[i]} \in \tilde{\mathbb{X}}_i : \quad (9.19) \text{ is feasible for } \tilde{x}_{[i]}(t) = s_{[i]}\}$$

and the collective feasibility region is $\tilde{\mathbb{X}}^N = \prod_{i\in\mathcal{M}} \tilde{\mathbb{X}}_i^N$.

The next theorem summarizes the key properties of the closed-loop system.

Theorem 9.1. *Let Assumptions 1.1 and 6.1 hold and assume sets $\mathbb{D}_i$ and $\mathbb{O}_i$ are defined as in (8.2) and (8.3), respectively. Assume output-feedback controllers are computed using Algorithm 9.2 and define $\Xi_i = (\mathbb{Z}_i \oplus \mathbb{S}_i) \times \mathbb{Z}_i$, $\Xi = \prod_{i\in\mathcal{M}} \Xi_i$. Then, the set Ξ is robustly attractive for the closed-loop system with state $\boldsymbol{\xi} = (\xi_{[1]}, \ldots, \xi_{[M]})$, $\xi_{[i]} = (x_{[i]}, \tilde{x}_{[i]})$, $i \in \mathcal{M}$. Furthermore, a region of attraction of Ξ is $\prod_{i\in\mathcal{M}} (\tilde{\mathbb{X}}_i^N \oplus \mathbb{S}_i) \times (\tilde{\mathbb{X}}_i^N)$. Finally, if $\mathbf{d} = \mathbf{0}_r$ and $\boldsymbol{\varrho} = \mathbf{0}_p$, $\tilde{\mathbf{x}}(0) \in \tilde{\mathbb{X}}^N$ and $\mathbf{x}(0) - \tilde{\mathbf{x}}(0) \in \mathbb{S}$ imply that $\mathbf{x}(t) \to \mathbf{0}_n$ as $t \to \infty$.*

Proof. Consider the nominal case, i.e. $\mathbf{d} = \mathbf{0}_r$ and $\boldsymbol{\varrho} = \mathbf{0}_p$. Using the results in Chapter 8, the LSEs are asymptotically stable, hence $\mathbf{e}(t) \rightarrow \mathbf{0}_n$. Similarly to the proof of Theorem 6.1, we can state that $\tilde{\mathbf{x}}(t) \rightarrow \mathbf{0}_n$. Since $\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{e}$, we can conclude that $\mathbf{x}(t) \rightarrow \mathbf{0}_n$. If $\mathbf{d} \neq \mathbf{0}_r$ or $\boldsymbol{\varrho} \neq \mathbf{0}_p$, using the results in Chapter 8 and Section 7.2 and similar considerations as in Theorem 7.1, since $\mathbf{e}(t) \in \mathbb{S}$, $t \geq 0$ and $\text{dist}(\tilde{\mathbf{x}}(t), \mathbb{Z}) \rightarrow 0$, then the set $\mathbb{Z} \oplus \mathbb{S}$ is robustly attractive, i.e. $\text{dist}(\mathbf{x}(t), \mathbb{Z} \oplus \mathbb{S}) \rightarrow 0$. □

We note that Algorithm 9.1 provides a decentralized procedure in order to design distributed PnP regulators and it can be executed in parallel for all subsystems. Therefore, as shown in Sections 5.4 and 6.4, plug-in or unplugging operations involve only the update of a limited number of controllers. Differently from the design of state-feedback regulators, Algorithm 9.1 must be executed receiving sets computed by parent subsystems during the synthesis of LSEs. Therefore, we can design local controllers $\tilde{\mathcal{C}}_{[i]}$ only after all parents of subsystem $\Sigma_{[i]}$ have terminated the execution of Algorithm 8.1.

9.4 Example: power network system

In this section, we apply the proposed output-feedback PnPMPC scheme to the PNS proposed in Appendix B. In the following we first design the AGC layer for the PNS of Scenario 1 in B.1.1 and then we show how in presence of connection/disconnection of an area (Scenarios 2 and 3, in Sections B.1.2 and B.1.2, respectively) the AGC can be redesigned via plug in and out operations[1].

Similarly to Section 3.6, we assume to measure both $\Delta\theta_{[i]}$ and $\Delta\omega_{[i]}$, $i \in \mathcal{M}$, of each area. Therefore the outputs are given by

$$y_{[i]} = C_i x_{[i]}, \qquad C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Moreover we consider bounded disturbances

$$\mathbb{D}_i = \{d_{[i]} \in \mathbb{R}^{n_i} : \; ||d_{[i]}||_\infty \leq 5 \cdot 10^{-4}\}$$

and we require to keep the state estimation error of each area in the following set

$$\mathbb{E}_i = \{e_{[i]} \in \mathbb{R}^{n_i} : \; ||e_{[i]}||_\infty \leq 0.01\}.$$

[1] All simulations have been done using a MacOS 10.7.5, with processor Intel Core i5, 1.7 GHz, MatLab r2013a, solver CPLEX [IBM11], YALMIP [Lö4] and MPT [KGB04].

All examples and simulations are implemented using the *PnPMPC-toolbox* for Matlab [RBFT12] (see Appendix C).

9.4.1 Scenario 1

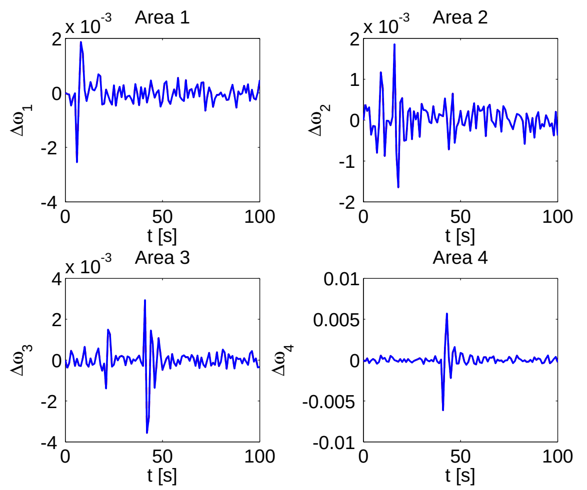
We consider the PNS proposed in Section B.1.2. For each system $\Sigma_{[i]}$ we synthesize the controller $\mathcal{C}_{[i]}$, $i \in \mathcal{M}$ using Algorithm 9.2. For the design of local estimators, we set $\tilde{\delta}_{ij} = 1$, $i \in 1:4$, $j \in \mathcal{N}_i$, that allow us to compute matrices L_{ij} such that $\bar{A}_{ij} = \mathbf{0}_{n_i \times n_j}$. For the design of local controllers we set $\delta_{ij} = 0$, therefore we do not use the state of parent subsystems in order to reduce the coupling terms.
In Figure 9.1a, 9.1b and 9.2 we show performance of the proposed output-feedback PnPMPC. We highlight that, differently from examples proposed in previous chapters, since each area is affected by bounded disturbances, the state and the input can not converge to the set-point $x^O_{[i]}$ and $u^O_{[i]}$ (see also Appendix B.2).

9.4.2 Scenario 2

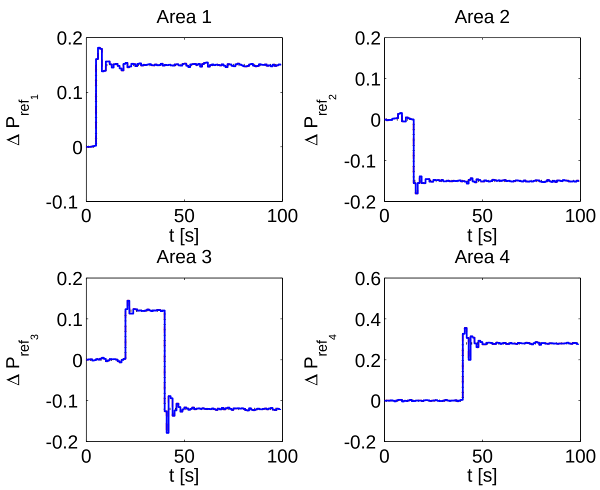
We consider the power network proposed in Scenario 1 and we add a fifth area connected as in Section B.1.2. Therefore, the set of children of subsystem 5 is $\mathcal{S}_5 = \{2, 4\}$. Since systems $\Sigma_{[j]}$, $j \in \mathcal{S}_5$ depend on a parameter related to the added system $\Sigma_{[5]}$, a retuning of their controllers is needed. For the design of local estimators, we set $\tilde{\delta}_{ij} = 1$, $i \in 1:5$, $j \in \mathcal{N}_i$, that allow us to compute matrices L_{ij} such that $\bar{A}_{ij} = \mathbf{0}_{n_i \times n_j}$. For the design of local controllers we set $\delta_{ij} = 0$, therefore we do not use the state of parent subsystems in order to reduce the coupling terms.
In Figure 9.3a, 9.3b and 9.4 we show performance of the proposed output-feedback PnPMPC. We highlight that, differently from examples proposed in previous chapters, since each area is affected by bounded disturbances, the state and the input can not converge to the set-point $x^O_{[i]}$ and $u^O_{[i]}$ (see also Appendix B.2).

9.4.3 Scenario 3

We consider the power network described in Scenario 2 and disconnect the area 4, as in Section B.1.3. The set of children of subsystem 4 is $\mathcal{S}_4 = \{3, 5\}$. Because of disconnection, subsystems $\Sigma_{[j]}$, $j \in \mathcal{S}_4$ change their parents and local dynamics A_{jj}. Then the retuning of controllers of child subsystems is needed. We highlight that retuning of controllers $\mathcal{C}_{[1]}$ and $\mathcal{C}_{[2]}$ is not needed



(a) Frequency deviation in each area controlled by the proposed output-feedback PnPMPC.



(b) Load reference set-point in each area controlled by the proposed output-feedback PnPMPC.

Figure 9.1: Simulation Scenario 1: 9.1a Frequency deviation and 9.1b Load reference in each area.

since systems $\Sigma_{[1]}$ and $\Sigma_{[2]}$ are not children of subsystem $\Sigma_{[4]}$. For the design of local estimators, we set $\tilde{\delta}_{ij} = 1$, $j \in \mathcal{N}_i$, that allow us to compute matrices L_{ij} such that $\bar{A}_{ij} = \mathbf{0}_{n_i \times n_j}$. For the design of local controllers we set $\delta_{ij} = 0$, therefore we do not use the state of parent subsystems in order

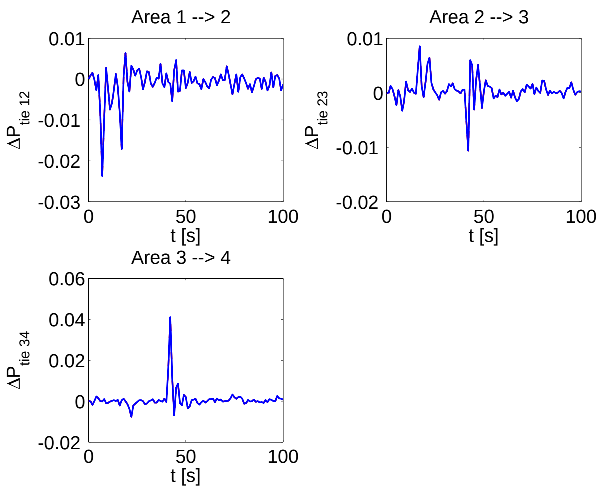
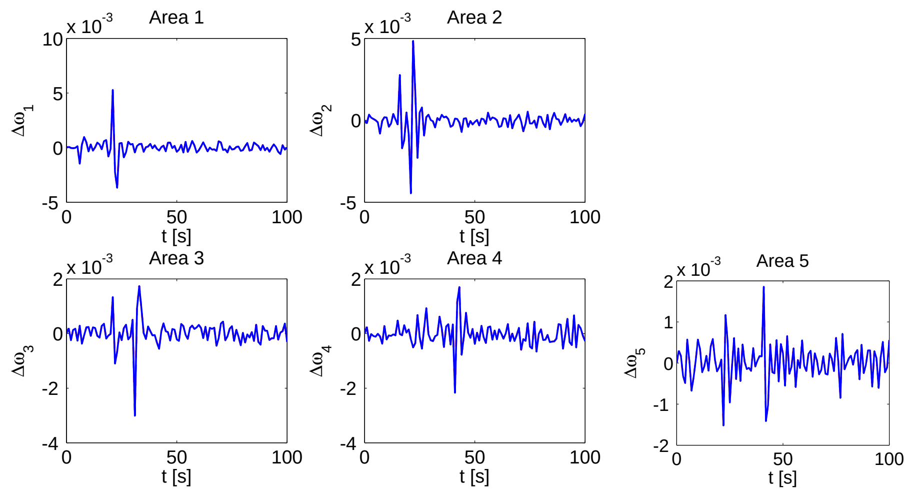


Figure 9.2: Simulation Scenario 1: tie-line power between each area controlled by the proposed output-feedback PnPMPC.

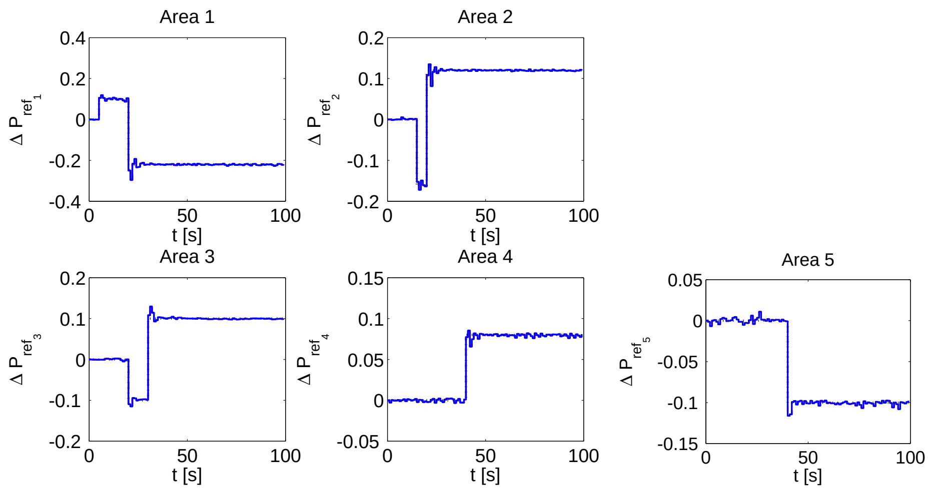
to reduce the coupling terms.
In Figure 9.5a, 9.5b and 9.6 we show performance of the proposed output-feedback PnPMPC. Differently from examples proposed in previous chapters, since each area is affected by bounded disturbances, the state and the input can not converge to the set-point $x^O_{[i]}$ and $u^O_{[i]}$ (see also Appendix B.2).

9.5 Final comments

In this chapter we proposed an output-feedback PnP-DiMPC scheme: we first designed PnP state estimators as in Chapter 8 and then we designed state-feedback PnP regulators in order to guarantee nominal attractiveness of the origin and constraint satisfaction for the closed-loop system. In future research we will address the problem of tracking of set-points instead of regulation to the origin.



(a) Frequency deviation in each area controlled by the proposed output-feedback PnPMPC.



(b) Load reference set-point in each area controlled by the proposed output-feedback PnPMPC.

Figure 9.3: Simulation Scenario 2: 9.3a Frequency deviation and 9.3b Load reference in each area.

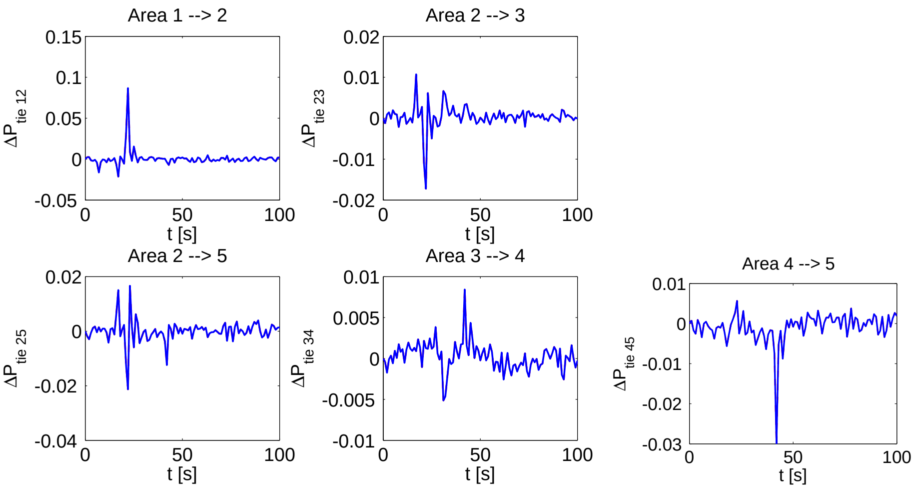
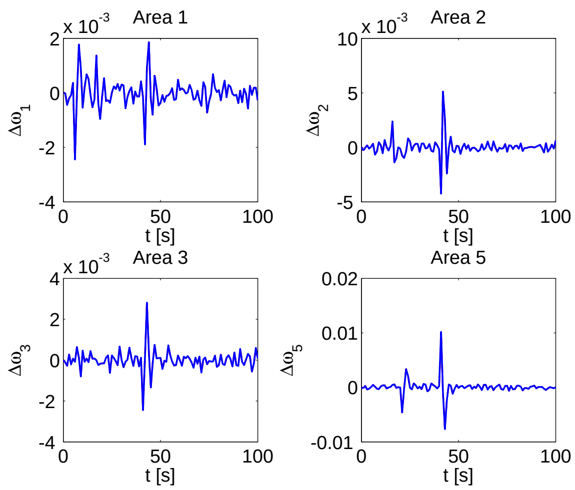
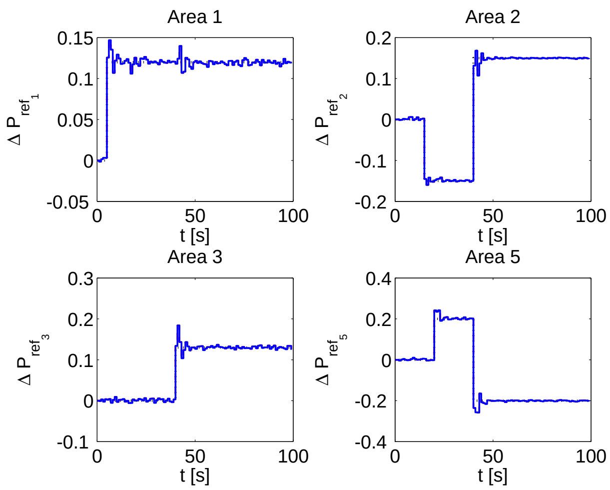


Figure 9.4: Simulation Scenario 2: tie-line power between each area controlled by the proposed output-feedback PnPMPC.



(a) Frequency deviation in each area controlled by the proposed output-feedback PnPMPC.



(b) Load reference set-point in each area controlled by the proposed output-feedback PnPMPC.

Figure 9.5: Simulation Scenario 3: 9.5a Frequency deviation and 9.5b Load reference in each area.

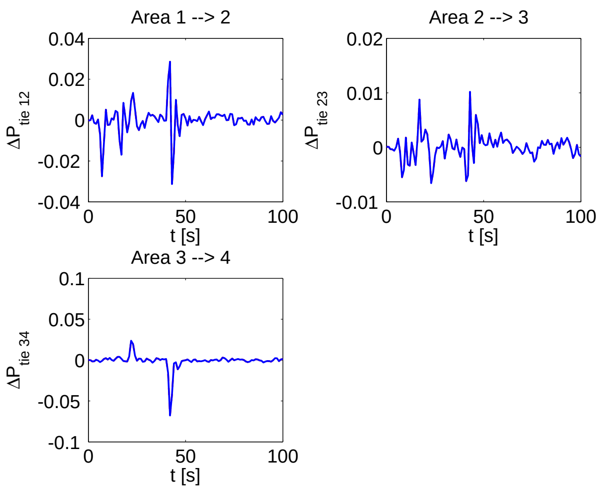


Figure 9.6: Simulation Scenario 3: tie-line power between each area controlled by the proposed output-feedback PnPMPC.

Chapter 10

Conclusions and future research

In this thesis we proposed decentralized and distributed control and state estimation schemes for constrained LTI LSSs.

Decentralized and distributed control architecture have recently attracted the attention of many researchers, in particular in the context of MPC [Sca09]. However an issue of most of the approaches available in the literature is that they cannot easily cope with the addition and removal of subsystems. In this thesis we proposed decentralized and PnP design procedures for MPC controllers. In particular, we gave contributions in two main directions. In Part I, we proposed an innovative DiMPC regulator and a state estimator that can be designed in a decentralized fashion. In Part II, we introduced novel PnP design methods for state-feedback MPC, distributed state estimators and output-feedback MPC.

The critical point of all decentralized and distributed MPC schemes is that the procedures for designing local controllers (or state estimators in the context of bounded-error state estimation) are conservative. This issue has been highlighted in Section 1.1.4: since the design of each controller (or state estimator) is based on different aims and the availability of local dynamics only, some degree of conservativity must be accepted. Moreover, it is often difficult to compare different MPC strategies (or state estimation algorithms), since they depend on the desired online features (communication topology and level of decentralization), design features (PnP capabilities) and the specific application under control.

In future research we will study in depth how to improve PnPMPC regulators in several directions.

- In Part II of the thesis, we proposed PnPMPC controllers guaranteeing asymptotic stability of the origin of the closed-loop system. In future research we will exploit methods proposed in [FBS13] and [BFS13], in order to design PnPMPC regulators capable of tracking set-points. More in details, the most promising approach seems to design in a PnP fashion decentralized and distributed controllers for systems described in "velocity-form" [PR01], [Wan08].

- In this thesis, we have shown how PnPMPC regulators allow PnP operations. In several man-made applications, subsystem dynamics can change due to the occurrence of a fault. Our future aim will be to equip subsystems with distributed fault detection schemes [BFPP11], in order to detect a fault in a subsystem and redesign its local controller in a PnP fashion to preserve closed-loop stability and constraints satisfaction of the overall system.

- PnP controllers can play a crucial role in microgrids. Microgrids are systems partitioned into several Distributed Generation Units (DGUs), where each DGU represents a renewable resource, such as solar panels, wind farms and microturbines. When microgrids operate in islanded mode (i.e. disconnected from the main grid), voltage control is of primary importance and local controllers must be resigned for guaranteeing voltage stability. In this context, the use of PnP control synthesis could pave they way to the design of scalable microgrids where DGUs owned by different players can enter or leave an existing grid without requiring substantial intervention of a central authority.

Finally, in this thesis addition and removal of subsystems have been considered as offline operations and further studies are required to allow for hot plugging, i.e. allow subsystems to enter and leave an existing network without shutting down the system. Preliminary results in this direction are available in [ZPR+13].

Part III

Appendices

Appendix A

Mathematical notation and definitions

Contents

A.1 Basic notation

$\mathbb{N}$	Set of integers, $\mathbb{N} = \{0, 1, 2, \ldots\}$.
$\mathbb{N}_+$	Set of positive integers, $\mathbb{N}_+ = \{1, 2, \ldots\}$.
$a : b$	Set of integers $\{a, a+1, \ldots, b\}$.
$\mathbb{R}$	Set of real numbers.
$\mathbb{R}_+$	Set of strictly positive real numbers.
$\mathbb{R}_{0+}$	Set of positive real numbers.
$\mathbf{v} = (v_1, \ldots, v_s)$	Column vector with s components.
$\mathrm{diag}(G_1, \ldots, G_s)$	Block-diagonal matrix composed by blocks G_i, $i = 1 : s$.

$\text{rank}(G)$	Rank of matrix G.
$\|\|G\|\|_p$	Norm p of matrix G.
$\bar{\rho}(G)$	Spectral radius of matrix G.
G^T	Transpose of matrix G.
$G^\flat$	Pseudo-inverse of matrix G.
$\mathbf{1}_\alpha$	Column vector with all α elements equal to 1.
$\mathbf{0}_\alpha$	Column vector with all α elements equal to 0.
$\mathbf{1}_{\alpha\times\beta}$	$\alpha \times \beta$ matrix with all elements equal to 1.
$\mathbf{0}_{\alpha\times\beta}$	$\alpha \times \beta$ matrix with all elements equal to 0.
$\mathbb{I}_n$	$n \times n$ identity matrix .
$\times$	Product of two sets, $\mathbb{X} \times \mathbb{Y}$.
$\prod$	Product of s sets $\mathbb{X}_i$, $\prod_{i=1}^{s} \mathbb{X}_i = \mathbb{X}_1 \times \ldots \times \mathbb{X}_s$.
$\oplus$	Minkowski sum of two sets, $\mathbb{X} \oplus \mathbb{Y}$.
$\bigoplus$	Minkowski sum of s sets $\mathbb{X}_i$, $\bigoplus_{i=1}^{s} \mathbb{X}_i = \mathbb{X}_1 \oplus \ldots \oplus \mathbb{X}_s$.
$\ominus$	Pontryagin difference of two set, $\mathbb{X} \ominus \mathbb{Y}$.
$\bigcup$	Set union.
$\bigcap$	Set intersection.
$\text{interior}(\mathbb{X})$	Interior of set $\mathbb{X}$.
$\text{convh}(\mathbb{X})$	Convex hull of set $\mathbb{X}$.
$\text{dist}(v, \mathbb{X})$	Distance among a vector v and a set $\mathbb{X}$, $\text{dist}(x, \mathbb{S}) = \inf_{s\in\mathbb{S}} \|\|x - s\|\|$.

A.2 Matrices

We use the following norms for vectors $\mathbf{v} \in \mathbb{R}^n$.

- *Infinity norm*: $||\mathbf{v}||_\infty = \max_{i=1:n} v_i$.
- *1-norm*: $||\mathbf{v}||_1 = \sum_{i=1}^{n} |v_i|$.
- *Euclidean norm*: $||\mathbf{v}||_2 = \sqrt{\sum_{i=1}^{n} v_i^2}$.
- *P-weighted seminorm*: $||x||_P = \sqrt{x^T P x}$, where P is positive semidefinite real symmetric matrix.

We use the following norms for matrices $A \in \mathbb{R}^{n\times m}$.

- *Infinity norm*: $||A||_\infty = \max_{i\in 1:n} \sum_{j=1}^{m} |a_{ij}|$.
- *1-norm*: $||A||_1 = \max_{j\in 1:m} \sum_{i=1}^{n} |a_{ij}|$.
- *Euclidean norm*: $||A||_2 = \sigma_{max}(A)$, where $\sigma_{max}(A)$ is the largest singular value of matrix A.

- *Frobenious norm*: $||A||_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^2}$.

Definition A.1 (Schur matrix)**.** A matrix $A \in \mathbb{R}^{n \times n}$ is Schur if all its eigenvalues λ verify $|\lambda| < 1$.

Definition A.2 (Hurwitz matrix)**.** A matrix $A \in \mathbb{R}^{n \times n}$ is Hurwitz if all its eigenvalues λ verify real$(\lambda) < 0$.

Definition A.3 (Metzler matrix [FR00])**.** A Metzler matrix is a matrix in which all the off-diagonal components are nonnegative.

In this thesis, we use some properties of Metzler matrices.

Lemma A.1 ([MS07])**.** *Let matrix* $\mathbb{M} \in \mathbb{R}^{M \times M}$ *be Metzler. Then* $\mathbb{M}$ *is Hurwitz if and only if there is a vector* $\nu \in \mathbb{R}_+^M$ *such that* $\mathbb{M}\nu < \mathbf{0}$*.*

Lemma A.2. *Define the matrix* $\Gamma = \mathbb{M} + \mathbb{I}_M$ *where* $\mathbb{M} \in \mathbb{R}^{M \times M}$, $\mathbb{I}_M$ *is the* $M \times M$ *identity matrix and* Γ *is non negative. Then the Metzler matrix* $\mathbb{M}$ *is Hurwitz if and only if* Γ *is Schur.*

The proof of Lemma A.2 easily follows from Theorem 13 in [FR00].

A.3 Sets

Definition A.4 (Polyhedron)**.** A polyhedron is the intersection of a finite number of closed half-spaces.

Definition A.5 (Polytope)**.** A polytope is a and bounded polyhedron.

A polytope can be represented through a list of inequalities, for example

$$\mathbb{X} = \{x \in \mathbb{R}^n : c_x x \le d_x\}$$

where $c_x \in \mathbb{R}^{r \times n}$ and $d_x \in \mathbb{R}^r$. Moreover if $0 \in \mathbb{X}$, there is always a matrix $\tilde{c}_x \in \mathbb{R}^{r \times n}$ such that

$$\mathbb{X} = \{x \in \mathbb{R}^n : \tilde{c}_x x \le \mathbf{1}_r\}.$$

Definition A.6 (Zonotope)**.** A zonotope is a centrally symmetric polytope.

Given a vector $p \in \mathbb{R}^n$ and a matrix $\Xi \in \mathbb{R}^{n \times s}$, the set

$$\mathbb{X} = \{x \in \mathbb{R}^n : x = p + \Xi\digamma, ||\digamma||_\infty \le 1\} \subset \mathbb{R}^n \tag{A.1}$$

where $\digamma \in \mathbb{R}^s$, is a zonotope. Furthermore every zonotope can be written as in (A.1) for a suitable vector p and a suitable matrix Ξ. More details on properties of zonotopes can be found in [RRS+12].

Definition A.7 (C-set)**.** A set $\mathbb{X} \subset \mathbb{R}^n$ is a C-set if it is compact, convex and contains the origin.

Definition A.8 (PC-set)**.** A set $\mathbb{X} \subset \mathbb{R}^n$ is a PC-set if it is a C-set and contains the origin in its nonempty interior.

Definition A.9 (Euclidean open ball)**.** $B_\delta(v)$ is the 2-norm open ball of radius $\delta > 0$ centered in $v \in \mathbb{R}^n$. If v is not specified then the open ball is centered in 0. Equivalently

$$B_\delta(v) = \{x \in \mathbb{R}^n : ||x - v||_2 < \delta\}.$$

Definition A.10 (Support function [KG98])**.** Given a set $\mathbb{X} \subset \mathbb{R}^n$, the support function evaluated at $c \in \mathbb{R}^n$ is defined as

$$\sup_{x\in\mathbb{X}} c^T x$$

If $\mathbb{X}$ is a zonotope the support function can be easily computed as

$$\sup_{x\in\mathbb{X}} c^T x = ||\Xi^T c + p||_1. \tag{A.2}$$

A.4 Functions

Definition A.11 ($\mathcal{K}$-function)**.** A function $\gamma : \mathbb{R}_{0+} \to \mathbb{R}_{0+}$ belongs to class $\mathcal{K}$ if it is continuous, zero at zero, and strictly increasing.

Definition A.12 ($\mathcal{K}_\infty$-function)**.** A function $\sigma : \mathbb{R}_{0+} \to \mathbb{R}_{0+}$ belongs to class $\mathcal{K}_\infty$ if it is a class $\mathcal{K}$ function and it is unbounded ($\sigma(s) \to \infty$ as $s \to \infty$).

Definition A.13 ($\mathcal{KL}$-function)**.** A function $\beta : \mathbb{R}_{0+} \times \mathbb{N} \to \mathbb{R}_{0+}$ belongs to class $\mathcal{KL}$ if (i) it is continuous, (ii) for each $t \geq 0$, $\beta(\cdot, t)$ is a class $\mathcal{K}$ function and (iii) for each $s \geq 0$, $\beta(s, \cdot)$ is non-increasing and satisfies $\lim_{t\to\infty} \beta(s,t) = 0$.

A.5 Set invariance

Next, we define invariant sets for nonlinear discrete-time systems. For more details, we refer the interested reader to [Bla99] and [BM08].

We consider a nonlinear discrete-time system $x^+ = f(x, u, w)$, where $x \in \mathbb{R}^n$, x^+ is the state x at time $t + 1$, $u \in \mathbb{R}^m$ and $w \in \mathbb{R}^r$. Moreover all variables can be constrained as $x \in \mathbb{X} \subseteq \mathbb{R}^n$, $u \in \mathbb{U} \subseteq \mathbb{R}^m$, $w \in \mathbb{W} \subseteq \mathbb{R}^r$.

Definition A.14 (Control invariant set)**.** A set $\mathbb{Z} \subseteq \mathbb{R}^n$ is a control invariant set for system $x^+ = f(x, u)$ and constraints $(\mathbb{X}, \mathbb{U})$ if $\mathbb{Z} \subseteq \mathbb{X}$ and, for every $x \in \mathbb{Z}$, there exists $u \in \mathbb{U}$ such that $f(x, u) \in \mathbb{X}$.

Definition A.15 (Positively invariant set)**.** A set $\mathbb{Z} \subseteq \mathbb{R}^n$ is a positively invariant set for system $x^+ = f(x)$ and constraints $\mathbb{X}$ if $\mathbb{Z} \subseteq \mathbb{X}$ and $f(x) \in \mathbb{X}$, for every $x \in \mathbb{Z}$.

Remark A.1. If $x^+ = Ax + Bu$, where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, control and positively invariant sets can be computed efficiently using methods proposed in [BM08], [RB09] and [RB10].

Definition A.16 (Robust Control Invariant (RCI) set)**.** A set $\mathbb{Z} \subseteq \mathbb{R}^n$ is a robust control invariant set for system $x^+ = f(x, u, w)$ and constraints $(\mathbb{X}, \mathbb{U}, \mathbb{W})$ if $\mathbb{Z} \subseteq \mathbb{X}$ and for every $x \in \mathbb{Z}$ there exists $u \in \mathbb{U}$ such that $f(x, u, w) \in \mathbb{X}$, for every $w \in \mathbb{W}$.

Definition A.17 (Robust Positively Invariant (RPI) set)**.** A set $\mathbb{Z} \subseteq \mathbb{R}^n$ is a robust positively invariant set for system $x^+ = f(x, w)$ and constraints $(\mathbb{X}, \mathbb{W})$ if $\mathbb{Z} \subseteq \mathbb{X}$ and $f(x, w) \in \mathbb{X}$, for every $x \in \mathbb{Z}$ and $w \in \mathbb{W}$.

Remark A.2. If $x^+ = Ax + Bu + w$, where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, RPI and RCI sets can be computed efficiently using methods proposed in [BM08] and [RB10].

Definition A.18 (minimal Robust Positively Invariant (mRPI) set)**.** A set $\underline{\mathbb{Z}} \subseteq \mathbb{R}^n$ is the minimal robust positively invariant set for system $x^+ = f(x, w)$ and constraints $(\mathbb{X}, \mathbb{W})$ if every other RPI $\mathbb{Z}$ verifies $\underline{\mathbb{Z}} \subseteq \mathbb{Z}$.

Definition A.19 (Maximal Robust Positively Invariant (MRPI) set)**.** A set $\bar{\mathbb{Z}} \subseteq \mathbb{R}^n$ is the maximal robust positively invariant set for system $x^+ = f(x, w)$ and constraints $(\mathbb{X}, \mathbb{W})$ if every other RPI $\mathbb{Z}$ verifies $\mathbb{Z} \subseteq \bar{\mathbb{Z}}$.

Definition A.20 (ϵ-outer approximation of minimal Robust Positively Invariant (ϵ-mRPI) set)**.** The RPI set $\mathbb{Z}(\epsilon)$ is a δ-outer approximation of the mRPI $\underline{\mathbb{Z}}$ for system $x^+ = f(x, w)$ and constraints $(\mathbb{X}, \mathbb{W})$ if

$$x \in \mathbb{Z}(\epsilon) \Rightarrow \exists\ \underline{x} \in \underline{\mathbb{Z}} \text{ and } \sigma \in B_\epsilon(0) : x = \underline{x} + \sigma. \tag{A.3}$$

where $B_\epsilon(v)$ is the 2-norm open ball of radius ϵ centered in $v \in \mathbb{R}^n$.

Remark A.3. If $x^+ = Ax + w$, $A \in \mathbb{R}^{n \times n}$, ϵ-mRPI sets can be computed efficiently using methods proposed in [RKKM05].

In the next definitions λ is a scalar in the interval $[0, 1)$.

Definition A.21 (λ-contractive control invariant set)**.** The set $\mathbb{Z} \subseteq \mathbb{R}^n$ is a λ-contractive control invariant set for system $x^+ = f(x, u)$ and constraints $(\mathbb{X}, \mathbb{U})$ if $\mathbb{Z} \subseteq \mathbb{X}$ and, for every $x \in \mathbb{Z}$, there exists $u \in \mathbb{U}$ such that $f(x, u) \in \lambda\mathbb{Z}$.

Definition A.22 (λ-contractive positively invariant set)**.** The set $\mathbb{Z} \subseteq \mathbb{R}^n$ is a λ-contractive invariant set for system $x^+ = f(x)$ and constraints $\mathbb{X}$ if $\mathbb{Z} \subseteq \mathbb{X}$ and $f(x) \in \lambda\mathbb{Z}$, for every $x \in \mathbb{Z}$.

Definition A.23 (λ-contractive RCI set)**.** The set $\mathbb{Z} \subseteq \mathbb{R}^n$ is a λ-contractive RCI set for system $x^+ = f(x, u, w)$ and constraints $(\mathbb{X}, \mathbb{U}, \mathbb{W})$ if $\mathbb{Z} \subseteq \mathbb{X}$ and, for every $x \in \mathbb{Z}$, there exists $u \in \mathbb{U}$ such that $f(x, u, w) \in \lambda\mathbb{Z}$, for every $w \in \mathbb{W}$.

Definition A.24 (λ-contractive RPI set)**.** The set $\mathbb{Z} \subseteq \mathbb{R}^n$ is a λ-contractive RPI set for system $x^+ = f(x, w)$ and constraint $(\mathbb{X}, \mathbb{W})$ if $\mathbb{Z} \subseteq \mathbb{X}$ and $f(x) \in \lambda\mathbb{Z}$, for every $x \in \mathbb{Z}$ and $w \in \mathbb{W}$.

Remark A.4. If $x^+ = Ax + Bu + w$, where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, λ-contractive sets can be computed efficiently using methods proposed in [BM08] and [RB10].

Appendix B

Benchmark power network systems

Contents

B.1 Introduction

An example of a real application that can benefit of decentralized and distributed control schemes is the regulation of a PNS. Here we describe the PNS proposed as a benchmark exercise [RFT12a] within the HYCON2 project [Hyc10].
We consider a PNS as composed by several power generation areas coupled through tie-lines [Saa02]. The aim is to design the AGC layer for frequency control with the goal of:

- keeping the frequency approximately at the nominal value;

- controlling the tie-line powers in order to reduce power exchanges between areas. In the asymptotic regime each area should compensate for local load steps and produce the required power.

We consider thermal power stations with single-stage turbines. The dynamics of an area equipped with primary control and linearized around equilibrium value for all variables can be described by the following continuous-time LTI model [Saa02]

$$\Sigma_{[i]}^{C}: \quad \dot{x}_{[i]} = A_{ii}x_{[i]} + B_i u_{[i]} + L_i \Delta P_{L_i} + \sum_{j\in\mathcal{N}_i} A_{ij}x_{[j]} \tag{B.1}$$

where $x_{[i]} = (\Delta\theta_i,\ \Delta\omega_i,\ \Delta P_{m_i},\ \Delta P_{v_i})$ is the state, $u_{[i]} = \Delta P_{ref_i}$ is the control input of each area, ΔP_{L_i} is the local power load and $\mathcal{N}_i$ is the sets of parent areas, i.e. areas directly connected to $\Sigma_{[i]}^{C}$ through tie-lines. The matrices of system (B.1) are defined as

$$A_{ii}(\{P_{ij}\}_{j\in\mathcal{N}_i}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{\sum_{j\in\mathcal{N}_i} P_{ij}}{2H_i} & -\frac{D_i}{2H_i} & \frac{1}{2H_i} & 0 \\ 0 & 0 & -\frac{1}{T_{t_i}} & \frac{1}{T_{t_i}} \\ 0 & -\frac{1}{R_i T_{g_i}} & 0 & -\frac{1}{T_{g_i}} \end{bmatrix} \quad B_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{T_{g_i}} \end{bmatrix}$$
$$A_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{P_{ij}}{2H_i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad L_i = \begin{bmatrix} 0 \\ -\frac{1}{2H_i} \\ 0 \\ 0 \end{bmatrix} \tag{B.2}$$

For the meaning of constants as well as some typical parameter values we refer the reader to Table B.1. We note that model (B.1) is input decoupled since both ΔP_{ref_i} and ΔP_{L_i} act only on subsystem $\Sigma_{[i]}^{C}$. Moreover, subsystems $\Sigma_{[i]}^{C}$ are parameter dependent since the local dynamics depends on the quantities $-\frac{\sum_{j\in\mathcal{N}_i} P_{ij}}{2H_i}$. One can consider different power plants and turbines:

- thermal power station and turbines with double stage

$$\Delta P_{m_i}(s) = \frac{1}{1+sT_{g_i}} \frac{1+s\alpha T_{rh_i}}{1+sT_{rh_i}} \Delta P_{v_i}(s);$$

$\Delta\theta_i$	Deviation of the angular displacement of the rotor with respect to the stationary reference axis on the stator
$\Delta\omega_i$	Speed deviation of rotating mass from nominal value
ΔP_{m_i}	Deviation of the mechanical power from nominal value (p.u.)
ΔP_{v_i}	Deviation of the steam valve position from nominal value (p.u.)
ΔP_{ref_i}	Deviation of the reference set power from nominal value (p.u.)
ΔP_{L_i}	Deviation of the nonfrequency-sensitive load change from nominal value (p.u.)
H_i	Inertia constant defined as $H_i = \frac{\text{kinetic energy at rated speed}}{\text{machine rating}}$ (typically values in range $[1 - 10]$ sec)
R_i	Speed regulation
D_i	Defined as $\frac{\text{percent change in load}}{\text{change in frequency}}$
T_{t_i}	Prime mover time constant (typically values in range $[0.2 - 2]$ sec)
T_{g_i}	Governor time constant (typically values in range $[0.1 - 0.6]$ sec)
P_{ij}	Slope of the power angle curve at the initial operating angle between area i and area j

Table B.1: Variables of a generation area with typical value ranges [Saa02]. (p.u.) stands for "per unit".

- hydroelectric power station and turbines using incompressible fluid

$$\Delta P_{m_i}(s) = \frac{1 - sT_{w_i}}{1 + sT_{w_i}} \Delta P_{v_i}(s);$$

- hydroelectric power station and turbines using compressible fluid

$$\Delta P_{m_i}(s) = \frac{1 - 2\mu \tanh(s\frac{\tau}{2})}{1 + \mu \tanh(s\frac{\tau}{2})} \Delta P_{v_i}(s)$$

where T_{rh_i} is the time constant of the superheater, α is the level of parallelization, T_{w_i} is the time constant of water starting in conduct, τ is the period of the water hammer and μ is the Allievi's coefficient.
In the following we introduce three scenarios corresponding to different interconnection topologies of generation areas. The model parameters and constraints on $\Delta\theta_i$ and on ΔP_{ref_i} for systems in all scenarios are given in Table B.2. We highlight that all parameter values are within the range of those used in Chapter 12 of [Saa02]. We define M as the number of areas in the power network. For each scenario, discrete-time models $\Sigma_{[i]}$ with $T_s = 1$ sec sampling time are obtained from $\Sigma_{[i]}^C$ using two alternative discretization schemes:

- exact discretization of the overall system (acronym D);

- discretization system-by-system, i.e. exact discretization for each area treating $u_{[i]}$, ΔP_{L_i} and $x_{[j]}$, $j \in \mathcal{N}_i$ as exogenous inputs (acronym Dss).

In particular, we note that Dss preserves the input-decoupled structure of $\Sigma^C_{[i]}$ while D does not. Relevant properties of the Dss discretization can be found in [FCS13]. Modeling and discretization can been performed in

	Area 1	Area 2	Area 3	Area 4	Area 5
H_i	12	10	8	8	10
R_i	0.05	0.0625	0.08	0.08	0.05
D_i	0.7	0.9	0.9	0.7	0.86
T_{t_i}	0.65	0.4	0.3	0.6	0.8
T_{g_i}	0.1	0.1	0.1	0.1	0.15

	Area 1	Area 2	Area 3	Area 4	Area 5
$\Delta\theta_i$	$\|\|x_{[1,1]}\|\|_\infty \leq 0.1$	$\|\|x_{[2,1]}\|\|_\infty \leq 0.1$	$\|\|x_{[3,1]}\|\|_\infty \leq 0.1$	$\|\|x_{[4,1]}\|\|_\infty \leq 0.1$	$\|\|x_{[5,1]}\|\|_\infty \leq 0.1$
ΔP_{ref_i}	$\|\|u_{[1]}\|\|_\infty \leq 0.5$	$\|\|u_{[2]}\|\|_\infty \leq 0.65$	$\|\|u_{[3]}\|\|_\infty \leq 0.65$	$\|\|u_{[4]}\|\|_\infty \leq 0.55$	$\|\|u_{[5]}\|\|_\infty \leq 0.5$

$$P_{12} = 4 \qquad P_{23} = 2 \qquad P_{34} = 2 \qquad P_{45} = 3 \qquad P_{25} = 3$$

Table B.2: Model parameters and constraints for systems $\Sigma_{[i]}$, $i \in 1:5$.

MatLab using the *PnPMPC-toolbox* that offers facilities for handling the interconnections of constrained subsystems [RBFT12].

B.1.1 Scenario 1

We consider four areas interconnected as in Figure B.1. We will simulate

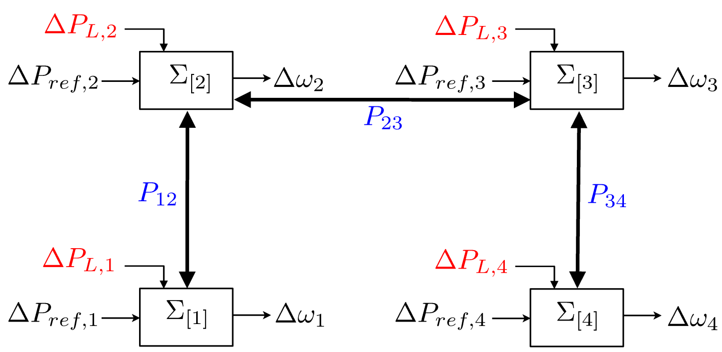


Figure B.1: Power network system of Scenario 1

Scenario 1 using the load steps specified in Table B.3.

Step time	Area i	ΔP_{L_i}
5	1	+0.15
15	2	-0.15
20	3	+0.12
40	3	-0.12
40	4	+0.28

Table B.3: Load of power ΔP_{L_i} (p.u.) for simulation in Scenario 1. $+\Delta P_{L_i}$ means a step of required power, hence a decrease of the frequency deviation $\Delta\omega_i$ and therefore an increase of the power reference ΔP_{ref_i}.

B.1.2 Scenario 2

We consider the power network proposed in Scenario 1 and add a fifth area connected as in Figure B.2. We will simulate Scenario 2 using the load

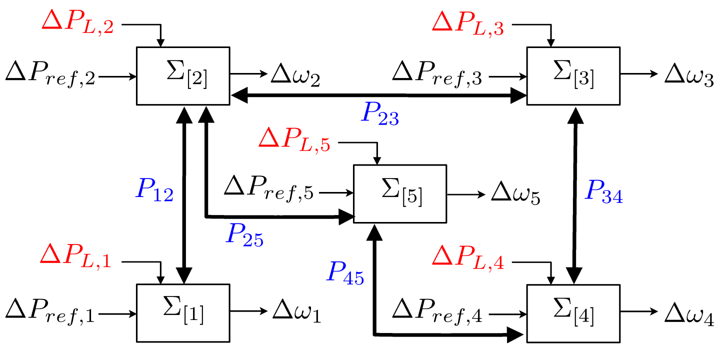


Figure B.2: Power network system of Scenario 2

steps specified in Table B.4.

B.1.3 Scenario 3

We consider the power network described in Scenario 2 and disconnect the area 4, hence obtaining the areas connected as in Figure B.3. We will simulate Scenario 3 using load steps specified in Table B.5.

Step time	Area i	ΔP_{L_i}
5	1	+0.10
15	2	-0.16
20	1	-0.22
20	2	+0.12
20	3	-0.10
30	3	+0.10
40	4	+0.08
40	5	-0.10

Table B.4: Load of power ΔP_{L_i} (p.u.) for simulation in Scenario 2. $+\Delta P_{L_i}$ means a step of required power, hence a decrease of the frequency deviation $\Delta\omega_i$ and therefore an increase of the power reference ΔP_{ref_i}.

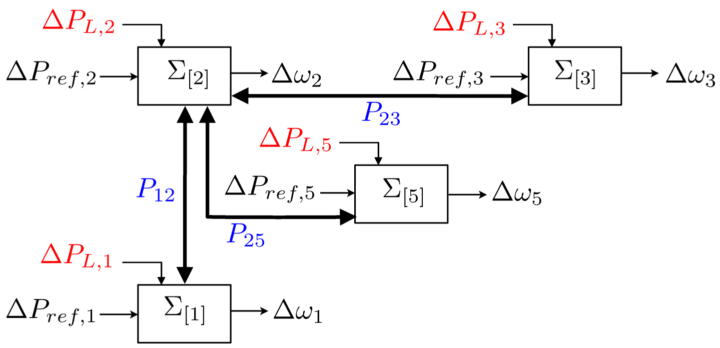


Figure B.3: Power network system of Scenario 3

Step time	Area i	ΔP_{L_i}
5	1	+0.12
15	2	-0.15
20	5	+0.20
40	2	+0.15
40	3	+0.13
40	5	-0.20

Table B.5: Load of power ΔP_{L_i} (p.u.) for simulation in Scenario 3. $+\Delta P_{L_i}$ means a step of required power, hence a decrease of the frequency deviation $\Delta\omega_i$ and therefore an increase of the power reference ΔP_{ref_i}.

B.2 Design of the AGC layer for a PNS using MPC

The goal of the benchmark is to design the AGC layer for the scenarios introduced in Section B.1. Different control schemes will be compared with the CeMPC scheme described next.
For a given Scenario, at time t we solve the centralized optimization problem

$$\mathbb{P}^N(\mathbf{x}(t), \mathbf{u}(t:t+N-1)) :$$

$$\min_{\mathbf{u}(t:t+N-1)} \sum_{k=t}^{t+N-1} (||\mathbf{x}(k) - \mathbf{x^O}||_{\mathbf{Q}} + ||\mathbf{u}(k) - \mathbf{u^O}||_{\mathbf{R}}) + ||\mathbf{x}(t+N) - \mathbf{x^O}||_{\mathbf{S}} \tag{B.3a}$$

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}\boldsymbol{\Delta}\mathbf{P_L}(t) \qquad k \in t : t+N-1 \tag{B.3b}$$

$$\mathbf{x}(k) \in \mathbb{X} \qquad k \in t : t+N-1 \tag{B.3c}$$

$$\mathbf{u}(k) \in \mathbb{U} \qquad k \in t : t+N-1 \tag{B.3d}$$

$$\mathbf{x}(t+N) \in \mathbb{X}_f \tag{B.3e}$$

and then apply $\boldsymbol{\Delta}\mathbf{P_{ref}} = \mathbf{u}(0)$. We note that the bold quantities in (B.3) collect the dynamics (B.1) of each area, moreover the cost function depend upon $\mathbf{x^O}$ and $\mathbf{u^O}$ that are defined as $x^O_{[i]} = (0,\ 0,\ \Delta P_{L_i},\ \Delta P_{L_i})$ and $u^O_{[i]} = \Delta P_{L_i}$. The constraints $\mathbb{X}$ and $\mathbb{U}$ in (B.3c) and (B.3d) are obtained from constraints listed in Table B.2. In the cost function (B.3a) we set $N = 15$, $\mathbf{Q} = \text{diag}(Q_1, \ldots, Q_M)$ and $\mathbf{R} = \text{diag}(R_1, \ldots, R_M)$, where

$$Q_i = \begin{bmatrix} 500 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \text{ and } R_i = 10. \tag{B.4}$$

Weights Q_i and R_i have been chosen in order to penalize the angular displacement $\Delta\theta_i$ and to penalize slow reactions to power load steps. Since the power transfer between areas i and j is given by

$$\Delta P_{tie_{ij}}(k) = P_{ij}(\Delta\theta_i(k) - \Delta\theta_j(k)) \tag{B.5}$$

the first requirement also penalizes huge power transfers.
In order to guarantee the stability of the closed loop system, we design the matrix $\mathbf{S}$ and the terminal constraint set $\mathbb{X}_f$ in three different ways.

- **S** *is full (MPCfull)*: we compute the symmetric positive-definite matrix **S** and the static state-feedback auxiliary control law $\mathbf{K}_{aux}\mathbf{x}$, by maximizing the volume of the ellipsoid described by **S** inside the state constraints while fulfilling the matrix inequality $(\mathbf{A}+\mathbf{B}\mathbf{K_{aux}})'\mathbf{S}(\mathbf{A}+\mathbf{B}\mathbf{K_{aux}})-\mathbf{S}\leq -\mathbf{Q}-\mathbf{K}'_{\mathbf{aux}}\mathbf{R}\mathbf{K_{aux}}$.

- **S** *is block diagonal (MPCdiag)*: we compute the decentralized symmetric positive-definite matrix **S** and the decentralized static state-feedback auxiliary control law $\mathbf{K_{aux}}\mathbf{x}$, $\mathbf{K_{aux}}=\text{diag}(K_1,\ldots,K_M)$ by maximizing the volume of the ellipsoid described by **S** inside the state constraints while fulfilling the matrix inequality $(\mathbf{A}+\mathbf{B}\mathbf{K_{aux}})'\mathbf{S}(\mathbf{A}+\mathbf{B}\mathbf{K_{aux}})-\mathbf{S}\leq -\mathbf{Q}-\mathbf{K}'_{\mathbf{aux}}\mathbf{R}\mathbf{K_{aux}}$.

- *Zero terminal constraint (MPCzero)*: we set $\mathbf{S}=\mathbf{0}_{n\times n}$ and $\mathbb{X}_f=\mathbf{x}^{\mathbf{O}}$.

B.2.1 Performance criteria

We propose the following performance criteria for evaluating different control schemes.

- *η-index*

$$\eta=\frac{1}{T_{sim}}\sum_{k=0}^{T_{sim}-1}\sum_{i=1}^{M}(||x_{[i]}(k)-x^O_{[i]}(k)||_{Q_i}+||u_{[i]}(k)-u^O_{[i]}(k)||_{R_i}) \quad \text{(B.6)}$$

  where T_{sim} is the time of the simulation. From (B.6), η is a weighted average of the error between the real state and the equilibrium state and between the real input and the equilibrium input.

- *Φ-index*

$$\Phi=\frac{1}{T_{sim}}\sum_{k=0}^{T_{sim}-1}\sum_{i=1}^{M}\sum_{j\in\mathcal{N}_i}|\Delta P_{tie_{ij}}(k)|T_s \quad \text{(B.7)}$$

  where T_{sim} is the time of the simulation and $\Delta P_{tie_{ij}}$ is the power transfer between areas i and j defined in (B.5). This index gives the average power transferred between areas. In particular, if the η-index is equal for two regulators, the best controller is the one that has the lower value of Φ.

B.3 Control experiments

We applied the CeMPC schemes introduced in the previous section to Scenarios 1, 2 and 3. Furthermore, for each scenario we discretized the continuos system with both discretization schemes D and Dss. At time t we solve the optimization problem (B.3) and then apply the control action to the continuos-time system, keeping the value constant between time t and $t+1$. If at time t the power load increases or decreases, we assume the controller can use this information at time t. This means at time t the controller knows exactly the value of $\mathbf{\Delta P_L}$ hence can use it. We highlight that violation of this assumption can impact considerably on the index η. In all experiments we use $T_{sim} = 100$. In Table B.6 and B.7 the values of the performance parameters η and Φ, respectively, are reported for each control experiment.[1]

	Scenario 1		Scenario 2		Scenario 3	
	D	Dss	D	Dss	D	Dss
$MPCfull$	0.0249	0.0249	0.0346	0.0347	0.0510	0.0511
$MPCdiag$	0.0249	0.0249	0.0346	0.0347	0.0510	0.0511
$MPCzero$	0.0249	0.0249	0.0346	0.0347	0.0510	0.0511

Table B.6: Values of the performance parameter η using different centralized MPC schemes for the AGC layer.

	Scenario 1		Scenario 2		Scenario 3	
	D	Dss	D	Dss	D	Dss
$MPCfull$	0.0030	0.0029	0.0063	0.0060	0.0060	0.0058
$MPCdiag$	0.0030	0.0029	0.0063	0.0061	0.0060	0.0058
$MPCzero$	0.0030	0.0028	0.0063	0.0059	0.0059	0.0058

Table B.7: Values of the performance parameter Φ using different CeMPC schemes for the AGC layer.

B.4 Measures of control performance

The aim is to design decentralized or distributed controllers for the scenarios described in Section B.1.

[1] All simulations have been done using a MacOS 10.7.5, with processor Intel Core i5, 1.7 GHz, MatLab r2013a, solver CPLEX [IBM11], YALMIP [Lö4] and MPT [KGB04].

Depending on the control technique adopted either D or Dss discretization schemes can be chosen.
The first goal of a decentralized or distributed AGC layer is to have performance in terms of η similar to CeMPC. Matching also the values of Φ can be seen as a secondary objective.
Alternative control schemes will be also ranked according to the degree of decentralization of the design process. Ideally, the controller of each area should be designed independently of the others and using information from a limited number of other areas. Decentralized design is important in PNS because if an area needs to be isolated or a new area is plugged into the network one would like to avoid the redesign the whole AGC layer and rather retune just a limited number of local controllers in order to guarantee asymptotic stability and constraints satisfaction for the whole network.

Appendix C

PnPMPC-toolbox for MatLab

The *PnPMPC-toolbox* is a GNU-licensed MatLab toolbox for the modeling of constrained LSSs described by LTI dynamics and for the software implementation of the PnPMPC schemes proposed in this thesis. The main goals of this toolbox are the following:

(i) to ease the modeling of large-scale constrained system;

(ii) to perform the design of PnPMPC regulators;

(iii) to perform the design of PnP LSEs.

There are many MatLab toolboxes and tools for modeling dynamical systems, but most of them have not been developed for achieving aim (i) above. Indeed, since a LSS is composed by several subsystems, one would like to easily perform the following operations: add and remove a subsystem, extract a subsystem from a whole system, extract the constraints of a subsystem, change the dynamics of a subsystem and add coupling among two subsystems. In the development of the *PnPMPC-toolbox* we provided methods in order to fulfill these and other requirements.

In the development of the *PnPMPC-toolbox* we used an object-oriented approach. This choice allows one to encapsulate in a single object all data characterizing a LSS. To this purpose we defined the *lss* (large-scale system) class. An *lss* object allows one to describe a constrained model, and, thanks to the sparsity of the matrices of the system, we reduced as much as possible the use of computational resources.

We also developed functions for the design of the distributed state estimator in Chapter 3 and for the design of PnPMPC regulators and LSEs proposed in the second part of the thesis.

Since most of the proposed decentralized and distributed algorithms require the computation of invariant sets, we provided functions to compute efficiently these sets implementing some of the procedures in [RKKM05] and [RB10]. Moreover, we developed a class to manage zonotopic sets.
We highlight that all simulations in this thesis, as well as the modeling of the LSSs, PnP design of controllers and state estimators, have been developed using the *PnPMPC-toolbox*.

The toolbox has been developed by S. Riverso, A. Battocchio and G. Ferrari-Trecate and is available at http://sisdin.unipv.it/pnpmpc/pnpmpc.php.

Bibliography

[ABB11] A. Alessio, D. Barcelli, and A. Bemporad, "Decentralized model predictive control of dynamically coupled linear systems," *Journal of Process Control*, vol. 21, no. 5, pp. 705–714, 2011.

[AM89] B. D. O. Anderson and J. B. Moore, *Optimal Control - Linear Quadratic Methods*. Englewood Cliffs, NJ, USA: Prentice Hall International, Inc., 1989.

[AR06] P. Alriksson and A. Rantzer, "Distributed Kalman Filtering Using Weighted Averaging," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, July 24-28, 2006.

[Bai66] F. N. Bailey, "The application of Lyapunov's second method to interconnected systems," *SIAM Journal on Control and Optimization*, vol. 3, no. 3, pp. 443–462, 1966.

[BDD06] T. Bellemans, B. De Schutter, and B. De Moor, "Model predictive control for ramp metering of motorway traffic: A case study," *Control Engineering Practice*, vol. 14, no. 7, pp. 757–767, 2006.

[BFPP11] F. Boem, R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "A distributed fault detection methodology for a class of large-scale uncertain input-output discrete-time nonlinear systems," in *Proceedings of the 50th IEEE Conference on Decision and Control, and the European Control Conference*, Orlando, FL, USA, December 12-15, 2011, pp. 897–902.

[BFS13] G. Betti, M. Farina, and R. Scattolini, "Decentralized predictive control for tracking constant references," in *Proceedings of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, December 10-13, 2013, p. Accepted.

[BFT04] A. Bemporad, C. Filippi, and F. D. Torrisi, "Inner and outer approximations of polytopes using boxes," *Computational Geometry*, vol. 27, no. 2, pp. 151–178, 2004.

[BK13] M. Babazadeh and H. Karimi, "A Robust Two-Degree-of-Freedom Control Strategy for an Islanded Microgrid," *IEEE Transactions on Power Delivery*, vol. 28, no. 3, pp. 1339–1347, 2013.

[BL88] L. Bakule and J. Lunze, "Decentralized design of feedback control for large-scale systems," *Kybernetika*, vol. 24, no. 8, pp. 3–96, 1988.

[Bla91] F. Blanchini, "Ultimate Boundedness Control for Uncertain Discrete-Time Systems via Set-Induced Lyapunov Functions," in *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, England, December 11-13, 1991, pp. 1755–1760.

[Bla99] ——, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[BM99] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–428, 1999.

[BM08] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Berlin, Germany: Springer, Birkhäuser Mathematics, Systems & Control: Foundations & Applications, 2008.

[Bor03] F. Borrelli, *Constrained optimal control of linear and hybrid systems*. Berlin, Germany: Springer, Lecture Notes in Control and Information Sciences vol. 290, 2003.

[BP70] F. M. Brasch Jr and J. B. Pearson, "Pole placement using dynamic compensators," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 34–43, 1970.

[BT97] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* Belmont, Massachusetts, USA: Athena Scientific, Optimization and Neural Computation Series, 1997.

[BTS13] J. Bendtsen, K. Trangbaek, and J. Stoustrup, "Plug-and-Play Control — Modifying Control Systems Online," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 79–93, 2013.

[BZ13] S. Bolognani and S. Zampieri, "A distributed control strategy for reactive power compensation in smart microgrids," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2818–2833, 2013.

[CCSZ08] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman Filtering Based on Consensus Strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, 2008.

[CJKT02] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[CSML13] P. D. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers and Chemical Engineering*, vol. 51, pp. 21–41, 2013.

[Dag97] C. F. Daganzo, *Fundamentals of Transportation and Traffic Operations.* Oxford, UK: Pergamon Press, 1997.

[DRW07] S. Dashkovskiy, B. S. Rüffer, and F. R. Wirth, "An ISS small gain theorem for general networks," *Mathematics of Control, Signals, and Systems*, vol. 19, no. 2, pp. 93–122, 2007.

[Dun07] W. B. Dunbar, "Distributed Receding Horizon Control of Dynamically Coupled Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1249–1263, 2007.

[EDI12] A. H. Etemadi, E. J. Davison, and R. Iravani, "A Decentralized Robust Control Strategy for Multi-DER Microgrids - Part I: Fundamental Concepts," *IEEE Transactions on Power Delivery*, vol. 27, no. 4, pp. 1843–1853, 2012.

[FBS13] M. Farina, G. Betti, and R. Scattolini, “A solution to the tracking problem using distributed predictive control,” in *Proceedings of the 12th European Control Conference*, Zurich, Switzerland, July 17-19, 2013, pp. 4347–4352.

[FCS13] M. Farina, P. Colaneri, and R. Scattolini, “Block-wise discretization accounting for structural constraints,” *Automatica*, vol. 49, no. 11, pp. 3411–3417, 2013.

[FFTS10a] M. Farina, G. Ferrari-Trecate, and R. Scattolini, “Distributed Moving Horizon Estimation for Linear Constrained Systems,” *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2462–2475, 2010.

[FFTS10b] ——, “Moving-horizon partition-based state estimation of large-scale systems,” *Automatica*, vol. 46, no. 5, pp. 910–918, 2010.

[FLAC13] A. Ferramosca, D. Limon, I. Alvarado, and E. F. Camacho, “Cooperative distributed MPC for tracking,” *Automatica*, vol. 49, no. 4, pp. 906–914, 2013.

[FR00] L. Farina and S. Rinaldi, *Positive Linear Systems*. New York. NY, USA: John Wiley & Sons, 2000.

[FS11a] M. Farina and R. Scattolini, “An output feedback distributed predictive control algorithm,” in *Proceedings of the 50th IEEE Conference on Decision and Control, and the European Control Conference*, Orlando, FL, USA, December 12-15, 2011, pp. 8139–8144.

[FS11b] ——, “Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems,” Rapporto interno 2011.13, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy, Tech. Rep., 2011.

[FS12] ——, “Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems,” *Automatica*, vol. 48, no. 6, pp. 1088–1096, 2012.

[Gal95] T. Gal, *Postoptimal Analyses, Parametric Programming and Related Topics*, 2nd ed. Berlin, Germany: de Gruyter, 1995.

[GDK+13] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.

[Gio11] L. Giovanini, "Game approach to distributed model predictive control," *IET Control Theory and Applications*, vol. 5, no. 15, p. 1729, 2011.

[GM86] P. Grosdidier and M. Morari, "Interaction measures for systems under decentralized control," *Automatica*, vol. 22, no. 3, pp. 309–319, 1986.

[GR06] R. D. Gudi and J. B. Rawlings, "Identification for decentralized model predictive control," *AIChE Journal*, vol. 52, no. 6, pp. 2198–2210, 2006.

[GR10] P. Giselsson and A. Rantzer, "Distributed Model Predictive Control with Suboptimality and Stability Guarantees," in *Proceedings of the 49th IEEE Conference on Decision and Control*, Hilton Atlanta Hotel, Atlanta, GA, USA, December 15-17, 2010, pp. 7272–7277.

[GT91] E. G. Gilbert and K. T. Tan, "Linear Systems with State and Control Constraints: The Theory and Application of Maximal Output Admissible Sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[Gv73] L. T. Grujic and D. D. Šiljak, "Asymptotic stability and instability of large-scale systems," *IEEE Transactions on Automatic Control*, vol. 18, no. 6, pp. 636–645, 1973.

[HLJ10] R. M. Hermans, M. Lazar, and A. Jokić, "Almost decentralized Lyapunov-based nonlinear model predictive control," in *Proceedings of American Control Conference 2010*, Baltimore, MD, USA, June 30 - July 02, 2010, pp. 3932–3938.

[Hyc10] Hycon2, "Highly-complex and networked control systems (HYCON2 Network of excellence)," 2010. [Online]. Available: http://www.hycon2.eu

[IBM11] IBM, "IBM ILOG CPLEX Optimization Studio," 2011.

[KBB06] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105–2115, 2006.

[KG98] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical Problems in Engineering*, vol. 4, no. 4, pp. 317–363, 1998.

[KGB04] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004. [Online]. Available: http://control.ee.ethz.ch/~mpt/

[Kha02] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[KM08] U. A. Khan and J. M. F. Moura, "Distributing the Kalman Filter for Large-Scale Systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4919–4935, 2008.

[Kon13] A. Konstantellos, "A short overview of control in European R&D programmes (1983–2013): From local loop designs, through networked and coordinated control, to stochastic, large scale and real time optimization systems," *European Journal of Control*, vol. 19, no. 5, pp. 351–357, 2013.

[KT08] M. Kamgarpour and C. Tomlin, "Convergence properties of a decentralized Kalman filter," *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 3205–3210, 2008.

[LÖ4] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proceedings of IEEE Symposium on Computer Aided Control Systems Design*, Taipei, Taiwan, September 2-4, 2004, pp. 284–289.

[LA06] J. Lavaei and A. G. Aghdam, "A necessary and sufficient condition for the existence of a LTI stabilizing decentralized overlapping controller," in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December 13-15, 2006, pp. 6179–6186.

[LA08] ——, "Control of continuous-time LTI systems by means of structurally constrained controllers," *Automatica*, vol. 44, no. 1, pp. 141–148, 2008.

[LCRM04] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[LMC09] J. Liu, D. Muñoz de la Peña, and P. D. Christofides, "Distributed model predictive control of nonlinear process systems," *AIChE Journal*, vol. 55, no. 5, pp. 1171–1184, 2009.

[Lun92] J. Lunze, *Feedback control of large scale systems.* Upper Saddle River, NJ, USA: Prentice Hall, Systems and Control Engineering, 1992.

[Mac02] J. M. Maciejowski, *Predictive Control: with constraints.* Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[Mar98] S. Markov, "On the Algebraic Properties of Convex Bodies," *Pliska Studia Mathematica Bulgarica*, vol. 12, pp. 119–132, 1998.

[MBH+12] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model Predictive Control for the Operation of Building Cooling Systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 796–803, 2012.

[Mey00] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra.* Philadelphia, Pennsylvania, USA: Society for Industrial and Applied Mathematics, 2000.

[ML99] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers and Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, May 1999.

[MRA12] M. A. Müller, M. Reble, and F. Allgöwer, "Cooperative control of dynamically decoupled systems via distributed model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 22, pp. 1376–1397, 2012.

[MRFA06] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer, "Robust output feedback model predictive control of constrained linear systems," *Automatica*, vol. 42, no. 7, pp. 1217–1222, Jul. 2006.

[MRRS00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[MS06] L. Magni and R. Scattolini, "Stabilizing decentralized model predictive control of nonlinear systems," *Automatica*, vol. 42, no. 7, pp. 1231–1236, 2006.

[MS07] O. Mason and R. Shorten, "On Linear Copositive Lyapunov Functions and the Stability of Switched Positive Linear Systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1346–1349, 2007.

[MSR05] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[Mut98] A. G. O. Mutambara, *Decentralized estimation and control for multisensor systems*. Boca Raton, FL: CRC Press, 1998.

[NS08] I. Necoara and J. A. K. Suykens, "Application of a Smoothing Technique to Decomposition in Convex Optimization," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2674–2679, 2008.

[OPJ+12] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari, "Use of model predictive control and weather forecasts for energy efficient building climate control," *Energy and Buildings*, vol. 45, pp. 15–27, 2012.

[PK02] M. Papageorgiou and A. Kotsialos, "Freeway ramp metering: an overview," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 271–281, 2002.

[PR01] G. Pannocchia and J. B. Rawlings, "The Velocity Algorithm LQR: a survey," Department of Chemical Engineering, University of Wisconsin-Madison, Number 2001-01, Madison, WI, USA, Tech. Rep., 2001.

[Rak05] S. V. Raković, "Robust Control of Constrained Discrete Time Systems: Characterization and Implementation," Ph.D. dissertation, Imperial College London, University of London, 2005.

[Raw00] J. B. Rawlings, “Tutorial Overview of Model Predictive Control,” *IEEE Control Systems Magazine*, vol. 20, no. 3, pp. 38–52, 2000.

[RB09] S. V. Raković and M. Baric, “Local Control Lyapunov Functions for Constrained Linear Discrete-Time Systems: The Minkowski Algebra Approach,” *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2686–2692, Nov. 2009.

[RB10] ——, “Parameterized Robust Control Invariant Sets for Linear Systems: Theoretical Advances and Computational Remarks,” *IEEE Transactions on Automatic Control*, vol. 55, no. 7, pp. 1599–1614, 2010.

[RBFT12] S. Riverso, A. Battocchio, and G. Ferrari-Trecate, “PnPMPC: a toolbox for MatLab,” 2012. [Online]. Available: http://sisdin.unipv.it/pnpmpc/pnpmpc.php

[RFFT12a] S. Riverso, M. Farina, and G. Ferrari-Trecate, “Plug-and-Play Decentralized Model Predictive Control,” in *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, Hawaii, USA December 10-13, 2012, pp. 4193–4198.

[RFFT12b] ——, “Plug-and-Play Decentralized Model Predictive Control,” Dipartimento di Ingegneria Industriale e dell’Informazione, Universita’ degli Studi di Pavia, Pavia, Italy, Tech. Rep., 2012. [Online]. Available: arXiv:1302.0226

[RFFT12c] ——, “Plug-and-Play Model Predictive Control based on robust control invariant sets,” Dipartimento di Ingegneria Industriale e dell’Informazione, Universita’ degli Studi di Pavia, Pavia, Italy, Tech. Rep., 2012. [Online]. Available: arXiv:1210.6927

[RFFT13a] ——, “Design of plug-and-play model predictive control: an approach based on linear programming,” in *Proceedings of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, December 10-13, 2013, pp. 6530–6535.

[RFFT13b] ——, “Plug-and-Play Decentralized Model Predictive Control for Linear Systems,” *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2608–2614, 2013.

[RFFT13c] ——, “Plug-and-Play Model Predictive Control based on robust control invariant sets,” *Automatica*, p. Submitted, 2013.

[RFSFT13a] S. Riverso, M. Farina, R. Scattolini, and G. Ferrari-Trecate, “Plug-and-play distributed state estimation for linear systems,” in *Proceedings of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, December 10-13, 2013, pp. 4889–4894.

[RFSFT13b] ——, “Plug-and-play distributed state estimation for linear systems,” Dipartimento di Ingegneria Industriale e dell’Informazione, Universita’ degli Studi di Pavia, Pavia, Italy, Tech. Rep., 2013. [Online]. Available: arXiv:1309.2002

[RFT12a] S. Riverso and G. Ferrari-Trecate, “Hycon2 Benchmark: Power Network System,” Dipartimento di Informatica e Sistemistica, Universita’ degli Studi di Pavia, Pavia, Italy, Tech. Rep., 2012. [Online]. Available: arXiv:1207.2000

[RFT12b] ——, “Tube-based distributed control of linear constrained systems,” *Automatica*, vol. 48, no. 11, pp. 2860–2865, 2012.

[RFT12c] ——, “Tube-based distributed control of linear constrained systems,” RIDIS 156/12, Dipartimento di Informatica e Sistemistica, Universita’ degli Studi di Pavia, Pavia, Italy, Tech. Rep., 2012. [Online]. Available: http://sisdin.unipv.it/lab/personale/pers_hp/ferrari/publication_details/RFT12.php

[RFT13] ——, “Plug-and-Play distributed model predictive control with coupling attenuation,” *Optimal Control Applications and Methods*, p. Submitted, 2013.

[RKF10] S. V. Raković, B. Kern, and R. Findeisen, “Practical Set Invariance for Decentralized Discrete Time Systems,” in *Proceedings of the 49th IEEE Conference on Decision and Control*, Hilton Atlanta Hotel, Atlanta, GA, USA, December 15-17, 2010, pp. 3283–3288.

[RKF11] ——, “Practical Robust Positive Invariance for Large – Scale Discrete Time Systems,” in *Proceedings of the 16th IFAC World Congress*, Milano, Italy, August 28 - September 2, 2011, pp. 6425–6430.

[RKKM05] S. V. Raković, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, "Invariant approximations of the minimal robust positively invariant set," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 406–410, 2005.

[RM05] S. V. Raković and D. Q. Mayne, "A simple tube controller for efficient robust model predictive control of constrained linear discrete time systems subject to bounded disturbances," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 4-8, 2005, pp. 241–246.

[RM09] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI, USA: Nob Hill Pub., 2009.

[RMS07] D. M. Raimondo, L. Magni, and R. Scattolini, "Decentralized MPC of nonlinear systems: An input-to-state stability approach," *International Journal of Robust and Nonlinear Control*, vol. 17, pp. 1651–1667, 2007.

[RRFT13a] S. Riverso, D. Rubini, and G. Ferrari-Trecate, "Distributed bounded-error state estimation based on practical robust positive invariance," *Systems & Control Letters*, p. Submitted, 2013.

[RRFT13b] ——, "Distributed bounded-error state estimation for partitioned systems based on practical robust positive invariance," in *Proceedings of the 12th European Control Conference*, Zurich, Switzerland, July 17-19, 2013, pp. 2633–2638.

[RRFT13c] ——, "Distributed bounded-error state estimation for partitioned systems based on practical robust positive invariance," Dipartimento di Informatica e Sistemistica, Universita' degli Studi di Pavia, Pavia, Italy, Tech. Rep., 2013. [Online]. Available: arXiv:1311.4306

[RRS+12] D. M. Raimondo, S. Riverso, S. Summers, C. N. Jones, J. Lygeros, and M. Morari, "A set theoretic method for verifying feasibility of a fast explicit nonlinear Model Predictive Controller," in *Distributed Decision Making and Control*, R. Johansson and A. Rantzer, Eds. Springer, Lecture Notes in Control and Information Sciences vol. 417, 2012, ch. 13, pp. 289–311.

[RS08] J. B. Rawlings and B. T. Stewart, "Coordinating multiple optimization-based controllers: New opportunities and challenges," *Journal of Process Control*, vol. 18, no. 9, pp. 839–845, 2008.

[RTMA06] S. V. Raković, A. R. Teel, D. Q. Mayne, and A. Astolfi, "Simple Robust Control Invariant Tubes for Some Classes of Nonlinear Discrete Time Systems," in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December 13-15, 2006, pp. 6397–6402.

[Saa02] H. Saadat, *Power System Analysis*, 2nd ed. New York. NY, USA: McGraw-Hill Series in Electrical and Computer Engineering, 2002.

[Sca09] R. Scattolini, "Architectures for distributed and hierarchical Model Predictive Control – A review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[Sch93] R. Schneider, *Convex bodies: the Brunn-Minkowski theory*. Cambridge University: Cambridge University Press, 1993.

[SGML08] G. Sánchez, L. Giovanini, M. Murillo, and A. Limache, "Distributed Model Predictive Control Based on Dynamic Games," in *Advanced Model Predictive Control*, T. Zheng, Ed. Rijeka, Croatia: Intech Open, Electrical and Electronic Engineering, Control Systems, 2008, pp. 65–90.

[SP96] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*. New York, NY, USA: John Wiley & Sons, 1996.

[SP11] T. Samad and T. Parisini, "Systems of Systems," in *The Impact of Control Technology*, T. Samad and A. M. Annaswamy, Eds. IEEE Control Systems Society, 2011, pp. 175–183. [Online]. Available: ieeecss.org/general/impact-control-technology

[SSS09a] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic, "Consensus based overlapping decentralized estimation with missing observations and communication faults," *Automatica*, vol. 45, pp. 1397–1406, 2009.

[SSS09b] ——, "Consensus based overlapping decentralized estimator," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 410–415, 2009.

[Sta04] S. S. Stankovic, "Inclusion Principle for Discrete-Time Time-Varying Systems," *Dynamics of Continuous, Discrete and Impulsive Systems*, vol. 11, no. Series A: Mathematical Analysis, pp. 321–338, 2004.

[Sto09] J. Stoustrup, "Plug & Play Control: Control Technology towards new Challenges," in *Proceedings of the 10th European Control Conference*, Budapest, Hungary, August 23-26, 2009, pp. 1668–1683.

[SVAS78] N. R. Sandell Jr, P. Varaiya, M. Athans, and M. G. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 108–128, 1978.

[SVR+10] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, "Cooperative distributed model predictive control," *Systems & Control Letters*, vol. 59, no. 8, pp. 460–469, 2010.

[SWR11] B. T. Stewart, S. J. Wright, and J. B. Rawlings, "Cooperative distributed model predictive control for nonlinear systems," *Journal of Process Control*, vol. 21, no. 5, pp. 698–704, 2011.

[TR10] P. Trodden and A. Richards, "Distributed model predictive control of linear systems with persistent disturbances," *International Journal of Control*, vol. 83, no. 8, pp. 1653–1663, 2010.

[VC94] L. N. Vicente and P. H. Calamai, "Bilevel and multilevel programming: A bibliography review," *Journal of Global Optimization*, vol. 5, no. 3, pp. 291–306, 1994.

[VD03] R. Vadigepalli and F. J. Doyle III, "A distributed state estimation and control algorithm for plantwide processes," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 1, pp. 119–127, 2003.

[Wan08] L. Wang, "A Tutorial on Model Predictive Control: Using a Linear Velocity-Form Model," *Developments in Chemical Engineering and Mineral Processing*, vol. 12, no. 5-6, pp. 573–614, 2008.

[WD73] S. Wang and E. J. Davison, "On the stabilization of decentralized control systems," *IEEE Transactions on Automatic Control*, vol. 18, no. 5, pp. 473–478, 1973.

[Wis71] D. A. Wismer, *Optimization methods for large-scale systems with applications*. New York. NY, USA: McGraw-Hill, 1971.

[ZIF01] G. Zhai, M. Ikeda, and Y. Fujisaki, "Decentralized H_∞ controller design: a matrix inequality approach using a homotopy method," *Automatica*, vol. 37, no. 4, pp. 565–572, 2001.

[ZPR+13] M. N. Zeilinger, Y. Pu, S. Riverso, G. Ferrari-Trecate, and C. N. Jones, "Plug and Play Distributed Model Predictive Control based on Distributed Invariance and Optimization," in *Proceedings of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, December 10-13, 2013, pp. 5770–5776.

[Zv94] A. I. Zečević and D. D. Šiljak, "Balanced decompositions of sparse systems for multilevel parallel processing," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, no. 3, pp. 220–233, 1994.

[Zv10] ——, *Control of complex systems: Structural constraints and uncertainty*. Berlin, Germany: Springer, Communications and Control Engineering, 2010.