European Journal of Control xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Control

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

Revisiting a fixed-point hierarchical control design for cryogenic refrigerators under constraints, nonlinearities and real-time considerations

Xuan-Huy Pham[a,b,*], Mazen Alamir[b,*], François Bonne[a], Patrick Bonnay[a]

[a] Univ. Grenoble Alpes, IRIG-DSBT, Grenoble F-38000, France
[b] CNRS, Univ. Grenoble Alpes, Gipsa-lab, Grenoble F-38000, France

ARTICLE INFO

Article history:
Received 5 March 2021
Revised 8 September 2021
Accepted 15 September 2021
Available online xxx

Recommended by Prof. T Parisini

Keywords:
Hierarchical MPC
Cryogenic refrigerator
Fixed-point iteration
Real-time
Modular control design
Constrained control

ABSTRACT

This paper presents an extension of a recently proposed hierarchical control framework applied to linearization-based unconstrained control of a cryogenic refrigerator. The extension concerns the validation of the framework in the presence of both nonlinear models and constraints. It is also shown that real-time handling of these features requires a specific complexity reduction technique. This technique performs the distribution of the optimization process over cyclically changed decision variables aiming at limiting the number of iterations per updating period. Numerical simulations are proposed in order to show the impact of the parameters choices and to assess the real-time implementability of the proposed framework.

1. Introduction

Cryogenic refrigerators are necessary to cool down the superconducting devices in many applications including nuclear fusion reactors as well as particle accelerators [6,9]. In fact, theses refrigerators are composed of many subsystems that span over large areas. For many reasons, it is common to make changes in these subsystems (PID re-tuning, valve changing) without any consistent assessment of how these changes might affect the overall system behavior. Several centralized architectures have been proposed and experimented on a experimental station at CEA[1]/IRIG[2]/ DSBT.[3] However, these control methods appear to be non scalable and even inappropriate from an operational point of view. This is in particular true when some actuators need to be changed or some sub-systems need to be turned off/on. Hence, the need for scalable and modular design that smoothly accommodates for changes in local sub-systems level triggered interest in distributed/hierarchical design investigation leading to the framework proposed in Alamir et al. [2] which the present work extends.

In the last decades, non-centralized control architectures [4,7,8,10,12,13] attracted attention within the control community due to their ability to break the problem into smaller tractable ones inducing modularity, security, and easiness of implementation. These approaches can be split into two categories: Distributed and Hierarchical.

Distributed architectures involve subsystems exchanging information with adjacent neighbors. In contrast, in hierarchical frameworks, the subsystems communicate only with a coordinator that uses the information received from the subsystems to compute new evaluation of the coupling signals that are then sent back to the subsystems in order to enhance the overall quality of the system behavior. The survey book [10] listed more than 35 different approaches to such frameworks.

This paper is an extension of the framework proposed in Alamir et al. [2] where the reader can find the precise positioning concerning the landscape of available approaches as well as the first application to a cryogenic refrigerator. Nevertheless, the following items recall the main points that enable to position the framework studied in this paper and [2] in the landscape of related contributions:

* Corresponding authors at: Univ. Grenoble Alpes, IRIG-DSBT, F-38000 Grenoble, France.
E-mail addresses: xuanHuy.pham@cea.fr (X.-H. Pham), mazen.alamir@grenoble-inp.fr (M. Alamir).
[1] Commissariat à l'Énergie Atomique.
[2] Institut de Recherche Inter-disciplinaire de Grenoble.
[3] Department of Low Temperature Systems.

https://doi.org/10.1016/j.ejcon.2021.09.009
0947-3580/

Nomenclature

MPC	model predictive control
CasADi	open-source software tool for numerical optimization in general and optimal control
PID	proportional-integral-derivative controller
r^d	desired set-point defined by the operator
r	auxiliary set-point used to evaluate $J_c(r)$
r^{opt}	optimal set-point computed by solving the problem $\text{argmin}_{r\in\mathcal{R}} J_c(r)$
r_c^{opt}	optimal set-point computed by solving the problem $\text{argmin}_{r\in\mathscr{P}} \hat{J}_c(r)$
z	decision variable with $n_z < n_r$
$r_c^\star$	candidate auxiliary set-point computed by $z^\star$ and $r_c^\star$ of last iteration
$r^\star$	sub-optimal set-point that the coordinator sends to the subsystems to calculate the final control

- ✓ Many studies have focused on the optimality and / or the fulfillment of constraints by assuming the closed-loop behavior to be stable [4,7,8]. This strong assumption is not required for the fixed-point iterations proposed in Alamir et al. [2] and which is revisited in the present paper. Although this method is not proven to be stable when constraints and nonlinearities are present, it enforces at least local closed-loop stability. The examples provided in this paper partially assess its effectiveness even when nonlinearities and constraints are present.
- ✓ In several studies, the same assumption is re-introduced by stipulating a weak coupling between subsystems. More precisely, it is assumed that the coupling signals do not lead to the destabilizing effect.
- ✓ In many works, the structure of the coupling between the subsystems is constrained. For instance, in Stewart et al. [13], it is assumed that there is only a coupling through control input actions. In [11], a hierarchical structure is proposed where the higher level is coupled with the control input provided by the lower level. Such structural assumptions on the coupling are not used here since they might highly compromise the scalability and the scope of application. Instead, in order to overcome the challenge of strong coupling in the scheme proposed in Alamir et al. [2], the coordinator takes appropriate explicit actions in order to stabilize the iterations with local subsystems.

The present work extends the results of [2] in three directions:

- ✓ This paper proposes an extended set of numerical experiments in which the feedback used in the subsystems level is a constrained and/or Nonlinear Model Predictive Control **NMPC**. More precisely, constrained NMPC is used for the Brayton cycle while constrained linear MPC is used for the Joule–Thomson cycle (see Fig. 1(a)).
- ✓ Due to the non negligible computation time associated with constrained nonlinear framework which has to be performed several times in a single sampling period (due to the fixed-point iterations), it is shown that standard use of the state-of-the-art numerical software [3] leads to a non-implementable scheme. The paper proposes a complexity reduction technique based on the ideas proposed in Alamir [1] in order to derive a real-time implementable version of the framework.
- ✓ As far as the comparison with a decentralized framework is concerned, while in Alamir et al. [2] the local controllers (at the subsystems level) assume nominal values of the coupling signals, in this paper, local observers are implemented which highly improve the performance of the decentralized setting rendering the comparison fairer and more challenging for the hierarchical settings.

The paper is organized as follows: Section 2 describes the system and states the control problem. Section 3 recalls the hierarchical framework and its formulation as stated in Alamir et al. [2] in a more general nonlinear and constrained setting. Section 4 proposes a method to distribute the optimization over time towards achieving real-time implementability. The simulation results and analysis are given in Section 5 while Section 6 concludes the paper and gives hints for future investigations.

2. Problem description

2.1. Brief description of the cryogenic refrigerator

This section briefly describes the cryogenic refrigerator that is studied in this paper.[4] This plant is used for the purpose of conducting experiments (testing cryogenic components, study of super-fluid helium, etc) aiming at improving the design and operation of cryogenic refrigerators. Fig. 1(b) shows the ideal underlying Claude thermodynamic cycle in the (entropy -temperature) coordinates.

Fig. 1 (a) shows the decomposition of the overall process into four interconnected subsystems. More precisely, the system can be viewed as the interconnection of four subsystems: Joule–Thomson (J-T) cycle (S_1), Brayton cycle (S_2), LN2 precooling (S_3), and warm compression station (WCS) (S_4).

A cryogenic refrigerator provides cooling power by using a cryogenic turbine (the one following the valve CV_{156} in S_2 of Fig. 1(a)) to extract thermal energy from the fluid and by exchanging heat power through a series of heat exchangers (represented by NEF_x in Fig. 1(a)). The main objective is to reject the disturbance represented by the heat power induced by the operation of the experimental facility. In our experimental facility, this disturbance is represented by the heat source denoted by NCR_{22} (see S_1 on Fig. 1(a)). A compressor is used to close the thermodynamic cycle in the so-called warm compression station (S_4). The role of this subsystem is to maintain the pressures (denoted by H_p (high pressure) and B_p (low pressure) in Fig. 1(a)) within some tightly imposed bounds which correspond to an optimal operation regime of the overall system.

Concerning the dynamic model of the cryogenic refrigerator, a library has been developed in Bonne et al. [5] in which nonlinear models of all the cryogenic components necessary for system modeling (valves, heat exchangers, Helium bath, turbines, compressor) are implemented and connected before a linearization is performed around the operational point of the refrigerator. All the details concerning this library can be found in Bonne et al. [5] and the references therein.

This paper extends the work of [2] (by addressing the constrained nonlinear setting as well as the induced real-time implementation issues) while keeping the framework in which the warm area (subsystem S_3 and S_4) is assumed to be properly controlled. Therefore, this paper focuses on the control of the so-called cold zone, including subsystems S_1 and S_2. The corresponding control problem is described in the next subsection.

In order to clearly state the control problem, the inputs, outputs and constraints are first defined in the next section.

2.1.1. The manipulated inputs

There are three control inputs which are CV_{155}, NCR_{22} belonging to Joule–Thomson cycle (S_1) and ΔP_{156} which is a part of the Brayton cycle (S_2). These actuators are defined below:

[4] The system is located at CEA/IRIG/DSBT, Grenoble.

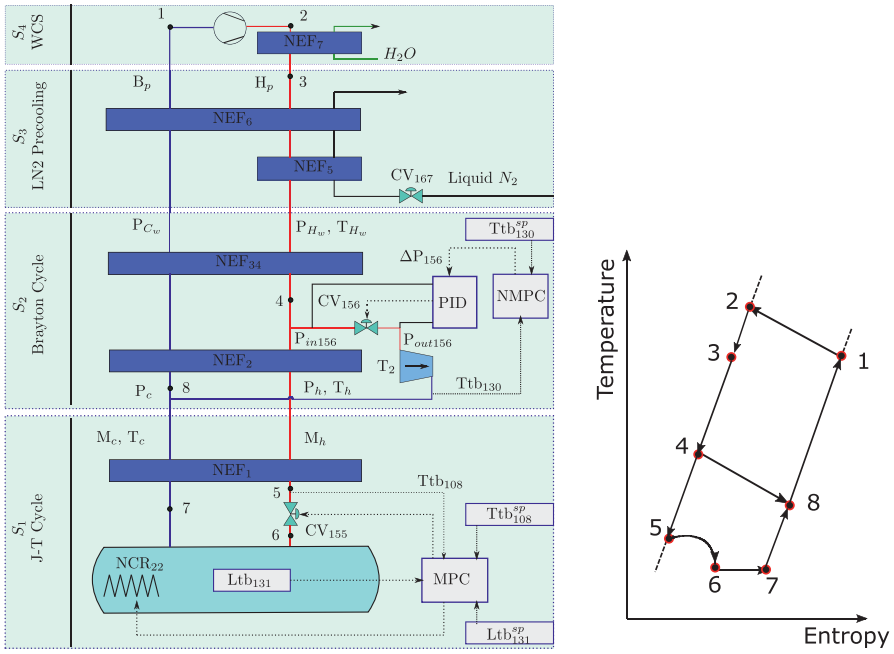

Fig. 1. Block diagram of the 400@1.8K experimental refrigerator (in the 400W@4.4K configuration) at CEA/IRIG/DSBT [2]. (a) shows the system can be decomposed into four parts: the Warm compression station (WCS), Precooling stage, Brayton cycle and Joule–Thomson cycle. (b) illustrates the ideal entropy -temperature diagram, which is implemented by the 400@1.8K refrigerator.

1. $CV_{155} \in [20\%, 90\%]$: This valve is situated at the inlet of the helium bath.
2. NCR_{22}: This actuator is located inside the helium bath (S_1). Note that this heating resistance is also used to simulate the heat pulses coming from the operation of physical experiments served by refrigerators. More precisely, the total power of the resistance is split into two components $NCR_{22} = NCR^{(a)}_{22} + NCR^{(w)}_{22}$, the first component is the control input inside S_1 and the second one is used as a disturbance signal. The value of $NCR^{(a)}_{22}$ is in the range of [0,100] W.
3. $\Delta P_{156} \in [0, 10]$ bar: The pressure drop between the inlet pressure and outlet pressure of the valve CV_{156}. It should be noted that the valve CV_{156} is used to control the pressure drop ΔP_{156} between its inlet and outlet pressure. To do this, the local NMPC of S_2 computes and sends an appropriate value of the pressure drop ΔP_{156} to the PID controller, which acts on the opening position of the valve CV_{156} (Fig. 1(a)).

2.1.2. The regulated outputs

There are three regulated outputs (see Fig. 1(a). for the notation) :

1. Ltb_{131}: The helium liquid level (%) that must be controlled to ensure that some thermal charges deposited inside the phase separator (e.g. used to cool super-critical helium at liquid helium temperature to be ready for the final customer) are immersed with liquid helium. The set-point is chosen by the operator. In the usual operation, it is set at $Ltb^{sp}_{131} = 60.5\%$.
2. Ttb_{108}: The temperature at the inlet of the J-T valve must be tightly controlled in order to ensure the efficiency of the liquefaction of the helium.
3. Ttb_{130}: Since the cryogenic turbine is a sensitive component, the temperature at its outlet must be regulated to avoid the risk of solid droplet forming at the outlet, potentially destructive for the turbine.

Table 1
Steady state values of the inputs and outputs. P_{H_w} and P_{C_w} represent respectively the high and the low pressures controlled by the warm compression station.

Variable	Value	Unit	Variable	Value	Unit
CV^0_{155}	55	%	Ttb^0_{130}	12.31	K
ΔP^0_{156}	1.5	bar	Ttb^0_{108}	5.37	K
NCR^0_{22}	310	W	Ltb^0_{131}	60.5	%
P_{H_w}	16	bar	P_{C_w}	1.05	bar

2.1.3. The measured outputs

The design of the local controllers needs the states of the subsystems to be reconstructed. This is done through dynamic observers. In order to synthesize the observers for these two subsystems, the measured outputs need to be specified. The observer of J-T cycle uses the measurement of Ltb_{131} and Ttb_{108} while the observer of the Brayton cycle uses the measurements of Ttb_{130}.

2.1.4. The nominal operation point

The nominal operation point is the usual operating point of the refrigerator. This operation point is determined by the steady values of the inputs, outputs and heating loads described above, namely:

$$CV^0_{155}, \quad NCR^{(a,0)}_{22}, \quad \Delta P^0_{156}, \quad Ltb^0_{131}, \quad Ttb^0_{130}, \quad NCR^{(w,0)}_{22} \tag{1}$$

The nominal values are shown in Table 1. These values are used to define the following deviation variables:

$$u_1 := \left[CV_{155} - CV^0_{155} \; ; \; NCR^{(a)}_{22} - NCR^{(a,0)}_{22}\right] \tag{2}$$

$$u_2 := \Delta P_{156} - \Delta P^0_{156} \tag{3}$$

$$y_1 := \left[Ltb_{131} - Ltb^0_{131} \; ; \; Ttb_{108} - Ttb^0_{108}\right] \tag{4}$$

$$y_2 := \mathrm{Ttb}_{130} - \mathrm{Ttb}^0_{130} \tag{5}$$

$$w := \mathrm{NCR}^{(w)}_{22} - \mathrm{NCR}^{(w,0)}_{22}. \tag{6}$$

In the sequel, the following condensed notation is used to refer to the nominal values of the controls and outputs to be regulated or constrained for the two subsystems involved in the study:

$$U^0_1 := \begin{bmatrix} \mathrm{CV}^0_{155} \\ \mathrm{NCR}^{(a,0)}_{22} \end{bmatrix}, \quad U^0_2 := \Delta \mathrm{P}^0_{156}, \tag{7}$$

$$Y^0_1 := \begin{bmatrix} \mathrm{Ltb}^0_{131} \\ \mathrm{Ttb}^0_{108} \end{bmatrix}, \quad Y^0_2 = \mathrm{Ttb}^0_{130} \tag{8}$$

Note that the operation point of the heating power is the sum of $\mathrm{NCR}^{(a,0)}_{22}$ and $\mathrm{NCR}^{(w,0)}_{22}$, namely:

$$\mathrm{NCR}^{(0)}_{22} = \mathrm{NCR}^{(a,0)}_{22} + \mathrm{NCR}^{(w,0)}_{22} \tag{9}$$

It is important to notice in order to keep the same operation point $\mathrm{NCR}^{(0)}_{22}$, higher values of pre-heating power $\mathrm{NCR}^{(a,0)}_{22}$ induce lower values of $\mathrm{NCR}^{(w,0)}_{22}$ which leads to the reduction of the amount of real heat disturbance power that can be rejected. This is the reason why $\mathrm{NCR}^{(a,0)}_{22}$ is chosen at the minimal value, which it can adopt in real-life configurations with the constraint ($\mathrm{NCR}^{(a)}_{22} > 0$). Table 1 shows the nominal values of the inputs and outputs described above.

In what follows, the bold-faced notation $\boldsymbol{p}$ denotes the profile of a vector variable p over a prediction horizon of length N. More precisely:

$$\boldsymbol{p} = [p^T(k), \ldots, p^T(k+N-1)]^T \in \mathbb{R}^{N \cdot n_p}$$

where n_p is the dimension of p.

2.1.5. The dynamic model

Recall that all the details regarding the physical models are given in Bonne et al. [5] and references therein. Only the formal and compact representation is given here in order to explain the hierarchical control framework. Using the sub-index $s \in \{1, 2\}$ to refer to the subsystem S_s, the nonlinear model governing the two subsystems defined, in terms of the deviation variable mentioned above, takes the following form:

$$\begin{aligned} x^+_s &= f_s(x_s, u_s, w_s, v_s) \\ y_s &= h_s(x_s, u_s, v_s) \\ v_{s'} &= g_{s'}(x_s, u_s, v_s) \end{aligned} \tag{10}$$

with $s' \neq s, \quad s, s' \in \{1, 2\}$. Note that the model of subsystem S_1 is fully linearized, namely $f_1(\cdot)$, $g_1(\cdot)$ and $h_1(\cdot)$ are linear functions which is compatible with the linear nature of underlying phenomenon. On the contrary, a nonlinear model is necessary to describe the dynamics of S_2. The notations x_s, y_s, v_s and w_s are defined as follows:

- $x_1 \in \mathbb{R}^{20}$ and $x_2 \in \mathbb{R}^{20}$ express the deviations from the steady state values of subsystem S_1 and S_2 respectively.
- $y_1 \in \mathbb{R}^2$ and $y_2 \in \mathbb{R}$ express the deviations of the regulated outputs of each subsystem.
- $v_1 \in \mathbb{R}^3$ and $v_2 \in \mathbb{R}^3$ are coupling signals by which the two systems interact dynamically. Indeed as indicated in the last equation of (10), v_1 depends on (x_2, u_2, v_2) while v_2 depends on (x_1, u_1, v_1). More precisely, as illustrated in Fig. 1(a), the coupling variable v_1 is given by

$$v_1 := (\mathrm{P}_h, \mathrm{T}_h, \mathrm{P}_c)^T \in \mathbb{R}^3 \tag{11}$$

where P_h and T_h are the pressure and temperature (deviations) at the downstream inlet of the heat exchanger NEF_1, whereas P_c stands for the pressure at the upstream outlet of the same exchanger. Similarly, the coupling signal v_2 is given by:

$$v_2 := (\mathrm{M}_h, \mathrm{T}_c, \mathrm{M}_c)^T \in \mathbb{R}^3 \tag{12}$$

in which M_h stands for the mass flow rate at the downstream inlet of the heat exchanger NEF_1 while T_c and M_c are the temperature and the mass flow rate at the upstream outlet of the same heat exchanger.

For any initial state $x(k)$ and any control profile $\mathbf{u}$ defined over some prediction horizon of length N, the corresponding nominal [disturbance-free] trajectories, denoted by $\mathbf{x}(\cdot, \mathbf{u}, x(k)|v)$, which are obtained by applying the recursive formula (10), lead to the following so-called **coherence constraints** defined for all $i \in \{0, \ldots, N-1\}$:

$$\begin{aligned} v_1(k+i) &= g_1(\mathbf{x}_2(k+i, \mathbf{u}_2, x_2(k)|v_2), u_2(k+i), v_2(k+i)), \\ v_2(k+i) &= g_2(\mathbf{x}_1(k+i, \mathbf{u}_1, x_1(k)|v_1), u_1(k+i), v_1(k+i)), \end{aligned}$$

These constraints simply express the fact that the signal profiles $\boldsymbol{v} := (\boldsymbol{v}_1, \boldsymbol{v}_2)$ are compatible with the system's nominal coupled equations. This can be shortly written by introducing the following straightforward notation:

$$\boldsymbol{v}_1 = \mathbf{g}_1(\mathbf{u}_2, x_2(k), \boldsymbol{v}_2) \tag{13}$$

$$\boldsymbol{v}_2 = \mathbf{g}_2(\mathbf{u}_1, x_1(k), \boldsymbol{v}_1) \tag{14}$$

Indeed, the r.h.s of (13) and (14) can be viewed as functions of $x_s(k)$, $\mathbf{u}_s$ and $\boldsymbol{v}_s$ (for $s \in 1, 2$), since the trajectories $\mathbf{x}_s$ depend on $x_s(k)$, $\mathbf{u}_s$ and $\boldsymbol{v}_s$ (for $s \in 1, 2$).

2.2. Overview of the hierarchical control architecture

In this section, the concept of coordinator that is invoked in the hierarchical design is introduced. The role of the coordinator is to control the system through two operation modes which are described hereafter:

1. In the first mode, the objective is to regulate the system around the nominal point $x = 0$ in spite of the unmeasured disturbances (disturbance-rejection mode). This is the main objective of the cryogenic refrigerator.
2. In the second mode, the coordinator can temporarily drive the system to a different steady-state corresponding to a new set-point $y \neq 0$. For instance, the operator might decide to change the level Ltb_{131} of liquid helium in the bath or the temperature Ttb_{108}. This corresponds to a change in the corresponding set-points.

These two modes can be taken into account by using different set-points and different weighting matrices in the centralized cost function, namely:

$$J_c(\mathbf{u}, x(k), r^d \ |v) = \sum_{s=1}^{2} J_s(\mathbf{u}_s, x_s(k), r^d_s, |\boldsymbol{v}_s) \tag{15}$$

where

$$J_s(\mathbf{u}_s, x_s(k), r^d_s, |\boldsymbol{v}_s) := \sum_{i=1}^{N} \|r^d_s - h_s(k+i)\|^2_{Q^{(s)}_c} + \|u_s(k+i) + U^0_s\|^2_{R^{(s)}_c}$$

where $h_s(k+\cdot)$ represents the output profile given by:

$$h_s(\sigma) = h_s(\mathbf{x}_s(\sigma, \mathbf{u}_s, x_s(k)|\boldsymbol{v}_s), \mathbf{u}_s(\sigma), \boldsymbol{v}_s(\sigma)) \tag{16}$$

Note that $J_s(\mathbf{u}_s, x_s(k), r^d_s|\boldsymbol{v}_s)$, $s \in \{1, 2\}$ stand for the local costs that are computed for given $\boldsymbol{v}_s$. $Q^{(1)}_c \in \mathbb{R}^{2\times 2}$, $Q^{(2)}_c \in \mathbb{R}$, $R^{(1)}_c \in \mathbb{R}^{(2\times 2)}$ and

$R_c^{(2)} \in \mathbb{R}^+$ are positive semidefinite weighting matrices to penalize the distance of outputs to set-points, and to penalize the usage of the control action. Note that the output set-points $r_1^d \in \mathbb{R}^2$ and $r_2^d \in \mathbb{R}$ represent the desired values of the deviation from the real set-points $r_s^d + Y_s^0$.

Remark 2.1. It is important to underline the difference between the central cost used by the coordinator and the stability oriented costs used by the local MPC controllers. Indeed, stability-oriented MPC formulations (local MPC) need to incorporate stabilizing terms that are not directly linked to the economic cost (central) and thus optimize a cost that is not exactly equal to the economic cost. The coordinator role focuses on the economic cost using the stable results coming from the local subsystems.

At this stage, we have all we need to state the hierarchical control requirements:

Hierarchical control requirements. A key feature in the proposed hierarchical framework is the concept of **auxiliary set-points**. Indeed, while r^d stands for the desired set-points at the central level, there is a need to define the so-called **auxiliary set-points**. These are the set-points that are to be used in the local level so that the central cost function (15) is minimized. These auxiliary set-points have no reasons to be equal to the central (original) set-points as the expressions and the weighting used in the local and the central levels are different, as mentioned above.

A hierarchical control scheme should be defined where both local MPC and NMPC (designed for S_1 and S_2 respectively), receive the appropriate optimal auxiliary set-points r_1^{opt} and r_2^{opt} that minimize the central cost (15). Moreover, the following operational conditions have to be satisfied:

1. S_1 and S_2 communicate only with the coordinator.
2. The coordinator has no knowledge regarding the mathematical models of S_1 and S_2 nor the details of their local controllers' settings.
3. The modes mentioned before are handled by changing only the desired set-points r^d and the weighting matrices Q_c and R_c. This should leave the local controller formulations and design unchanged.

Therefore, the coordinator computes the optimal **auxiliary set-points** r^{opt} in order to minimize the central cost J_c corresponding to the desired set-point r^d. In order to do this, the hierarchical control algorithm illustrated in Fig. 2 is proposed. More precisely, the coordinator constructs a grid of auxiliary set-points $\mathbf{G}$. Then, the fixed-point iteration described in Section 3 is executed in order to evaluate the central cost $J_c(r)$ at all candidate auxiliary set-points $r \in \mathbf{G}$. Using these evaluations, a quadratic approximation $\hat{J}_c$ of the central cost J_c is obtained.

Having this map at its disposal, the coordinator computes the candidate auxiliary set-points r_c^{opt} by solving a quadratic optimization problem inside a continuously updated trust-region $\mathcal{R}$. A new round of set-point iterations is used to check whether this presumably optimal value does effectively improve the central cost. Depending on the answer, the trust-region size used to define the grid of auxiliary set-points in the next sampling instant is updated (size is increased in case of success and decreased otherwise). Depending on the case, the optimal value $r^{opt}(k)$ to be used is either the one just found, namely r_c^{opt} or the previous value $r^{opt}(k-1)$ adopted in the previous sampling period.

All these steps are successively explained in the remainder of the paper before a set of numerical investigations is proposed to illustrate the efficiency of the proposed framework.

3. Recalls on the fixed-point iteration framework

Let us first recall the fixed-point iteration based algorithm proposed in Alamir et al. [2] which is used to evaluate the central cost for any candidate auxiliary set-point r. Then the way the loop is closed by finding the best auxiliary set-point is explained.

3.1. Computing the central cost for a given auxiliary set-point: the fixed-point iteration

In this section, a global set-point $r^d = (r_1^d, r_2^d)$ is given together with a value r of the auxiliary set-point at some sampling instant k.

The task of the coordinator is to evaluate the value of the central cost J_c (15) if the auxiliary set-point $r = (r_1, r_2)$ is adopted by the local controllers. This has to be done in spite of the absence of knowledge regarding the mathematical model, the current states or control design used at the local subsystems level.

Note that the central optimization problem can be viewed as a problem in the extended decision variable $(\mathbf{u}, \boldsymbol{\nu})$ provided that one adds the coherence constraints (13) and (14) on $\boldsymbol{\nu}$, namely:

$$\min_{\mathbf{u}_1, \mathbf{u}_2, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2} \sum_{s=1}^{2} J_s(\mathbf{u}_s, x_s(k), r_s^d | \boldsymbol{\nu}_s) \tag{17}$$

$$\text{subject to } \begin{cases} \boldsymbol{\nu}_1 = \mathbf{g}_1(\mathbf{u}_2, x_2(k), \boldsymbol{\nu}_2) \\ \boldsymbol{\nu}_2 = \mathbf{g}_2(\mathbf{u}_1, x_1(k), \boldsymbol{\nu}_1) \end{cases} \tag{18}$$

Note that the two individual costs are now conceptually decoupled for any given choice of coupling signal profiles $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$. The coupling now appears in the equality constraints (18).

In [2], a method has been proposed to estimate the central cost defined by (17) using a fixed-point iteration in which, given a pair of set-points $r = (r_1, r_2)$, the coordinator starts with some initial guesses regarding the coupling signals:

$$\boldsymbol{\nu}_1^{(\sigma)}, \boldsymbol{\nu}_2^{(\sigma)} \quad ; \quad \sigma = 0 \tag{19}$$

These current guesses are sent to the subsystems S_s, $s \in \{1, 2\}$ so that each subsystem can compute the corresponding optimal control profile (should the coupling signal profile be correct), namely:

$$\mathbf{u}_s^{opt}(r_s, x_s(k) | \boldsymbol{\nu}_s^{(\sigma)}) \tag{20}$$

In order to compute $\mathbf{u}_s^{opt}$, given the candidate auxiliary set-points r_s and coupling profile $\boldsymbol{\nu}_s^{(\sigma)}$, the local controllers solve in parallel and independently their local stability-oriented optimization problems, namely:

$$\begin{aligned} \mathcal{P}_s: \quad & \min_{\mathbf{u}_s} \sum_{i=1}^{N} \left(\frac{i}{N}\right)^q \left[\|x_s(k+i) - x_s^{sp}(r_s)\|_{Q_s}^2 \right. \\ & \left. + \|u_s(k+i) - u_s^{sp}(r_s)\|_{R_s}^2 \right] \\ & \text{subject to :} \\ & x_s(k+i) = \mathbf{x}_s(k+i, \mathbf{u}_s, x_s(k) | \boldsymbol{\nu}_s^{(\sigma)}) \\ & \underline{\mathbf{u}}_s \le \mathbf{u}_s \le \overline{\mathbf{u}}_s \end{aligned} \tag{21}$$

where

- ✓ $(x_s^{sp}(r_s), u_s^{sp}(r_s))$ is the steady pair that is consistent with the candidate set-point r_s and the given exogenous signal $\boldsymbol{\nu}^{(\sigma)}$.
- ✓ $x_s(k+i) = \mathbf{x}_s(k+i, \mathbf{u}_s, x_s(k) | \boldsymbol{\nu}_s^{(\sigma)})$ is the predicted state of the subsystem S_s at instant $k+i$ given the control profile $\mathbf{u}_s$, the initial state $x_s(k)$ and the presumed coupling profile $\boldsymbol{\nu}_s^{(\sigma)}$.
- ✓ $Q_1 \in \mathbb{R}^{20\times20}$, $Q_2 \in \mathbb{R}^{20\times20}$, $R_1 \in \mathbb{R}^{2\times2}$ and $R_2 \in \mathbb{R}$ are the weighting matrices on the states and the control inputs of subsystems S_1 and S_2, respectively.

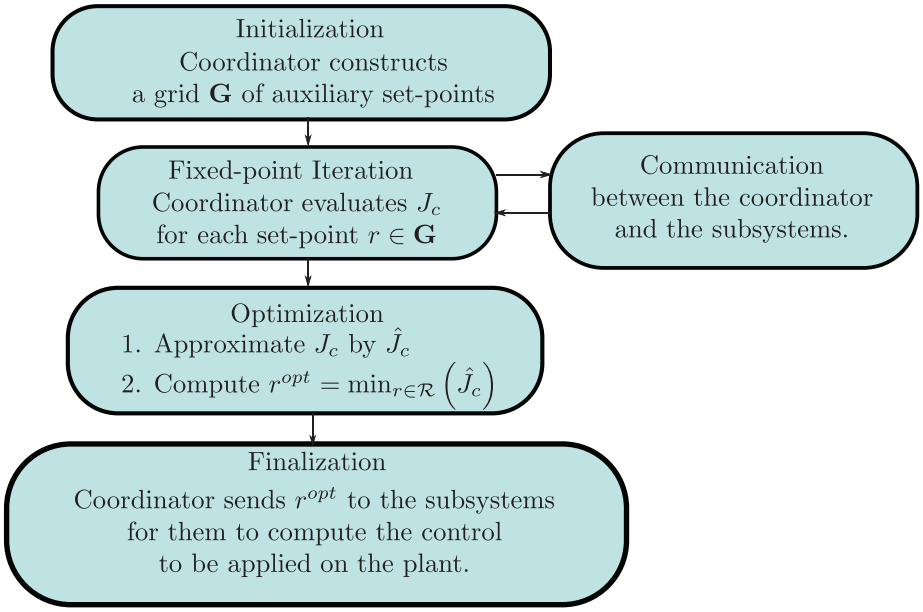

Fig. 2. block diagram of the hierarchical control algorithm.

Remark 3.1. As mentionned in Alamir et al. [2], the time-dependent weighting term $(i/N)^q$ for some $q \in \mathbb{N}$ allows to put high weight on the tail of the prediction horizon by choosing non-zero values q.

Note that the steady pair $(x_s^{sp}(r_s), u_s^{sp}(r_s))$ corresponding to the candidate set-point r_s, $s \in \{1, 2\}$, is computed by solving the optimization problem below:

$$\mathcal{P}_s^{sp}: \quad \min_{x_s^{sp}, u_s^{sp}} \|y_s^{sp} - r_s\|^2_{Q_s^{sp}} + \|u_s^{sp}\|_{R_s^{sp}} \tag{22}$$

subject to :

$$y_s^{sp} = h_s(x_s^{sp}, u_s^{sp}, v_s^{end}) \tag{23}$$

$$x_s^{sp} = f_s(x_s^{sp}, u_s^{sp}, w_s, v_s^{end}) \tag{24}$$

where $v_s^{end} := v_s^{(\sigma)}(k+N-1)$ is the last element of the exogenous profile $\boldsymbol{v}_s^{(\sigma)}$, whereas $Q_1^{sp} \in \mathbb{R}^{2\times 2}$, $Q_2^{sp} \in \mathbb{R}$, $R_1^{sp} \in \mathbb{R}^{2\times 2}$ and $R_2^{sp} \in \mathbb{R}$ are respectively the weighting matrices on output and input. Note that (24) is the stationary condition associated to the set-point r_s.

Once the control input profiles $\mathbf{u}_s^{opt}$, $s \in \{1, 2\}$, are obtained, each subsystem computes the corresponding coupling profile and sends it to the coordinator :

$$\begin{aligned} &S_1 \text{ sends to coordinator} \rightarrow \hat{\boldsymbol{v}}_2^{(\sigma+1)} := \mathbf{g}_2(\mathbf{u}_1^{opt}, x_1(k), \boldsymbol{v}_1^{(\sigma)}) \\ &S_2 \text{ sends to coordinator} \rightarrow \hat{\boldsymbol{v}}_1^{(\sigma+1)} := \mathbf{g}_1(\mathbf{u}_2^{opt}, x_2(k), \boldsymbol{v}_2^{(\sigma)}) \end{aligned}$$

In order to force the convergence of the iteration, a filtering dynamics has been proposed in Alamir et al. [2] which is defined by:

$$\begin{aligned} \boldsymbol{v}^{(\sigma+1)} &= F(\boldsymbol{v}^{(\sigma)}, \hat{\boldsymbol{v}}^{(\sigma+1)}) \\ &= (\mathbb{I} - \Pi)\cdot \boldsymbol{v}^{(\sigma)} + \Pi \cdot \hat{\boldsymbol{v}}^{(\sigma+1)} \quad \text{with} \quad \boldsymbol{v}^{(\sigma)} = \begin{bmatrix} \boldsymbol{v}_1^{(\sigma)} \\ \boldsymbol{v}_2^{(\sigma)} \end{bmatrix} \end{aligned} \tag{25}$$

The computation of the Π matrix and the resulting proof of convergence (in the linear unconstrained case) are described in Alamir et al. [2].

The filtered estimates $\boldsymbol{v}_s^{(\sigma+1)}$, are sent by the coordinator to the subsystems. Based on these values, the subsystems are able to compute the new optimal control profiles

$$\mathbf{u}_s^{opt}(r_s, x_s(k)|\boldsymbol{v}_s^{(\sigma+1)})$$

and corresponding profile of $\hat{\boldsymbol{v}}^{(\sigma+2)}$, which is then sent back to the coordinator for a new iteration. It is important to note that it is only when the iteration of the fixed point converges to some $\boldsymbol{v}^{(\infty)}(r)$ that the coherence constraints (13) and (14) become satisfied and the quadruplet:

$$\left(\mathbf{u}_1^{opt}(r_1, x_1(k)|\boldsymbol{v}_1^{(\infty)}(r)), \mathbf{u}_2^{opt}(r_2, x_2(k)|\boldsymbol{v}_2^{(\infty)}(r)), \boldsymbol{v}_1^{(\infty)}(r), \boldsymbol{v}_2^{(\infty)}(r)\right) \tag{26}$$

can be viewed an admissible sub-optimal solution to the constrained central problem (17) and (18). Practically, the iterations are stopped if one of two termination criteria is reached:

$$\sigma \geq \sigma_{max} \quad \text{or} \quad \epsilon := \max(|\boldsymbol{v}^{(\sigma+1)} - \boldsymbol{v}^{(\sigma)}|) \leq \epsilon_{max} \tag{27}$$

Then, each subsystem computes its contribution to the central cost (using the central r^d), namely:

$$J_s(\mathbf{u}_s^{opt}(r_s), x_s(k), r_s^d|\boldsymbol{v}_s^{(\infty)}(r))$$

and send it to the coordinator. After receiving these evaluations, the coordinator computes the central cost (15), which is now considered as a function of the auxiliary set-point $r := (r_1, r_2)$, r^d and $\boldsymbol{v}^{(\infty)}(r)$:

$$J_c(r|r^d, \boldsymbol{v}^{(\infty)}(r)) := \sum_{s=1}^{2} J_s\left(\mathbf{u}_s^{opt}(r_s), x_s(k), r_s^d|\boldsymbol{v}_s^{(\infty)}(r)\right) \tag{28}$$

Note that the estimation process of the central cost (left-hand side of (28)) by the coordinator, as described above, does not involve the knowledge of the state $x_s(k)$ nor the knowledge of the optimal control $\mathbf{u}_s^{opt}(r_s)$ but the estimation of the local costs and the coupling signals sent by the subsystems. Indeed, the estimation of local costs depends on the coupling signals profiles $\boldsymbol{v}^{(\infty)}(r)$ which encompasses all the information inaccessible to the coordinator and which are transmitted to the latter upon the convergence of the fixed point iterations. The previous discussion is summarized in Fig. 3 and Algorithm 1 .

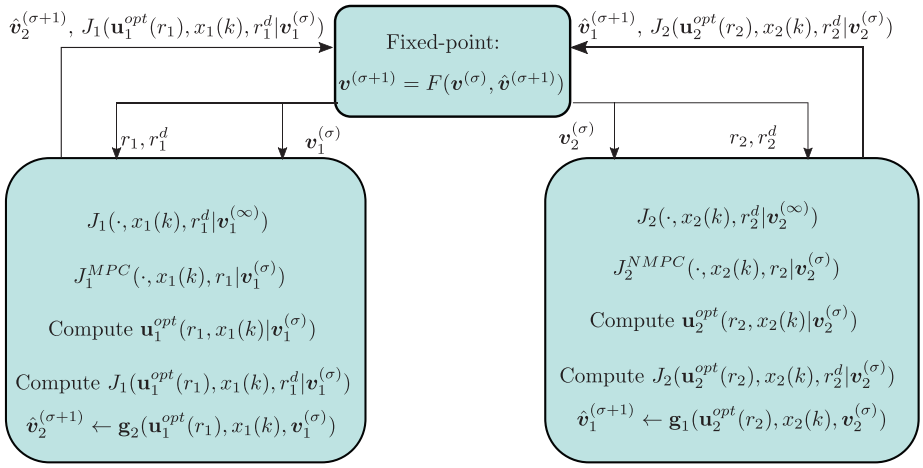

Fig. 3. Schematic view of the fixed-point iteration at instant k [2]. The states $x_1(k)$, $x_2(k)$ and the set-points r_s and r_s^d are considered frozen. Note that subsystem S_2 uses NMPC as a local controller.

Algorithm 1: Fixed-point-iteration-based-algorithm for a given set-point vector r.

Initialization: $\sigma \leftarrow 0, \boldsymbol{v}_s^{(0)} \leftarrow \mathbf{0}, \epsilon \leftarrow \infty$;
while $(\sigma \leq \sigma_{\max})$ *and* $(\epsilon \geq \epsilon_{\max})$ **do**
  for $s \leftarrow 1, \ldots, 2$ **do**
    `// Parallel computation performed by the subsystems`
    $x_s^{sp}, u_s^{sp} \leftarrow$ subsystem S_s solves $\mathcal{P}_s^{sp}$ in (22);
    $\mathbf{x}_s, \mathbf{u}_s \leftarrow$ subsystem S_s solves $\mathcal{P}_s$ in (21);
    Subsystem S_s computes $\hat{\boldsymbol{v}}_{s'}^{(\sigma)}$ $(s' \neq s)$ and sends it to coordinator;
  end
  Coordinator computes $\boldsymbol{v}_s^{(\sigma+1)}$ by (25) and sends it to subsystem S_s;
  $\epsilon \leftarrow \max(|\boldsymbol{v}_s^{(\sigma+1)} - \boldsymbol{v}_s^{(\sigma)}|)_{s=1.2}$;
  $\sigma \leftarrow \sigma + 1$
end
for $s \leftarrow 1, \ldots, 2$ **do**
  Subsystem s computes $\mathbf{h}_s$ then J_s, and sends J_s to coordinator;
end
Coordinator computes $J_c(r) \leftarrow \sum_{s=1}^{2} J_s(r_s)$

3.2. Optimizing the auxiliary set-points

In the previous section, it has been shown how the fixed-point iterations help the coordinator computing the central cost for a **given auxiliary set-point** r. Recall, however, that the role of the coordinator is to optimize the choice of the auxiliary set-points so that the central cost can be minimized. This section explains how the coordinator can use successive evaluations of the central cost for different candidate auxiliary set-points to build a quadratic approximation of the central cost (as a function of the auxiliary set-points at the current sampling time k) in order to derive a candidate optimal auxiliary set-point r^{opt}.

3.2.1. Approximating the central cost

Using the fixed-point iteration, the coordinator can compute for each auxiliary set-point $r = (r_1^T, r_2^T)^T$ the corresponding value of the central cost:

$$J_c(r|r^d, \boldsymbol{v}^{\infty}(r)) \tag{29}$$

after convergence of the fixed-point iteration.

The central problem in the coordination layer can now be defined as follows:

$$r^{opt} = \text{argmin}_r J_c(r\ |r^d, \boldsymbol{v}^{\infty}(r)) \tag{30}$$

In order to solve (30), the central cost (15) will be approximated by a quadratic function, namely:

$$\hat{J_c} = \frac{1}{2} r^T Q r + f^T r + c \tag{31}$$

where $Q \in \mathbb{R}^{n_r \times n_r}$, $f \in \mathbb{R}^{n_r}$ and $c \in \mathbb{R}$, with n_r being the dimension of vector r.

These unknown parameters can be identified if the coordinator disposes of the values of the central cost at, at least, $(n_r+1)(n_r+2)/2$ different auxiliary set-points. The remainder of this section is devoted to explaining the way this is done by the coordinator. Note that this is a single possibility among many other possibilities of optimizing a black-box given function through different evaluations of its values at a set of possible points within its domain of definition. This is linked to the general task of derivative-free optimization.

In [2], the grid of auxiliary set-points $\mathbf{G}(k)$ is constructed using a fixed grid around the original set-point r^d. In the present version, a moving grid around the suboptimal solution found at the last instant. Moreover, the size ρ of the trust-region is modified, which will be described in Section 3.2.2, depending on the relevance of quadratic approximation. At each sampling instant k, the grid $\mathbf{G}(k)$ of auxiliary set-points for the evaluation (at the sampling instant k) of the central cost is defined **around the previous optimal value** $r^{opt}(k-1)$ **as follows:**

$$\mathbf{G}(k) := \text{Pr}\Big(r^{opt}(k-1) + \Delta(\rho(k-1)), \mathcal{R}\Big) \tag{32}$$

where

- $\mathcal{R}$ is an admissible set of r, namely:
$$\mathcal{R} = \{r \quad | \quad r_{\min} \leq r \leq r_{\max}\} \tag{33}$$
where $r_{\min}, r_{\max} \in \mathbb{R}^{n_r}$ are a priori defined bounds on possible values of the set-points.
- for a discrete subset $\mathbb{D} \subset \mathbb{R}^{n_r}$, the notation $\text{Pr}(\mathbb{D}, \mathcal{R})$ denotes the discrete set obtained by projecting all the elements of $\mathbb{D}$ on the hypercube $\mathcal{R}$.
- $\rho \in \mathbb{R}_+$ is positive real (size of the trust region where the quadratic approximation is presumably relevant).
- $\Delta(\rho)$ is a discrete set of displacements in $\mathbb{R}^{n_r}$ defined around 0 with distances that are proportional to ρ so that

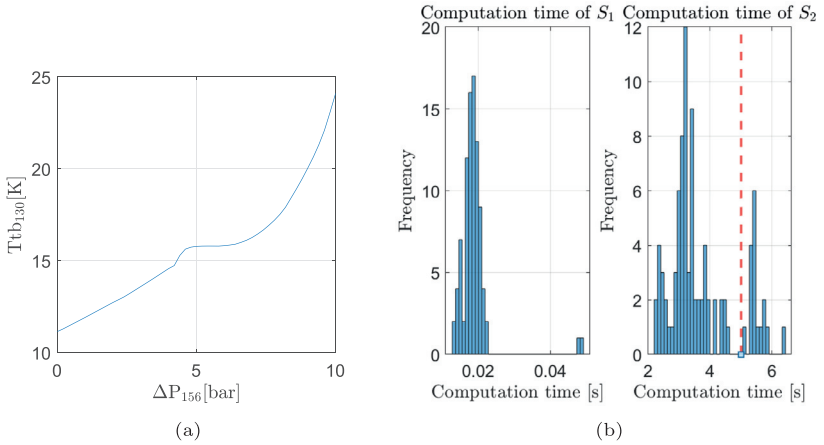

Fig. 4. (a) shows the evolution of the stationary value of Ttb_{130} as a function of $\Delta \mathrm{P}_{155}$. (b) shows the histograms of the computation times of S_1 and S_2 during the simulated scenario.

$r^{opt}(k-1)+\Delta(\rho)$ represent the set of different auxiliary set-points around the previous optimal value to be visited and where the cost is to be evaluated). More precisely, the subset $\Delta(\rho) \subset \mathbb{R}^{n_r}$ is defined by (m is supposed to be odd):

$$\Delta(\rho) := \left\{-\frac{(m-1)}{2}\rho, \ldots, -\rho, 0, \rho, \ldots, \frac{(m-1)}{2}\rho\right\}^{n_r} \tag{34}$$

The trust region size ρ is updated at each instant k, which will be described later. Recall that the identifiability of the quadratic form coefficients is possible provided that $m^{n_r} \geq (n_r+1)(n_r+2)/2$.

Based on the above definitions, the grid $\mathbf{G}(k)$ is constructed by using (32), the evaluation of the central cost $J_c(\cdot)$ at every set-point $r \in \mathbf{G}(k)$ is performed by using the fixed-point methodology introduced in Section 3. The values

$$J_c(r^{(j)}), \quad j = 1, \ldots, n_{ev} \leq m^{n_r}$$

enable to compute the parameters of the quadratic form:

$$\text{(Coordinator)} \min_{Q,f,c} \sum_{j=1}^{n_{ev}} \left| J_c(r^{(j)}) - \left[\frac{1}{2}\|r^{(j)}\|_Q^2 + f^T r^{(j)} + c\right]\right| \tag{35}$$

Once Q, f and c are available, a candidate optimal set-point r_c^{opt} (that minimizes the quadratic approximation) can be computed. Note, however, that since the central cost is not necessarily quadratic, this candidate optimal cost does not necessarily induce a decrease in the central cost. This can happen when the trust-region parameter ρ is too large for the quadratic approximation to be relevant. In such case, the size ρ should be reduced. This mechanism is discussed in the next section.

3.2.2. Trust region updating law of ρ

As mentioned previously, the parameter ρ defines the size of the neighborhood of the current desired set-point r^d over which the better value is computed based on the current quadratic approximation of the cost function. On one hand, ρ must be sufficiently high to ensure a rapid decrease of the cost value. On the other hand, small values of ρ might be required in order for the quadratic approximation to be relevant. Hence, ρ should be updated accordingly: ρ is increased if the quadratic approximation induces a decrease of the cost function while ρ is decreased otherwise.

Concretely, the following quadratic problem is first solved to obtain the candidate value $r_c(k)$

$$r_c^{opt}(k) = \text{argmin}_{r \in \mathscr{P}(k)} \hat{J}_c(r) \tag{36}$$

where $\mathscr{P}(k)$ is given by:

$$\mathscr{P}(k) := \text{Conv}\{\text{Pr}(r_d(k) + \Delta(\rho(k-1)), \mathcal{R})\} \tag{37}$$

Once the candidate $r_c^{opt}(k)$ is obtained, the corresponding cost is computed by launching the Algorithm 1 to obtain $J_c(r_c^{opt})$. The quadratic approximation is said relevant if it meets the condition below:

$$J_c(r_c^{opt}) < \min\{J_c(r^{(j)}) \mid r^{(j)} \in \mathbf{G}(k)\} \tag{38}$$

Therefore, the trust-region size ρ is updated according to:

$$\rho(k) := \begin{cases} \beta^+ \cdot \rho(k-1) & \text{if (38) is satisfied} \\ \beta^- \cdot \rho(k-1) & \text{otherwise} \end{cases} \tag{39}$$

where $\beta^+ \geq 1$ and $\beta^- \in (0,1)$ denote respectively the expansion and the contraction factors. Finally, the updating law for r^{opt} is given by:

$$r^{opt}(k) := \begin{cases} r_c^{opt}(k) & \text{if (38) is satisfied} \\ r^{opt}(k-1) & \text{otherwise} \end{cases} \tag{40}$$

where $r^{opt}(k-1)$ is the solution found at the previous instant $k-1$. The so adopted set-point $r^{opt}(k)$ is then sent to the subsystems with an *end-of-iterations* flag, which allows the subsystems to compute their corresponding control profiles. Finally, according to the MPC definition, the first action in each profile, namely:

$$u_s(k) := [\mathbb{I}_{n_s^u}, \mathbb{O}_{n_s^u}, \ldots, \mathbb{O}_{n_s^u}]\mathbf{u}_s^{opt}(r_s^{opt}(k), x_s(k) | \boldsymbol{v}_s^{(\sigma_{\max})}) \tag{41}$$

is applied by subsystem S_s during the sampling period $[k, k+1]$.

Unfortunately, because of the presence of nonlinearity and constraints, the computation of the n_{ev} necessary evaluations for central cost approximation might require a computation time that goes beyond the available sampling time T_s. The following section proposes a method to reduce the computation time with a rather little impact on the quality of the resulting closed-loop performance.

4. Distributing the optimization over time

Since using constrained nonlinear MPC induces a significant increase in the computation time, it might be impossible to compute a solution r^{opt} (following the scheme of the previous section) for the next sampling period in the presence of limited computational resources. In order to overcome this potential issue, this section proposes a technique inspired by Alamir [1] which is based on the idea of distributing the optimization over time. To facilitate the following explanations, the notation k and $k+1$ are used to refer to

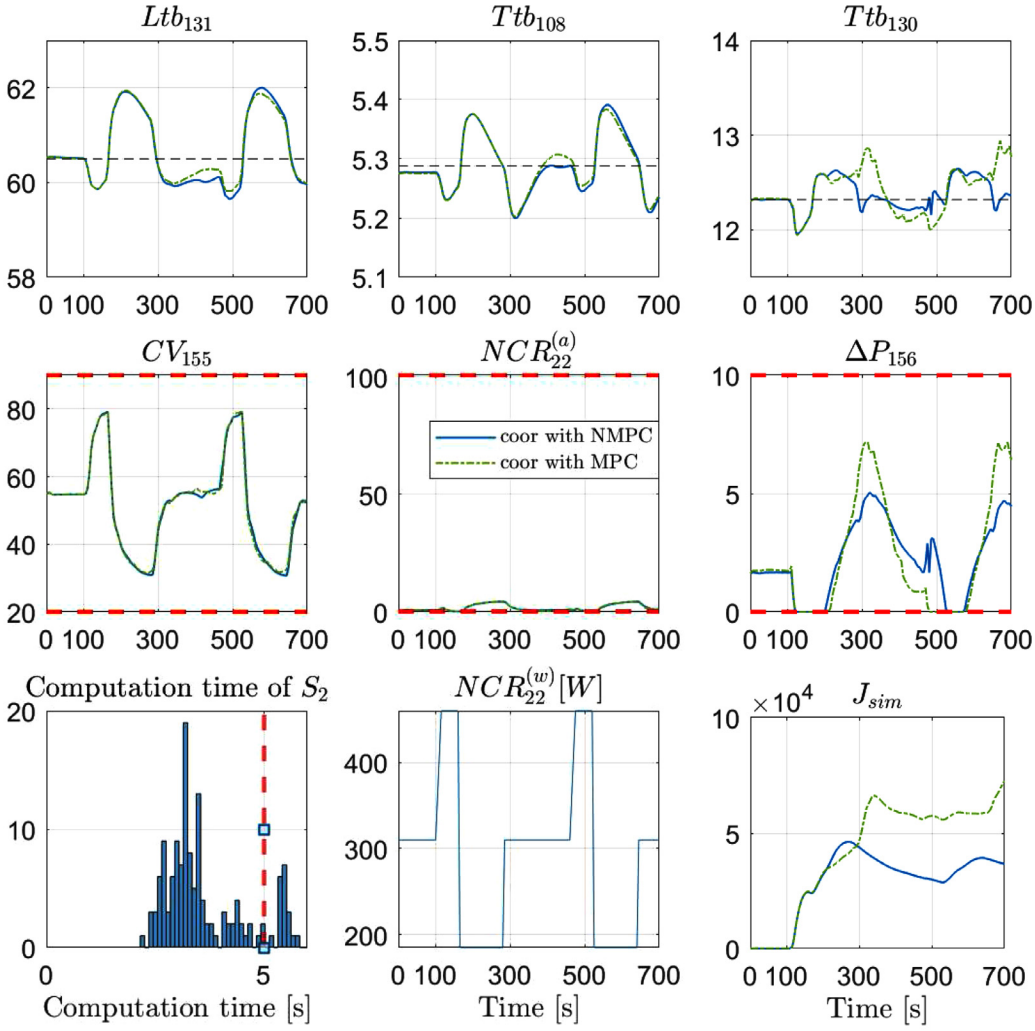

Fig. 5. Comparison of closed-loop behavior under the proposed hierarchical framework with Linear MPC (dash-dot green line) and Nonlinear MPC (solid blue line) used in modeling the Brayton cycle. The first row presents the outputs, and the second one presents the inputs of the system. The subplot(3,3,8) illustrates the periodic disturbance applied to the helium bath. The subplot(3,3,7) shows the computation time when using the Nonlinear MPC. Note also that the computation time limit is not considered for this simulation (the solver takes the time needed to perform the assigned computations). The subplot(3,3,9) depicts the evolution of the cost index J_{sim} in time. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

instants $k\tau_u$ and $(k+1)\tau_u$ with τ_u being the control updating period, namely, the time during which the computation of a new optimal open-loop sequence is recomputed to implement the MPC feedback. Note that τ_u is not necessarily equal to the sampling time T_s. The process described in this section will be executed during the updating period $[k, k+1]$ as long as the computation time does not exceed τ_u.

Recall that the approximation of the cost function $J_c(r)$ needs the evaluation of J_c at $n_{ev} \geq (n_r+1)(n_r+2)/2$ values of the auxiliary set-points. By reducing the number of degrees of freedom (DOF) of vector r to be improved from n_r to $n_z < n_r$, only $(n_z+1)(n_z+2)/2$ realizations would be needed, which accordingly leads to a decrease of the computation burden per updating period.

More precisely, a change in the decision variable is cyclically operated by defining a reduced dimensional parameterization of r of the form:

$$\tilde{r} = Mr + Dz \tag{42}$$

where $\tilde{r} \in \mathbb{R}^{n_r}$, $M \in \mathbb{R}^{n_r \times n_r}$ and $D \in \mathbb{R}^{n_r \times n_z}$. Moreover, the transformation matrices M and D are changed in a cyclic way in order to *explore* all the degrees of freedom of r after a finite number of successive iterations. This is explained in a more detailed way in the remainder of this section.

At the beginning of each updating period k, the optimization problem to be solved is given by:

$$z^\star(k) = \text{argmin}_z \hat{J}_c(M^{(j_k)} r^\star(k-1) + D^{(j_k)} z) \tag{43}$$

where the transformation defined by the matrices $M^{(j_k)}$ and $D^{(j_k)}$ is defined in order to assign some components of the vector r to be equal to the corresponding components of the previous solution $r^\star(k-1)$ while leaving as degrees of freedom the n_z remaining components that define the reduced dimensional decision variable z. Note that the definition of the transformation matrices depends on the updating instant k through the upper index j_k which is a cyclic variable defined by:

$$j_k = (j_{k-1} + 1) \mod n_r \tag{44}$$

In the numerical investigation, the following two configurations are tested in order to illustrate the proposed methodology:

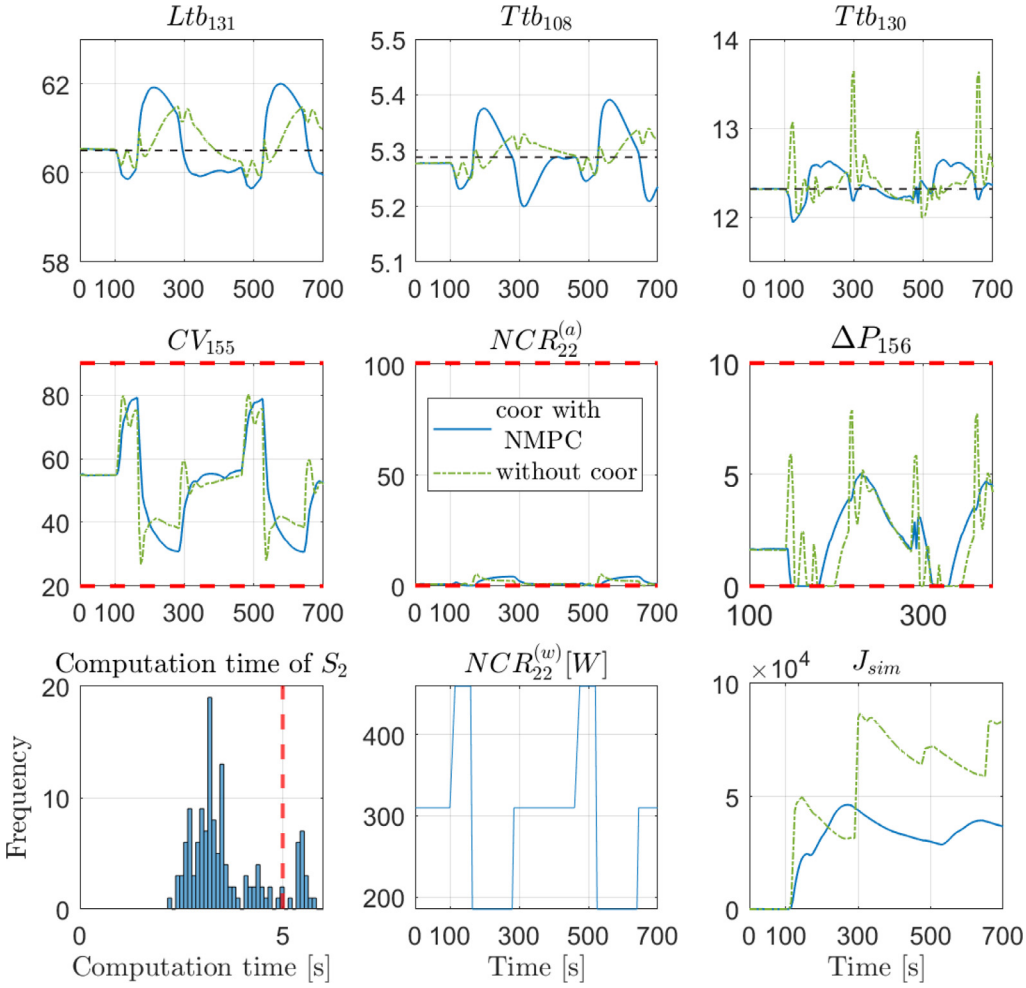

Fig. 6. Closed-loop behavior with coordination (solid blue line) and without coordination (dash-dot green line). The first row presents the outputs, and the second one presents the inputs of the system. The hierarchical control method gives a better cost J_{sim} than the decentralized method (closed-loop cost decreased by 56%). Note that the cost under linear MPC (Fig. 5) corresponds to a reduction of only 12% compared to the decentralized control without coordination). Note also that the computation time limit is not considered in this simulation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1. Configuration 1: $n_z = 1$, $n_r = 3$

$$M^{(0)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} D^{(0)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{45}$$

$$M^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} D^{(1)} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{46}$$

$$M^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} D^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{47}$$

4.2. Configuration 2: $n_z = 2$, $n_r = 3$

$$M^{(0)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} D^{(0)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{48}$$

$$M^{(1)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} D^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{49}$$

$$M^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} D^{(2)} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{50}$$

Note that the same methodology explained before regarding the definition of the grid of points is adopted with r and n_r respectively replaced by z and n_z. The only difference is that the number of degrees of freedom to be considered at the beginning of each updating period is reduced, and the significance of the degrees of freedom in terms of the components of r changes at each updating period.

When a sub-optimal solution $z^\star(k)$ to (43) is obtained (after the allowed number of iterations), the corresponding candidate sub-optimal auxiliary set-point $r_c^\star(k)$ is given by

$$r_c^\star(k) = M^{(j_k)} r^\star(k-1) + D^{(j_k)} z^\star(k) \tag{51}$$

This candidate value is then used to update the size of the trust region in a similar way as explained above. The method can be simply sketched by Algorithm 2 for a given updating cycle involving n_d iterations. More precisely, for-loop in Algorithm 2 allows to perform n_d iterations within the updating period. Indeed, if the computation time does not exceed the updating period $[k, k+1]$, the whole process mentioned in this section can be repeated in order to improve the sub-optimal candidate auxiliary set-point $r_c^\star(k)$.

Algorithm 2: Pseudo code for the distributed-in-time optimization.

for $l \leftarrow 1, \ldots, n_d$ **do**
  Coordinator defines a grid of auxiliary set-points $\mathbf{G}(M^{(j_k)}r^\star(k-1) + D^{(j_k)}\Delta(\rho(k-1)), \mathcal{R})$;
  Coordinator evaluates the cost function for each element r in the grid $\mathbf{G}(M^{(j_k)}r^\star(k-1) + D^{(j_k)}\Delta(\rho(k-1)), \mathcal{R})$;
  Coordinator computes the quadratic approximation $\hat{J}_c(z)$ of $J(z)$;
  Coordinator finds $z^\star(k)$ by solving (43)
  Coordinator computes the candidate auxiliary set-point $r_c^\star(k)$ according to (51);
  Coordinator updates ρ and $r^\star(k)$ using (39) and (40).
end
Coordinator sends $r^\star(k)$ to the subsystems.

5. Simulation results

In this section, some numerical simulations are proposed in order to illustrate the different concepts and solutions introduced in the paper.

5.1. The simulation parameters

First of all, a cost index is necessary in order to evaluate and compare the performances in terms of closed-loop costs associated to the different framework settings. The commonly used closed-loop central cost will be adopted, namely:

$$J_{sim} = \frac{1}{N_{sim}} \sum_{s=1}^{2} \sum_{i=1}^{N_{sim}} \|y(i) - y^d\|_{Q_{c,s}} + \|u(i)\|_{R_{c,s}} \tag{52}$$

where N_{sim} is the length of the simulation (in terms of sampling periods $T_s = 5$s), $Q_{c,s}$ and $R_{c,s}$ are the penalty matrices used in the central cost (15). There are two sets of weighting matrices for the central cost (15), which will be used in the two distinct modes describes earlier (see the beginning of Section 2.2), namely:

✓ **Mode 1**: For disturbance rejecting scenario:

$$Q_{c,1} = \begin{bmatrix} 10^4 & 0 \\ 0 & 10^4 \end{bmatrix}, \quad R_{c,1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{53}$$

$$Q_{c,2} = 10^6, \quad R_{c,2} = 0.1 \tag{54}$$

✓ **Mode 2**: For set-point tracking scenario:

$$Q_{c,1} = \begin{bmatrix} 10^6 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad R_{c,1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{55}$$

$$Q_{c,2} = 10^4, \quad R_{c,2} = 0 \tag{56}$$

where the second mode is dedicated to the tight regulation of the first output of the first subsystem, namely the liquid helium level in the bath (Ltb_{131}).

The penalty matrices for the local MPCs are fixed regardless of the scenario to the following values:

$$Q_1 = C_1^T \cdot \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \cdot C_1 + 10 \cdot \mathbb{I}_{20\times 20} \qquad R_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q_2 = C_2^T \cdot 10^3 \cdot C_2 + 10 \cdot \mathbb{I}_{20\times 20} \qquad R_2 = 1$$

where C_1 is the matrix involved in the expression of the regulated output of the subsystem S_1:

$$y_1(k) = C_1 \cdot x_1(k) + D_1 \cdot u_1(k) + D_{v1} \cdot v_1(k) \tag{57}$$

Whereas, the matrix C_2 is the partial derivative of $h_2(x_2, u_2, z_2)$ in (10) with respect to x_2 at an operating point $(x_2^{op}, u_2^{op}, v_2^{op})$. Note that in each of the local MPC settings, a penalty on the whole state is used in order to enforce the stability of the local closed-loop.

The prediction horizon N is chosen to have a length of NT_s (where $N = 100$ and $T_s = 5$ s). This corresponds roughly to 8 minutes. This setting is currently used at CEA/IRIG/DSBT and is also the one that has been used in many previous studies involving MPC control design. As for the definition of the stage costs, the exponents $q_1 = 1$ and $q_2 = 1$ are used for the cost functions used in the local MPC design [see (21)].

In order to estimate the states to be used in the local MPC implementation, each subsystem uses an appropriate observer whose synthesis is beyond the scope of this paper. The observer not only estimates the states x_s but also estimates the exogenous input v_s (extended observer). For the decentralized settings (without coordinator), the exogenous signal $v_s(k)$ is supposed to remain constant over the prediction horizon. For the updating rules of the trust region size, the parameters β^-, β^+ are set to 0.7 and 1.25, respectively.

In the next sections, the following aspects will be illustrated through the numerical simulations :

The benefit from using nonlinear models

To start, Fig. 4(a) shows the relation between the stationary values of Ttb_{130} and ΔP_{156}. From this figure, it appears clearly that the visited interval of values of ΔP_{156} involves rather high nonlinearities which explains the benefit from using Nonlinear MPC. Indeed, a comparison between the coordination with Linear MPC and Nonlinear MPC used in the Brayton cycle is presented (Fig. 5). It is shown that the temperature Ttb_{130} is better controlled when using the nonlinear models which results in an obvious improvement of the cost. This validates the first *purpose* of this contribution (need for explicit handling of the nonlinearities). Therefore, only the Nonlinear MPC design in the Brayton cycle is used in the forthcoming investigations. However, the related computation time of S_2 exceeds the time limit (subplot(3,3,7) of Fig. 5), which means that the simulated feedback actions cannot be really implemented.

Fig. 4 (b) compares the computation time of S_1 and S_2 showing that since the subsystem S_1 only involves linear MPC problem, the related computation time is very small compared to the time needed to solve general non quadratic optimization problem that is associated to subsystem S_2. The same can be said regarding the computations done by the coordinator, in view of the fact that it only performs basic operations.

In what follows, the simulation results aim at showing the benefit of using the hierarchical control framework (compared to observer-based decentralized framework) and the effectiveness of the distributed-in-time optimization heuristic, proposed in the previous section, in addressing the computation time and real-time implementability issue.

The benefit from hierarchical design

In this section, the disturbance rejection mode is simulated. The closed-loop cost of the proposed hierarchical framework is compared to the one obtained under the extended observer-based decentralized approach. More precisely, two scenarios are simulated in which the limitation on the available computation time is respectively enforced or not:

Without constraints on the computation time

Fig. 6 shows simulations without limitation in the computation time (the decided number of evaluations is assumed to be possible to achieve within the sampling period T_s). This means that the corresponding $u_s^{opt}(k)$ is implemented even if its computation time exceeds the available time within the sampling period. With

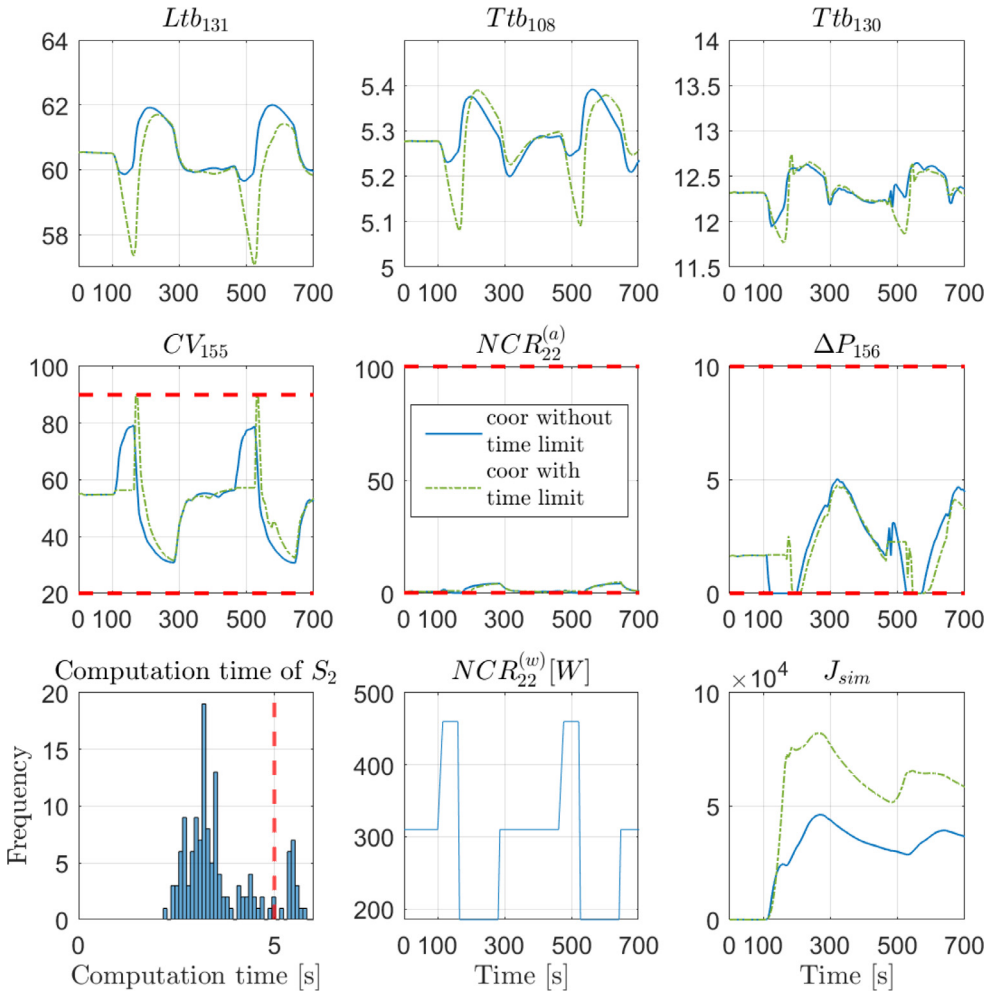

Fig. 7. Closed-loop cost comparison between the case where the limitation of the available computation time T_s is taken into account in the hierarchical framework's implementation (dash-dot green line) and the ideal case where this constraint is not considered (solid blue line). The first row presents the outputs, and the second one presents the inputs of the system. Note in particular how the control of the liquid helium level Ltb_{131} is visibly deteriorated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the use of the proposed framework, the regulation of the temperature Ttb_{130} is highly enforced compared to its behavior in the absence of coordination. One can also note that the level Ltb_{131} seems to loosely track the desired set-point which is expected since in this disturbance rejection mode 1, the focus is on the temperature Ttb_{130} as suggested by the choice of the penalty matrices [see (53) and (54)]. This can be simply checked by examining the last plot that compares the closed-loop index evolutions for the two examined settings. The comparison with the observer-based decentralized approach shows the advantage of using the hierarchical scheme should the computation be possible to perform in the allowed sampling period T_s.

Enforcing the cpu-constraint ($\tau_u \leq T_s$) on the available computation time

In this section, the constraint induced by the limited available computation time is taken into account in order to underline the benefit from using the proposed distributed-in-time optimization framework. In order to take into account the cost increase induced by the impossibility to compute the solution r^{opt} within the allocated time, Fig. 7 shows the closed-loop behavior when the subsystems apply the previous control $u_s^{opt}(k-1)$ each time the computation time exceeds the available computation time T_s. Indeed, in this case, the master cannot dispose of the needed information in order to update the approximation of the cost function, which is needed to update the value of the auxiliary set-point and the associated coupling signals that are needed to compute the updated control to be applied.

In order to conclude this subsection, Table 2 shows the normalized cost index J_{sim}^{norm} for all the settings mentioned previously. This table clearly shows the cost increase that is induced by the limitation on the available computation time. Indeed, the cost increases from 44% to 70.64% of the baseline represented by the observer-based decentralized scheme.

In the next section, the possibility to partially recover the optimal performance through the distributed-in-time optimization scheme proposed in Section 4 is investigated.

Impact of the distributed-in-time setting's parameters

Different configurations of the distributed-in-time optimization parameters (n_z, n_d and τ_u) are simulated, and the corresponding closed-loop costs J_{sim}^{norm} (normalized cost index) are reported in order to give a flavour of the impact of each choice on the results. More precisely, the testing scenario of a periodic heating disturbance is simulated again with the distributed-in-time optimization framework being implemented in the hierarchical framework.

Fig. 8 shows the comparison of the behavior of the process when using the hierarchical control combined with optimization distribution and the one without distribution under time limit constraint. Fig. 9 shows the computation time for different configura-

Table 2
The normalized cost index of for the different settings mentioned above.

	J_{sim}^{norm}
Without Coordination	100 %
Coordination with Linear MPC	87.57%
Coordination with Nonlinear MPC (Enforcing computation time limit)	70.64%
Coordination with Nonlinear MPC (Without computation time limit)	44.21%

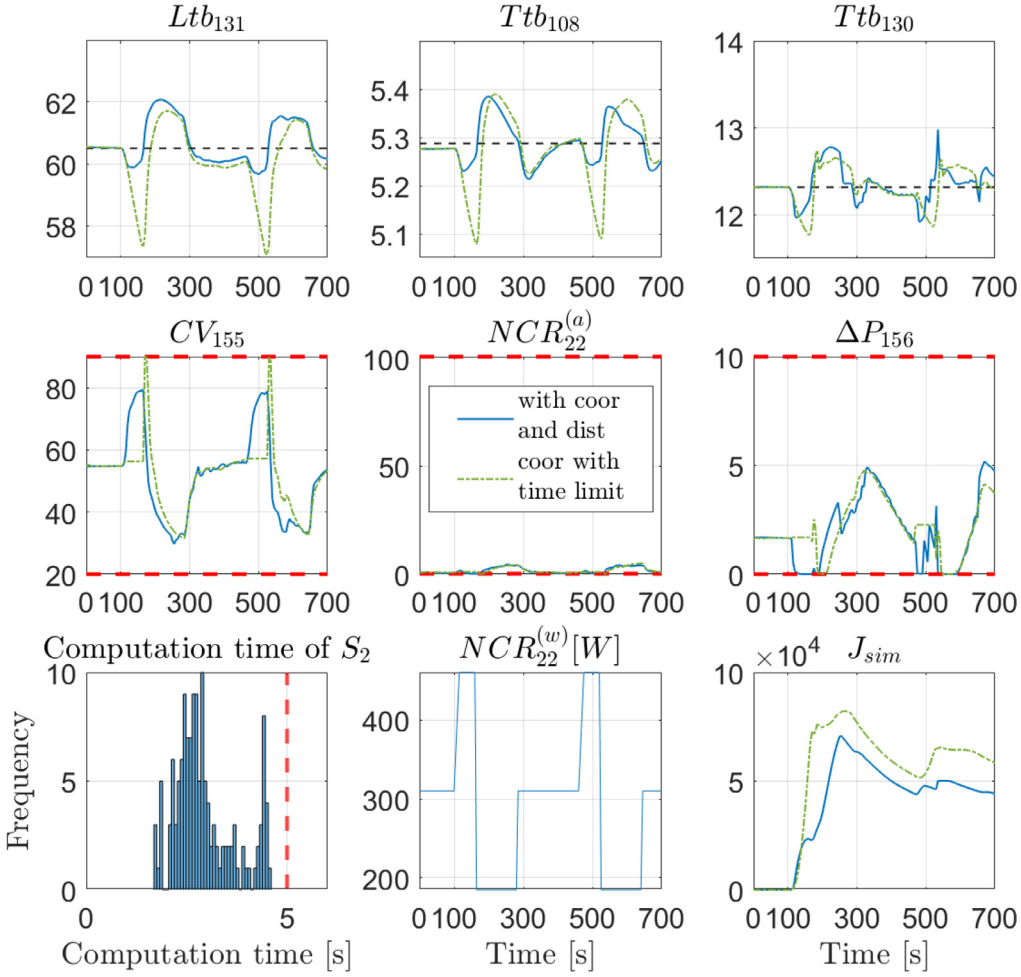

Fig. 8. Comparison of the closed-loop behavior under the proposed hierarchical framework with two different settings: 1) with distributed-in-time optimization being implemented and 2) without distributed-in-time implementation under available time limitation constraint. The first row presents the outputs and the second one presents the inputs of the system. The parameters $n_z = 2$, $n_d = 1$ and $\tau_u = 5$ s are used in the distributed-in-time optimization framework. subplot(3,3,7) shows the computation time associated to this configuration which now meet the constraint on the available computation time.

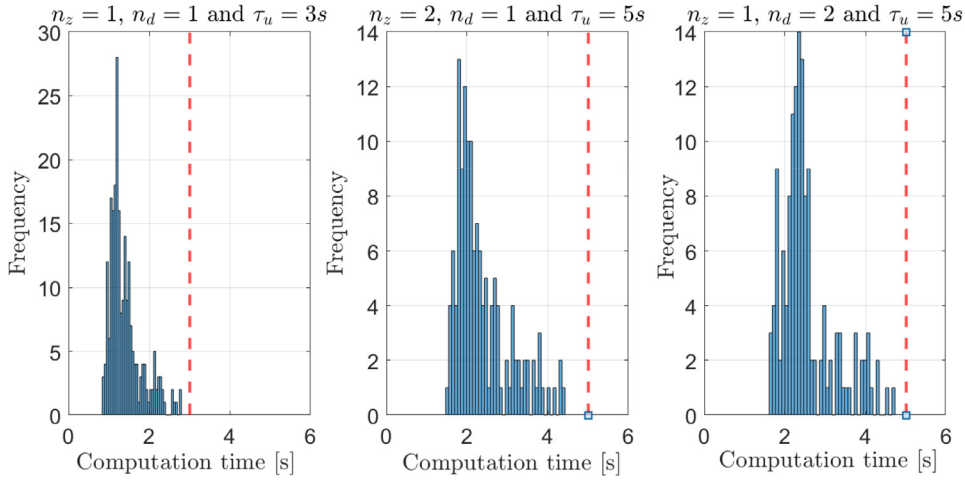

Fig. 9. Histogram of computation time of different choices of n_z and n_d. Note that the computation time of each configuration is always lower than the updating time τ_u.

Table 3
The normalized cost J_{sim}^{norm} for different configurations n_z, n_d and τ_u of the distributed-in-time optimization.

	n_z	n_d	τ_u	$t_{\max}$	J_{sim}^{norm}
Decentralized method	–	–	–	–	100%
Without distribution (with computation time limit)	–	–	–	–	70.64%
Hierarchical method	1	1	3 s	2.8 s	68.74%
With distribution	1	2	5 s	4.7 s	59.43%
	2	1	5 s	4.4 s	53.49%
Without distribution (without computation time limit)	–	–	–	–	44.21%

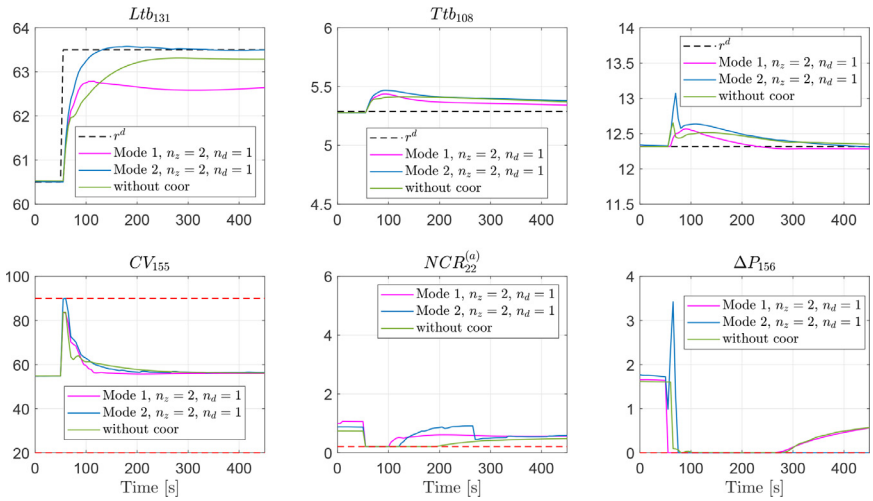

Fig. 10. Set-point tracking scenario: closed-loop responses under coordination, using distributed-in-time optimization in two different mode for the centralized cost on one hand and without coordination on the other hand. The first row presents the outputs, and the second one presents the inputs of the system. The set-point on Ltb_{131} is increased. Two configurations of Q_c and R_c of Mode 1 and Mode 2 are tested. Mode 2 (corresponding to higher penalty on Ltb_{131} deviations) allows better reference tracking while mode 1 which is dedicated to disturbance rejection and not especially to track set-point on the level. With the set of parameters $n_z = 2$, $n_d = 1$ and $\tau_u = 5s$. Note that both hierarchical design with distributed optimization are real-time compatible.

tion of n_z and n_d. Finally, Table 3 shows the associated closed-loop cost for the different configurations of the distributed-in-time optimization scheme. It comes out that the setting corresponding to $n_z = 2$, $n_d = 1$ enables to get closer to the ideal cost index drop (44%) corresponding to the non constrained computation time simulation while being fully real-time compatible.

Checking modularity: controlling the system by only tuning the central cost's definition

One of the claims of this contribution concerns the possibility to keep the local controllers unchanged (in terms of penalty) while changing the penalties of the economic cost (by the coordinator) in order to achieve different behaviors of the closed-loop system. Here, it is assumed that the operator need to change the set-point of the helium liquid level Ltb_{131}, for example, in order to embed a test facility below the liquid level. Thus, the reference tracking scenario is simulated. More precisely, the closed-loop behavior is compared under the two different modes defined in Section 5.1. This is done in order to illustrate the fact that the results can be affected in the desired direction by only modifying the central cost's definition while keeping the local controllers unchanged.

For the tracking set-point mode, Fig. 10 shows the comparison of the behavior of the process between using hierarchical control combined with optimization distribution and using decentralized control.

6. Conclusion and future works

This paper extends a recently proposed hierarchical control design framework with application to a the control of cryogenic refrigerator. It is particularly shown that the proposed framework enables the incorporation of constraints and nonlinearity in the involved models of subsystems. Moreover, a dedicated distributed-in-time optimization scheme is proposed and validated. This mechanism enables to recover a great part of the loss of optimality that might be induced by the impossibility to achieve the associated computation within the available time.

The undergoing investigations aim at pushing forward the splitting of the system into subsystems. In particular, the heat exchangers typically introduce many states with mainly linear relationship except at their boundary conditions. It should therefore be possible to split the subsystems they belong to into two subsystems of two categories: linear high dimensional ones and nonlinear small dimensional ones. However, this implies that the framework should be adapted to the case where some of the subsystems do not have controlled inputs and/or regulated outputs. Some preliminary results are quite encouraging and might be proposed in future communication.

Declaration of Competing Interest

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

References

[1] M. Alamir, A framework for real-time implementation of low-dimensional parameterized NMPC, Automatica 48 (1) (2012) 198–204.
[2] M. Alamir, P. Bonnay, F. Bonne, V.-V. Trinh, Fixed-point based hierarchical MPC control design for a cryogenic refrigerator, J. Process Control 58 (2017) 117–130.
[3] J. Andersson, J. Gillis, G. Horn, J. Rawlings, M. Diehl, Casadi: a software framework for nonlinear optimization and optimal control, Math. Program. Comput. 11 (2019) 1–36.

[4] D. Barcelli, A. Bemporad, G. Ripaccioli, Hierarchical multi-rate control design for constrained linear systems, in: 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 5216–5221, doi:10.1109/CDC.2010.5717405.
[5] F. Bonne, S. Varin, A. Vassal, P. Bonnay, C. Hoa, F. Millet, J.-M. Poncet, Simcryogenics: a library to simulate and optimize cryoplant and cryodistribution dynamics, IOP Conf. Ser. 755 (2020) 012076.
[6] S. Claudet, P. Gayet, P. Lebrun, L. Tavian, U. Wagner, Economics of large helium cryogenic systems: experience from recent projects at CERN, in: Advances in Cryogenic Engineering, Springer, 2000, pp. 1301–1308.
[7] M.D. Doan, T. Keviczky, B. De Schutter, A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints, in: 2011 50th IEEE Conference on Decision and Control and European Control Conference, 2011, pp. 5236–5241, doi:10.1109/CDC.2011.6160708.
[8] M.D. Doan, T. Keviczky, B.D. Schutter, A Hierarchical MPC Approach with Guaranteed Feasibility for Dynamically Coupled Linear Systems, Springer Netherlands, Dordrecht, 2014, pp. 393–406.
[9] D. Henry, J. Journeaux, P. Roussel, F. Michel, J. Poncet, A. Girard, V. Kalinin, P. Chesny, Analysis of the ITER cryoplant operational modes, Fusion Eng. Des. 82 (5) (2007) 1454–1459. Proceedings of the 24th Symposium on Fusion Technology
[10] R. Negenborn, J. Maestre, On 35 approaches for distributed MPC made easy, in: Distributed Model Predictive Control Made Easy, Springer, 2014, pp. 1–37.
[11] B. Picasso, D. De Vito, R. Scattolini, P. Colaneri, An MPC approach to the design of two-layer hierarchical control systems, Automatica 46 (5) (2010) 823–831.
[12] R. Scattolini, Architectures for distributed and hierarchical model predictive control—A review, Journal. Process Control 19 (5) (2009) 723–731.
[13] B.T. Stewart, J.B. Rawlings, S.J. Wright, Hierarchical cooperative distributed model predictive control, in: Proceedings of the 2010 American Control Conference, 2010, pp. 3963–3968, doi:10.1109/ACC.2010.5530634.