UNIVERSITÀ DELLA CALABRIA

Dipartimento di Elettronica,
Informatica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi e Informatica
XIV ciclo

Tesi di Dottorato

Distributed Command Governor Strategies for Multi-Agent Dynamical Systems

Francesco Tedesco

UNIVERSITÀ DELLA CALABRIA

Dipartimento di Elettronica,
Informatica e Sistemistica

Dottorato di Ricerca in
Ingegneria dei Sistemi e Informatica
ciclo

Tesi di Dottorato

Distributed Command Governor Strategies for Multi-Agent Dynamical Systems

Francesco Tedesco

Coordinatore
Prof. Luigi Palopoli

Supervisore
Prof. Alessandro Casavola

DEIS

DEIS- DIPARTIMENTO DI ELETTRONICA, INFORMATICA E SISTEMISTICA
Novembre 2011

Settore Scientifico Disciplinare: ING-INF/04

Abstract

This dissertation presents a class of novel distributed supervision strategies for multi-agent linear systems connected via data networks and subject to coordination constraints. Such a coordination-by-constraint paradigm is characterized by a set of spatially distributed dynamic systems, connected via communication channels, with possibly dynamical coupling amongst them which need to be supervised and coordinated in order to accomplish their overall objective. The basic design philosophy of the Command Governor (CG) set-point management is used here in order to maintain a pre-stabilized system within prescribed constraints.

While in traditional CG schemes the set-point manipulation is undertaken on the basis of the actual measure of the state, in this dissertation it is shown that the CG design problem can be solved also in the case that such an explicit measure is not available by forcing the state evolutions to stay "not too far" from the manifold of feasible steady-states. This approach, referred to as Feed-Forward CG (FF-CG), is a convenient solution to be used in distributed applications where the cost of measuring the overall state and distributing it amongst the agents may be a severe limitation.

Several distributed strategies, based both on CG and FF-CG ideas, will be fully described and analyzed. First, we propose some "non-iterative" schemes in which the agents acting as supervisors communicate once during the decision process. In this respect, a "sequential" distributed strategy in which only one agent at the time is allowed to manipulate its own reference signal is proposed. Such a strategy, although interesting by itself in some applications, will be instrumental to introduce a more effective "parallel" distributed strategy, in which all agents are allowed, under certain conditions, to modify their own reference signals simultaneously. Then an "iterative" procedure, borrowed from the literature, has been here adapted in order to build more efficient distributed schemes which however require larger amount of data exchanges for their implementation. With the aim of evaluating the distributed methods here proposed, several cases of study involving the coordination autonomous vehicles, power networks and water networks management are illustrated.

Contents

1

Introduction

1.1 The Command Governor approach

The Command Governor (CG) approach can be regarded as a constrained supervision strategy which computes its actions by solving on-line a constrained convex optimization problem based on future system predictions, similarly to any Model Predictive Control (MPC) strategy. A CG unit is a nonlinear device which is added to a plant, regulated by a primal controller and separately designed w.r.t. the CG to guarantee stability and tracking performance requirements. The CG main objective is that of modifying, whenever necessary, the reference signal to be supplied to such a pre-compensated system when its unmodified application would lead to constraint violations. This modification is typically achieved according to a receding horizon philosophy consisting of solving on-line at each time instant a constrained optimization problem whose constraints take into account future system predictions. As in MPC schemes, a modified reference sequence is computed at each sampling instant and, according to the receding horizon control (RHC) policy, only the first sample of such a sequence is applied and the entire procedure is repeated at the next time instant.

Command Governor strategies have recently gained interest in the control system literature for their capability to rigorously fulfil on-line set-membership-in-time constraints, allowing the off-line control design phase to be undertaken without considering their presence altogether.

Although a CG exhibits typically degraded performances with respect to the constrained predictive control, its design results simplified. Furthermore, in most cases, MPC computations amount to solve on-line high dimensional programming problems. For this reasons, CG usage can be justified in industrial applications wherein a massive amount of flops per sampling time is not allowed, and/or peripheral units which do not alter the structure of the primal compensated system need to be added to existing standard control stucture (PID like compensator etc.).

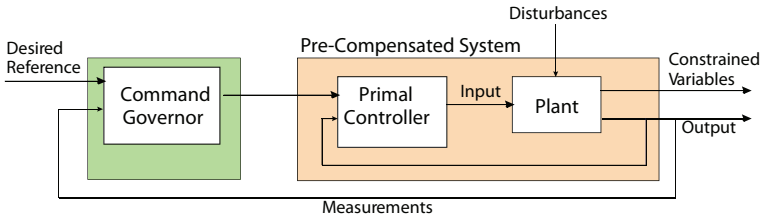

Fig. 1.1. Command Governor Control Scheme

The CG design problem is usually solved by resorting to the theory of the *Maximal Output Admissible Set* introduced in [1] and [2]. Many mature assessments of the CG approach can be found in [1]-[4]. In particular, CG schemes dealing with disturbances were considered in [3] and [5], with model uncertainties in [6] and [5] and with partial state information in [7]. For specific results on CG applied to nonlinear systems see e.g. [8, 9, 10] and [4], for networked master/slave frameworks [11] and for recent results on hybrid Piecewise-Affine systems [12]. Different perspectives on the proposed reference management strategy have been reported in [13] and [14]. For recent applications see [15] and [16].

1.2 Motivations

The reasons of the development of large-scale networked systems are mainly technological and economical. Examples include groups of vehicles, large-scale chemical processes, supply-chain management systems and electrical generators in networked power grids, just to mention a few. The derivation of efficient supervisory control strategies for these systems is of paramount importance for the economic growth, the environment preservation and the quality of life. It is sufficient to estimate, as an example, the economic loss and public transport problems due to blackouts.

For these reasons, their management should be characterized by efficiency, reliability and safety. Large Scale Networked systems are often a composition of a huge number of interacting subsystems. In such a case, a centralized control architecture could not be an adequate choice because the corresponding computational burdens could be unnecessarily prohibitive and the needed communication infrastructures could not be directly implemented. All these reasons justify new research efforts in the development of distributed control and supervising methodologies. In such contexts, the control action is decentralized and performed by a set of control devices, often called *agents*, acting in general on a restricted part of the overall system to be supervised.

Because of its natural capability to handle in a systematic manner hard constraints on inputs and state-related variables, the Command Governor (CG) approach seems to be very suitable for the above tasks. To this end, our approach aims mainly at the development of novel CG methods for supervising

efficiently large-scale distributed systems when a centralized solution is hard to be design or implemented. In particular, several distributed CG variants based on different optimization methods and coordination algorithms will be presented and discussed and their properties fully investigated.

1.3 Decentralized and Distributed Control Architectures

In this section the *distributed* and *decentralized* control concepts will be clarified to put in light the differences. Although a clear distinction is not present in the literature and, even, the two terms are used one in place of the other, in this dissertation we will refer to the definitions adopted in the survey [21] that can be summarized as follows:

- **Decentralized Control** - The agents do not exchange any kind of information and consequently their control decisions are taken independently (see Figure 1.2);

- **Distributed Control** - The agents exchange information (control decision and state measurement) and control decisions can be taken independently if no negotiations are undertaken. However, in some cases there could be several data exchange iterations among the agents before they arrive to an agreement and take decisions (see Figure 1.3).

In this thesis will focus on the more general class of distributed control strategies only and additional notions will be introduced to classify the paradigms and algorithms categories used within this dissertation.

- **Sequential Schemes** - In this case, one agent per time updates its control action according a prefixed order while all others agents keep applying previous applied commands

- **Parallel Schemes** - In this case, agents computed independently their new commands and apply them simultaneously.

Also note that mixed sequential-parallel approaches are possible. In general sequential methods are easier to be implemented but the related performance decrease steadily with the number of agents. On the contrary, parallel methods are usually more efficient but require more complex and numerically demanding coordination algorithms to be implemented.

As mentioned above, distributed and decentralized schemes differ in the possibility fthat a decentralized paradigm share information amongst agents. In particular, such a data exchange can occur in two ways ([21]):

- **Noniterative Schemes** - The information is transmitted/received only once by the agents within the decision time

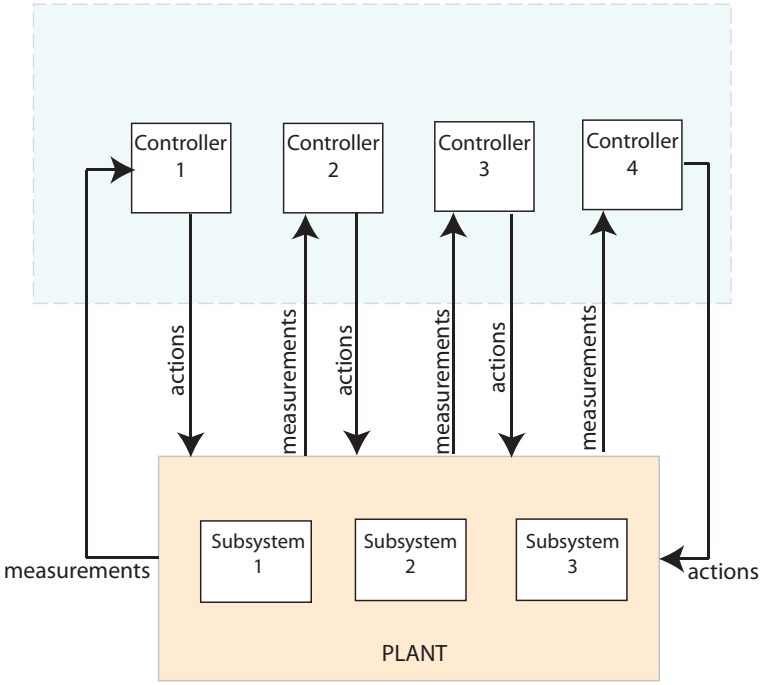

Fig. 1.2. Decentralized Control Schemes

- **Iterative Schemes** - Pieces of information can be transmitted/received several times during the decision time as long an agreement is found

When considering iterative schemes, the large amount of exchanged data is usually exploited by the agents to achieve, by successive refinements, an agreement amongst all agents on the action to be performed on the overall system. In this respect, a further classification is mandatory in order to describe the *behavior* of the agents during the decision process ([21])

- **Non-Cooperating Agents** - Each agent computes its action by minimizing a local performance index

- **Cooperating Agents** - All agents contribute individually to the minimization of a global performance index

In general, as mentioned in [21], as far as the performance is concerned cooperating schemes are better that non-cooperating ones. In particular, from a game theory point of view ([22]), an iterative procedure with agents acting cooperatively leads to a so-called Pareto Optimal solution. On the contrary, when agents have an non-cooperating (egoistic) behaviour, a Nash equilibrium can be reached and usually such a kind of solutions have not any optimality or stability properties ([22]). In this work we will focus our attention on both iter-

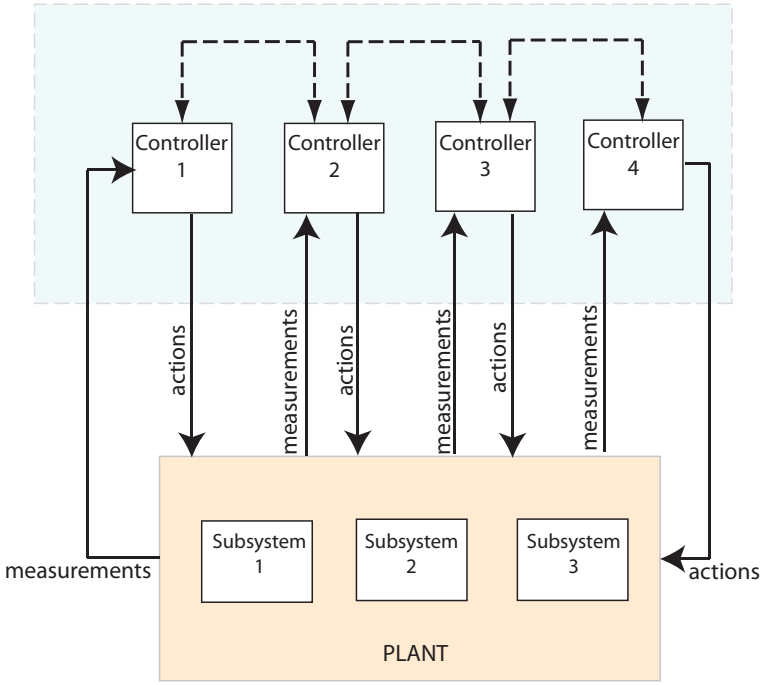

Fig. 1.3. Distributed Control Schemes

ative and non-iterative non-cooperative methods. Both parallel and sequential implementations will be discussed.

1.4 Coordination-by-Constraint Approach

As previously stated, we will focus on the development of distributed supervision strategies based on Command Governor (CG) ideas for multi-agent systems when using a centralized coordination unit is impracticable because of unrealistic or unavailable communication infrastructures. A centralized solution to this problem has been recently proposed in [11] in a quite general context depicted in Fig. 1.4 and it will be referred hereafter as the *Coordination-by-Constraint* approach.

There, the master station is in charge of supervising and coordinating the slave systems via a data network. In particular, r_i, g_i, x_i, y_i and c_i represent respectively: the nominal references, the feasible references, the states, performance-related and coordination-related outputs of the slave systems. In such a context, the supervision task can be expressed as the requirement of satisfying some tracking performance, viz. $y_i \approx r_i$, whereas the coordination task consists of enforcing some pointwise-in-time constraints $c_i \in \mathcal{C}_i$ and/or $f(c_1, c_2, ..., c_N) \in \mathcal{C}$ on each slave system and/or on the overall network evolutions. To this end, the supervisor is in charge of modifying the nominal

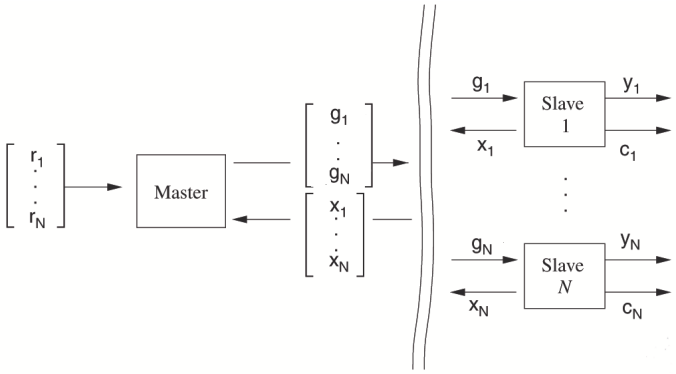

Fig. 1.4. Multi-agent master/slave architectures

references into the feasible ones, when the tracking of the nominal references would lead the evolution of any c_i to violate prefixed constraints.

Here we will consider several distributed supervision strategies for the coordination problem without considering the presence of a single centralized master unit in charge of modifying the nominal set-point. On the contrary, the supervisory task is now distributed amongst many master agents which are assumed to be able to communicate each other and with the slave systems as well.

The scheme here described is depicted in Figure 1.5 where the same constrained supervision and coordination task of Figure 1.4 is addressed but with the noticeable difference that the single subsystem reference sequence r_i is manipulated by a dedicated master agent, which is informed on all other master agents/remote subsystems states x_i and commands g_i via a data network. In addition to the general scheme of Figure 1.5 we will also consider a slight different simplified approach depicted in Figure 1.6 where each agent, in order to accomplish the supervision and coordination task, is not aware on use of any measurement or state estimates of its own and other subsystems. In this case, it is required that the coordination amongst agents would be achieved only by exchanging applied references g_i.

Such an approach represents a particularly interesting solution in large-scale problems. In fact, this solution is based on a recently proposed alternative solution ([23],[24]) to the CG design problem, referred to as the *Feed Forward* CG (FF-CG) approach, that, at the price of some additional conservativeness, is able to accomplish the CG task in the absence of an explicit measure of the state.

The idea behind such an approach is that, if sufficiently smooth transitions in the set-point modifications are acted by the CG unit, it is possible to achieve a certain level confidence on the expected value of the state, even in the absence of an explicit measure, thanks to the asymptotical stability property enjoyed by the system model.

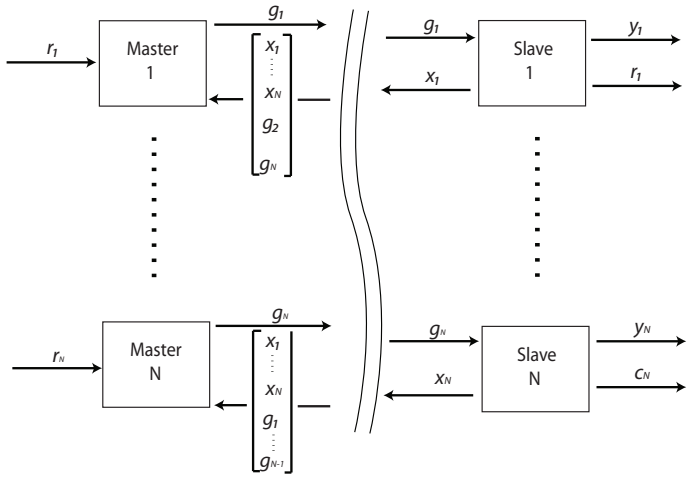

Fig. 1.5. Multi-agent distributed master/slave architectures with state measurement

The FF-CG scheme is an attractive solution for distributed supervision and coordination frameworks because it is not necessary to build the entire aggregate state, or substantially parts of it, known to all agents at each time instant. This results in a lower amount of information on-line exchanged amongst agents.

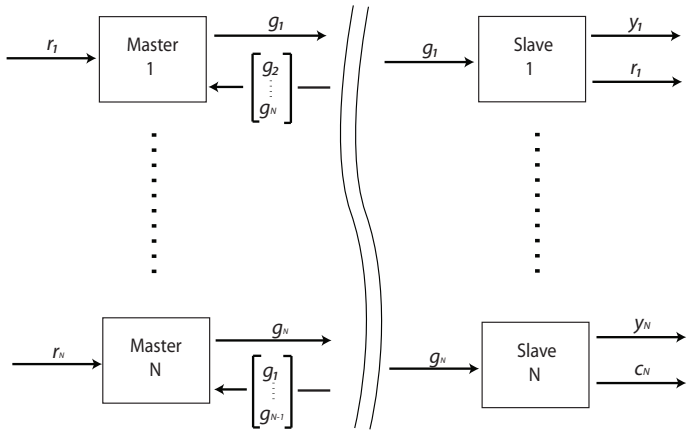

Fig. 1.6. Multi-agent distributed master/slave architectures without state measurement

1.5 Distributed Model Predictive Control: a short overview

Most of the distributed strategies presented in the literature are based on the well known Model Predictive Control Scheme (MPC, see [64, 19]) that has gained a large popularity in the process industry and represents a very powerful method, supported by well established theoretical foundations, capable to handle a huge number of industrial control problems.

Here we chronologically describe significant work in literature dealing with the Distributed MPC. Such a choice is justified due to the CG features which are taken from Predictive Control main concepts, as the resolution of a optimization based control strategy, the RHC philosophy and the direct inclusion of constraints in the optimization problem. Nonetheless, CG solutions have peculiar aspects making them in some sense different from standard MPC approaches. In particular, the CG optimization problems are usually more simpler and the overall on-line strategy is less numerically demanding. Furthermore, it does not seem to exist MPC schemes working without an explicit state (estimate) or system output measurements. This is a crucial difference in that, on the contrary, such GG strategies exist and will be described in next chapters.

In the past years several approaches have been proposed in literature where MPC is presented in a distributed or decentralized fashion to tackle situations described in the section 1.2. From a theoretical point of view, distributed control policies for dynamically coupled systems have been studied in [31, 32] where non-iterative and non-cooperating algorithms have been presented for discrete-time linear time-invariant systems. Discrete-time LTI models are also considered in [33], where an iterative cooperating procedure is described. In particular, the proposed approach ensures the achievement of the global (Pareto) optimum when the iterative procedure converges. Moreover, the method still guarantees closed-loop stability and feasibility if the procedure is stopped at any intermediate iterate. Following the same lines, in [36] a distributed model predictive control scheme for decoupled systems with preliminary stability and feasibility results has been proposed. There, each agent optimizes its local objective which contains also a penalizing term on the coupling neighboring dynamics. Such a problem has been faced also in [35] where vehicles with independent second-order non-linear continuous-time dynamics and coupling cost functions have been considered.

A non-cooperating non-iterative MPC algorithm guaranteeing stability for non-linear continuous-time systems has been presented in [34], where information is transmitted only amongst neighboring subsystems. The stabilizing property of that method proposed strongly depends on a weak dynamic interactions between agents and on a particular constraints considered in the MPC problem that forces the actual input and state sequences to not differ too much from their predicted values. Disturbed subsystems with discrete-time LTI models with independent dynamics but subject to coupling constraints

are considered in [38]. A non-iterative procedure is proposed as solution in which each regulators perform an optimization of a local index but it is aware of the most recent or predicted action of other agents. In the non-iterative and non-cooperating MPC algorithm presented in [39, 40], conditions for a-posteriori stability analysis are introduced which are also valid in the case of data-loss among agents.

1.6 Thesis Overview

1.6.1 Thesis Outline

This thesis is organized in two parts as follows:

Part I

This part starts with the description of the existing centralized standard Command Governor (CG) approach and moves towards the presentation of the FeedForward Command Governor (FF-CG) approach which is an original contribution of this thesis. In particular,

- **Chapter 2** describes the problem, the basic ideas and the main properties of the CG approach. All computational details concerning the solution and the implementation of Command Governor (CG) devices are also reported.
- In **Chapter 3** the FeedForward Command Governor (FF-CG) approach is presented in full details.
- **Chapter 4** addresses the problem of improving the effectiveness of the FF-CG scheme proposed in Chapter 3. A novel less conservative FF-CG solution is discussed which leads to improved performance. Although more efficient, its extension to the distributed case is still under investigation because of higher level of complexity introduced and will be presented elsewhere. Moreover both proposed FF-CG schemes are applied to a position servomechanism and numerical simulations are reported in order to show the effectiveness of the proposed schemes also with respect to the standard state-based CG scheme.

Part II

- In **Chapter 5** two distributed strategies based on the FF-CG approach presented in Chapter 3 will be fully described and analyzed. In particular a sequential distributed strategy (S-FFCG) will be presented in which only one agent at the time is allowed to manipulate its own reference signal. Moreover a complete liveliness analysis will be reported. A geometrical Constraints Qualification (CQ) condition on the prescribed constraints is proposed whose fulfilment avoid deadlock situations and ensure viable solutions. Such a sequential scheme will be instrumental to introduce, a

more effective parallel distributed strategy (P-FFCG), in which all agents are allowed, under certain conditions, to modify their own reference signals simultaneously.
- In **Chapter 6** a customization of the distributed schemes described in Chapter 5 to situations where each agent is provided with the local state measurements is undertaken. Although the resulting schemes (S-CG and P-CG) present a higher level of complexity, a complete characterization is given.
- In **Chapter 7** a distributed iterative optimization method borrowed from [60] is reported for solving the coordination problem of interconnected nonlinear discrete-time dynamic systems with multiple agents. Such a method is shown to be instrumental for the development of CG and FF-CG based distributed iterative methods.
- **Chapter 8** reports a wide comparison of all proposed strategies on an eight-tank cascaded system.

Last chapter summarizes the results of this thesis and gives directions for future research.

1.6.2 Contributions

The main contributions of the research described in this Ph.D thesis with respect to Command Governor and its distributed versions are the following:

- A sensorless CG scheme for linear-time invariant system here referred to as Feed-Forward CG (FF-CG) has been presented in [23, 24].
- A different and more efficient FF-CG scheme has been presented in [42].
- Sequential and parallel non-iterative FF-CG schemes with application to autonomous vehicles have been presented in [43] and [44]. Moreover, a complete analysis of the sequential scheme with the investigation of its liveliness properties can be found in [48] and [49].
- A parallel non-iterative FF-CG scheme with application to a network of water tanks is appeared in [42].
- A sequential non-iterative CG based scheme with application to Load/Frequency control problems in multi-area power systems has been presented in [46].
- A mixed-integer formulation of the CG strategy for collision avoidance problems has been presented in [45].

Part I

State-Based and Feed-Forward Command Governor schemes

2

The Command Governor Design Problem (Standard CG)

In order to make precise statements and comparisons with existing techniques and introduce the relevant notation, this chapter recalls the notions and ideas underlying the standard CG approach for linear systems subject to bounded disturbances (see Figure 2.1). The entire chapter is instrumental to introduce the novel FF-CG scheme in the next chapter.

As mentioned before in the Introduction, a CG is a nonlinear device which is added to a primal compensated control system. The latter, in the absence of the CG, is designed so as to perform satisfactorily in the absence of constraints. Whenever necessary, the CG modifies the input to the primal control system so as to avoid violation of the constraints. Seminal works about this kind of supervision strategy involved in more complicated scenarios characterized by nonlinear systems, uncertain systems, tele-operated systems are represented e.g. by [5] and [17].

This chapter is organized as follows. In Section 2.1 the Feedback CG design problem is formulated whereas its solution is recalled in Section 2.2 where its main properties are pointed out as well. Finally, in Section 2.3 all computational details are reported.

2.1 System Description and Problem Formulation

Let the closed-loop system, consisting of the plant and primal controller of Fig. 3.1, be described by the following discrete-time model

$$\begin{cases} x(t+1) = \Phi x(t) + Gg(t) + G_d d(t) \\ y(t) \quad = H_y x(t) \\ c(t) \quad = H_c x(t) + Lg(t) + L_d d(t) \end{cases} \tag{2.1}$$

where: $t \in \mathbb{Z}$, $x \in \mathbb{R}^n$ is the state vector (which includes the controller states under dynamic regulation), $g \in \mathbb{R}^m$ the manipulable reference vector which, if no constraints (and no CG) were present, would coincide with the desired reference $r \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ the output vector which is required to track r.

The vector $d \in \mathbb{R}^{n_d}$ is a disturbance signal assumed to belong to the convex and compact set $\mathcal{D} \subset \mathbb{R}^{n_d}$:

$$d(t) \in \mathcal{D}, \ \forall t \in \mathbb{Z}_+. \tag{2.2}$$

It is also assumed that $0_{n_d} \in \mathcal{D}$. Finally, $c \in \mathbb{R}^{n_c}$ represents the constrained outputs vector which has to fulfill the set-membership constraints

$$c(t) \in \mathcal{C}, \ \forall t \in \mathbb{Z}_+, \tag{2.3}$$

regardless of any possible admissible disturbance sequence realization $d(\cdot) \in \mathcal{D}$, with $\mathcal{C}$ being a prescribed convex and compact set with non empty interior. It is further assumed that:

A1. The overall system (2.1) is asymptotically stable.
A2. System (2.1) is offset free, i.e. $H_y(I_n - \Phi)^{-1}G = I_m$.

In words, the CG design problem we want to solve is that of determining, at each time step t, a suitably modified reference signal $g(t)$ which is the best approximation of $r(t)$ compatible with the constraints, such that its application never produces constraints violation along the system evolutions induced by the CG commands, i.e. $c(t) \in \mathcal{C}, \ \forall t \in \mathbb{Z}_+, \forall d(\cdot) \in \mathcal{D}$.

2.2 The Standard Command Governor Approach

The classical solution of the above stated CG design problem is achieved by finding, at each time t, a modified command $g(t)$ as a function of the current reference $r(t)$ and measured state $x(t)$

$$g(t) := g(r(t), x(t)) \tag{2.4}$$

such that $g(t)$ is the best approximation of $r(t)$ under the condition $c(t) \in \mathcal{C}, \ \forall d(\cdot) \in \mathcal{D}$.

Let us consider the closed-loop system (2.1)-(2.3) satisfying assumptions **A1-A2**. We will suppose the state be measurable at each time instant and consider the CG design problem as formulated in (2.4). As a first step let us define the following set recursions

$$\mathcal{C}_k := \mathcal{C} \sim \Delta_k, \quad k > 0, \qquad \mathcal{C}_\infty := \mathcal{C} \sim \Delta_\infty \tag{2.5}$$

where

$$\Delta_0 = L_d\mathcal{D}, \quad \Delta_k = \Delta_{k-1} \oplus H_c\Phi^{k-1}G_d\mathcal{D}. \tag{2.6}$$

In (2.5) and (2.6), for given sets $\mathcal{A}, \mathcal{E} \subset \mathbb{R}^n$, $\mathcal{A} \sim \mathcal{E}$ denotes the Pontryagin set difference defined as ([50])

Definition 2.1. ***(Pontryagin-Minkowski set difference) -***

$$\mathcal{A} \sim \mathcal{E} := \{a : \ a + e \in \mathcal{A}, \forall e \in \mathcal{E}\}$$

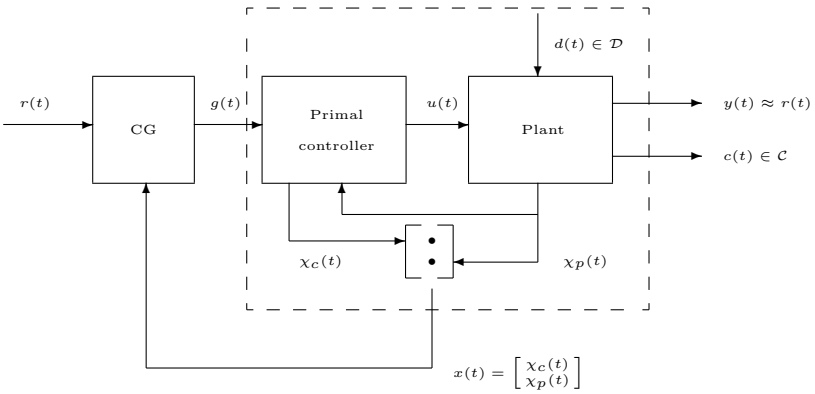

Fig. 2.1. Classical CG structure (based on state measurement)

and $\mathcal{A} \oplus \mathcal{E}$ the Pontryagin-Minkowski set sum, i.e.

Definition 2.2. ***(Pontryagin-Minkowski set sum) -***

$$\mathcal{A} \oplus \mathcal{B} := \{a + b | a \in \mathcal{A}, b \in \mathcal{B}\}.$$

Moreover, Δ_∞ is the Hausdorff limit of the set sequence Δ_k. Because of asymptotical stability of Φ, it can be proved (see [3]) that Δ_∞ is convex and compact and that, provided all $\mathcal{C}_k$ are non-empty, they are also convex and compact, satisfy the nesting condition $\mathcal{C}_k \subset \mathcal{C}_{k-1}$ and make $\mathcal{C}_\infty$ a nonempty convex and compact set.

Let us introduce now the set-valued future predictions (*virtual evolutions*) of the c-variable along the *virtual time* k under a constant *virtual command* $g(k) \equiv g$ and for all possible disturbance sequence realizations $\{d(l) \in \mathcal{D}\}_{l=0}^{k}$ from the initial state x (at virtual time $k = 0$)

$$c(k, x, g, d(\cdot)) = \bigcup_{d(\cdot) \in \mathcal{D}} \left\{ H_c \left(\Phi^k x + \sum_{i=0}^{k-1} \Phi^{k-i-1} (Gg + G_d d(i)) \right) + Lg + L_d d(k) \right\} \tag{2.7}$$

By linearity, the latter can be rewritten as the sum of two terms: $c(k, x, g, d(\cdot)) = \overline{c}(k, x, g) + \tilde{c}(k, d(\cdot))$, where

$$\overline{c}(k, x, g) = H_c \left(\Phi^k x + \sum_{i=0}^{k-1} \Phi^{k-i-1} G g \right) + Lg \tag{2.8}$$

represents the disturbance-free evolution of the c-variable along the *virtual time* k under a constant *virtual command* $g(k) \equiv g$ and initial state x and

$$\tilde{c}(k, d(\cdot)) := \bigcup_{d(\cdot) \in \mathcal{D}} \left\{ \sum_{i=0}^{k-1} H_c \Phi^{k-i-1} G_d d(i) + L_d d(k) \right\} \subseteq \Delta_k, \tag{2.9}$$

the set-valued virtual evolutions due to all possible disturbance sequence realizations, all contained in Δ_k. As a consequence we have

$$\begin{array}{c} \overline{c}(k,x,g) \in \mathcal{C}_k,\ \forall k \in \mathbb{Z}_+ \\ \Downarrow \\ c(k,x,g) = \overline{c}(k,x,g) + \tilde{c}(k,d(\cdot)) \subset \mathcal{C},\ \forall k \in \mathbb{Z}_+ \end{array} \tag{2.10}$$

Thus, constraints fulfilment can be ensured by only considering the disturbance-free evolutions of the system (2.1) and adopting a "worst-case" approach. To this end, let us introduce, for a given sufficiently small scalar $\delta > 0$, the sets:

$$\mathcal{C}^\delta := \mathcal{C}_\infty \sim \mathcal{B}_\delta, \qquad \mathcal{W}^\delta := \{g \in \mathbb{R}^m :\ c_g \in \mathcal{C}^\delta\} \tag{2.11}$$

where $\mathcal{B}_\delta$ is the ball of radius δ centered at the origin and $\mathcal{W}^\delta$, which we assume non-empty, the closed and convex set of all constant commands g whose corresponding disturbance-free equilibrium points

$$c_g := H_c x_g + Lg \tag{2.12}$$

satisfy the constraints with margin δ, being

$$x_g := (I_n - \Phi)^{-1} G g \tag{2.13}$$

the steady state solutions for 2.1.

The main idea behind this CG design problem is to choose at each time step a constant virtual command $g(\cdot) \equiv g$, with $g \in \mathcal{W}^\delta$, such that the corresponding virtual evolutions fulfil the constraints over a semi-infinite horizon $k \in [0,\ \infty)$ and its "distance" from the constant reference of value $r(t)$, valuated by $\|g - r(t)\|^2_\Psi$, $\Psi = \Psi^T > 0$, is minimal. Such a command is applied, a new state is measured and the procedure is repeated at the next time instant, as shown in Figure 2.2. Formally, if we introduce the set of all admissible virtual sequences from initial state x

$$\mathcal{V}(x) = \left\{g \in \mathcal{W}^\delta :\ \bar{c}(k,x,g) \in \mathcal{C}_k,\ \forall k \in \mathcal{Z}_+\right\} \tag{2.14}$$

the CG will choose a command $g(t)$ at each time instant t as the solution of the following constrained optimization problem

$$g(t) = \arg \min_{g \in \mathcal{V}(x(t))} \|g - r(t)\|^2_\Psi \tag{2.15}$$

The following Theorem, proved in [5] and [17], summarizes the properties of the standard CG Algorithm:

Theorem 2.3. - *Let assumptions **A1-A2** be fulfilled. Consider system (2.1)-(2.3) along with the **CG** selection rule (2.15) and let $\mathcal{V}(x(0))$ be non-empty. Then:*

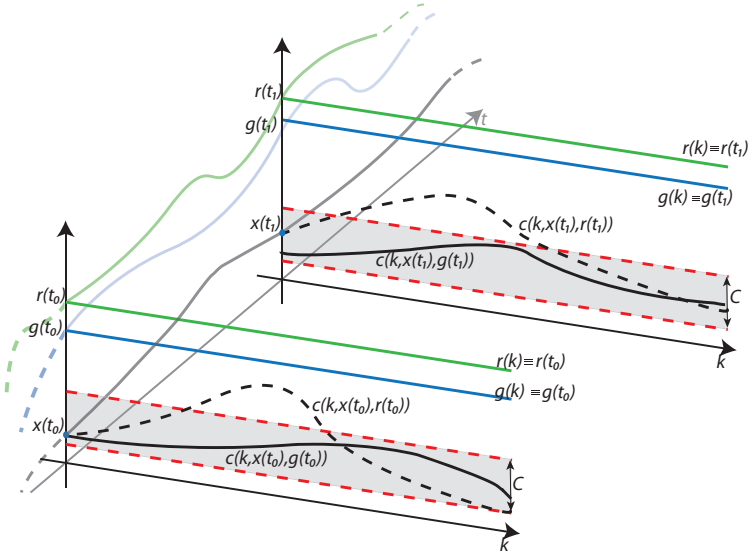

Fig. 2.2. CG evolution: the CG selects at each time step a constant virtual command $g(\cdot) \equiv g$, such that the corresponding virtual evolutions fulfil the constraints over a semi-infinite horizon $k \in [0,\ \infty)$ and its "distance" from the constant reference of value $r(t)$, is as small as possible. The reference is applied at time t and the procedure is reiterated at time instant $t+1$.

1. *At each decision time t, the minimizer in (2.15) uniquely exists and can be obtained by solving a convex constrained optimization problem;*
2. *The set $\mathcal{V}(x), \forall x \in \mathbb{R}^n$, is finitely determined, viz. there exists an a priori known integer k_0 (see [1]) such that if $\bar{c}(k,x,w) \in \mathcal{C}_k$, $k \in \{0,1,\dots k_0\}$, then $\bar{c}(k,x,w) \in \mathcal{C}_k\ \forall k \in \mathbb{Z}_+$.*
3. *The system supervised by the **CG** never violates the constraints, i.e. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$ regardless of any possible admissible disturbance realization $d(\cdot) \in \mathcal{D}$;*
4. *The overall system is asymptotically stable. Moreover, whenever $r(t) \equiv r$, with r a constant set-point, the sequence of $g(t)$'s converges in finite time either to r or to its best admissible steady-state approximation $\hat{r}$:*

$$\exists t_s > 0 \ \ t.c.\ \ g(t) = \hat{r} := \arg\min_{g \in \mathcal{W}_\delta} \|g - r\|_\Psi^2, \forall t \geq t_s \tag{2.16}$$

and

$$\lim_{t\to\infty} \hat{x}(t) = x_{\hat{r}}, \qquad \lim_{t\to\infty} \hat{y}(t) = y_{\hat{r}} = \hat{r}, \qquad \lim_{t\to\infty} \hat{c}(t) = c_{\hat{r}}. \tag{2.17}$$

2.3 Computational details

This section is added for the sake of completeness and recalls material presented in [51].

2.3.1 Linear constraints: The disturbance-free case

In this section we will present all computational details for the design and the implementation of the above CG strategy in the disturbance-free case $d(t) = 0,\ \forall t$. Because linear constraints for $c(t) \in \mathcal{C}$ are assumed, they can be represented as

$$c(t) \in \mathcal{C} \Longleftrightarrow \mathcal{C} := \{c \in \mathbb{R}^{n_c} : Tc \leq q\} \tag{2.18}$$

$$T = \begin{bmatrix} T_1^T \\ T_2^T \\ \cdots \\ T_z^T \end{bmatrix} \in \mathbb{R}^{z \times n_c},\ q = \begin{bmatrix} q_1 \\ q_2 \\ \cdots \\ q_z \end{bmatrix} \in \mathbb{R}^z,\ z \geq n_c, \tag{2.19}$$

$$\text{rank } (T) = n_c \tag{2.20}$$

$$\tag{2.21}$$

with T_i^T and q_i denoting respectively the i-th rows of T and q. Notice that the number z of rows of T is in general larger than the number of columns and the rank condition on T ensures that $\mathcal{C}$ is a bounded set. Closure and convexity trivially follow.

In the disturbance-free case all sets $\mathcal{C}_k$ defined in (2.5) coincide with $\mathcal{C}$. Therefore, the set $\mathcal{C}_\delta$ is given by

$$\mathcal{C}_\delta = \mathcal{C} \sim \mathcal{B}_\delta = \{c \in \mathbb{R}^{n_c} \ :\ T_i^T c \leq q_i - \delta\sqrt{T_i^T T_i},\ i = 1, ..., z\}$$

where the terms $\delta\sqrt{T_i^T T_i}$ is the support function of the ball $\mathcal{B}_\delta$. See the Appendix for the computation of the support functions of use for the computation of the P-difference. Then, the $\mathcal{W}_\delta$ set can be directly computed and results

$$\mathcal{W}_\delta = \{w \in \mathbb{R}^m :\ \bar{c}_w \in \mathcal{C}_\delta\} \tag{2.22}$$

$$= \{w \in \mathbb{R}^m :\ T\left(H_c(I - \Phi)^{-1}G + L\right) w \leq q - \delta[\sqrt{T_i^T T_i}]\} \tag{2.23}$$

where

$$[\sqrt{T_i^T T_i}] := \begin{bmatrix} \sqrt{T_1^T T_1} \\ \sqrt{T_2^T T_2} \\ .... \\ \sqrt{T_z^T T_z} \end{bmatrix} \tag{2.24}$$

It remains to characterize the set $\mathcal{V}(x)$ which, in the disturbance-free case, is defined as

$$\mathcal{V}(x) = \{w \in \mathcal{W}^{\delta} : \ \bar{c}(k, x, w) \in \mathcal{C}, \ k = 0, ..., k_0\} \tag{2.25}$$

where the integer k_0 will be specified in a while. It is also easy to understand that

$$\begin{aligned} \bar{x}(k, x, g) &= \Phi^k x + \left(\sum_{i=0}^{k-1} \Phi^i G \right) g \\ &= \Phi^k x + R_k^x g \qquad (2.26) \\ \bar{c}(k, x, g) &= H_c \bar{x}(k, x, g) + Lg \\ &= H_c \Phi^k x + (H_c R_k + L)\, g \\ &= H_c \Phi^k x + R_k^c g \qquad (2.27) \end{aligned}$$

and hence

$$\mathcal{V}(x) = \{w \in \mathcal{W}^{\delta} : \ TH_c \Phi^k x + TR_k^c g \leq q, \ k = 0, ..., k_0\} \tag{2.28}$$

Finally, the CG action computation consists of solving the following QP optimization problem

$$g(t) = \arg\min_g (g - r(t))^T \Psi (g - r(t)) \tag{2.29}$$

$$\text{subject to} \tag{2.30}$$

$$TH_c \Phi^k x(t) + TR_k^c g \leq q, \ k = 0, ..., k_0 \tag{2.31}$$

$$T\left(H_c (I - \Phi)^{-1} G + L\right) g \leq q - \delta[\sqrt{T_i^T T_i}] \tag{2.32}$$

In order to compute the constraint horizon k_0 define

$$\begin{gathered} G_k(j) := \max_{x \in \mathbb{R}^n, g \in \mathcal{W}_{\delta}} \quad T_j^T \bar{c}(k, x, g) - q_j \qquad (2.33) \\ \text{subject to} \\ T_j^T \bar{c}(i, x, g) \leq q_j, \quad i = 0, ..., k-1 \end{gathered}$$

Then, the following procedure can be used k_0:

Algorithm 2.3.1 (k_0 computation)

1. $k = 1$
2. *if* $G_k(j) \leq 0, \ \forall j = 1, ..., z$ *then* $k_0 = k$ *and stop*;
3. $k = k + 1$, *goto 2*

which is ensured to converge in a finite number of steps.

2.3.2 Linear constraints: The disturbance acting case

We assume here that the disturbances are bounded and characterizable as follows:

$$d(t) \in \mathcal{D} \Longleftrightarrow \mathcal{D} := \{d \in \mathbb{R}^{n_d} \ : \ Ud \leq h\} \tag{2.34}$$

$$U = \begin{bmatrix} U_1^T \\ U_2^T \\ \cdots \\ U_u^T \end{bmatrix} \in \mathbb{R}^{u \times n_d}, \quad h = \begin{bmatrix} h_1 \\ h_2 \\ \cdots \\ h_u \end{bmatrix} \in \mathbb{R}^u, \tag{2.35}$$

$$u \geq n_d, \ \text{rank}\ U = n_d, \ h_i \geq 0 \tag{2.36}$$

with U_i^T and h_i denoting respectively the i-th rows of U and h. Notice that the number u of rows of U is in general larger than the number of columns. The rank condition on U ensures that $\mathcal{D}$ is a bounded set whereas $h_i \geq 0$ ensures that $0_{n_d} \in \mathcal{D}$. Closure and convexity trivially follow.

The set inclusions (2.5-2.6) can be computed as follows

$$\begin{aligned} \mathcal{C}_0 &= \mathcal{C} \sim L_d\mathcal{D} = \{c \in \mathbb{R}^{n_c} : T_i^T c \leq q_i - \sup_{d\in\mathcal{D}} T_i^T L_d d, \ i = 1, ..., z\} \\ &= \{c \in \mathbb{R}^{n_c} : T_i^T c \leq q_i^0, \ i = 1, ..., z\} \end{aligned} \tag{2.37}$$

$$\begin{aligned} \mathcal{C}_1 &= \mathcal{C}_0 \sim H_c G_d \mathcal{D} = \{c \in \mathbb{R}^{n_c} : T_i^T c \leq q_i^0 - \sup_{d\in\mathcal{D}} T_i^T H_c G_d d, \ i = 1, ..., z\} \\ &= \{c \in \mathbb{R}^{n_c} : T_i^T c \leq q_i^1, \ i = 1, ..., z\} \end{aligned} \tag{2.38}$$

$$\cdots$$

$$\begin{aligned} \mathcal{C}_k &= \mathcal{C}_{k-1} \sim H_c \Phi^{k-1} G_d \mathcal{D} = \{c \in \mathbb{R}^{n_c} : T_i^T c \leq q_i^{k-1} \\ &\quad - \sup_{d\in\mathcal{D}} T_i^T H_c \Phi^{k-1} G_d d, \ i = 1, ..., z\} = \{c \in \mathbb{R}^{n_c} : T_i^T c \leq q_i^k, \ i = 1, ..., z\} \end{aligned} \tag{2.39}$$

In order to compute $\mathcal{W}_\delta$ one should evaluate $\mathcal{C}_\delta$ by computing $\mathcal{C}_\infty$ first. To this end, a convenient approximation of $\mathcal{C}_\infty$ can becomputed by looking for a convenient set $\mathcal{C}_\infty(\varepsilon)$ such that

$$\mathcal{C}_\infty(\varepsilon) \subset \mathcal{C}_\infty \subset \mathcal{C}_\infty(\varepsilon) + \mathcal{B}_\varepsilon \tag{2.40}$$

Such a set, unlike $\mathcal{C}_\infty$, is computable in a finite number of steps. In fact, it can be shown that

$$\mathcal{C}_\infty = \mathcal{C}_k \sim \left(\sum_{i=k}^{\infty} H_c \Phi^i G_d \mathcal{D} \right) \tag{2.41}$$

Moreover, because of stability of Φ, there exist two constants $M > 0$ and $\lambda \in (0, 1)$ such that

$$\|\Phi^k\| \leq M\lambda^k$$

Boundedness of $\mathcal{D}$ also implies that there exists finite

$$\bar{d} := \max_{d\in\mathcal{D}} \|d\|_2 \tag{2.42}$$

The above facts are enough to ensure that, once chosen the desired accuracy ε, there exists an index k_ε such that

$$\sum_{i=k_\varepsilon}^{\infty} H_c \Phi^i G_d \mathcal{D} \subset \mathcal{B}_\varepsilon \tag{2.43}$$

In fact, it suffices that

$$\bar{d}\bar{\sigma}(H_c)\bar{\sigma}(G_d)M \sum_{i=k_\varepsilon}^{\infty} \lambda^i \leq \varepsilon \tag{2.44}$$

which, after direct steps, gives rise to

$$k_\varepsilon = \frac{\log \varepsilon + \log(1-\lambda) - \log[\bar{\sigma}(H_c)\bar{\sigma}(G_d)M\bar{d}]}{\log \lambda} \tag{2.45}$$

Then, the desired approximation can be computed as

$$\mathcal{C}_\infty(\varepsilon) = \mathcal{C}_{k_\varepsilon} \sim \mathcal{B}_\varepsilon \tag{2.46}$$

and, in turn,

$$\begin{aligned}
\mathcal{C}_\delta(\varepsilon) &= (\mathcal{C}_{k_\varepsilon} \sim \mathcal{B}_\varepsilon) \sim \mathcal{B}_\delta \\
&= \{c \in \mathbb{R}^{n_c} \ : \ T_i^T c \leq q^{k_\varepsilon} - (\varepsilon+\delta)[\sqrt{T_i^T T_i}]\} && (2.47) \\
\mathcal{W}_\delta &= \{g \in \mathbb{R}^m : \ \bar{c}_w \in \mathcal{C}_\infty^\delta(\varepsilon)\} && (2.48) \\
&= \{w \in \mathbb{R}^m : T\left(H_c(I-\Phi)^{-1}G+L\right) w \leq q^{k_\varepsilon} - (\varepsilon+\delta)[\sqrt{T_i^T T_i}]\} && (2.49)
\end{aligned}$$

Of course, emptiness of $\mathcal{C}_\infty^\delta(\varepsilon)$ or $\mathcal{W}_\delta$, that is at least one component of $q^{k_\varepsilon} - (\varepsilon+\delta)[\sqrt{T_i^T T_i}]$ is strictly negative, implies that the problem is not solvable.

In this case, $\mathcal{V}(x)$ can be characterized as

$$\mathcal{V}(x) = \{g \in \mathcal{W}^\delta : \ \bar{c}(k,x,g) \in \mathcal{C}_k, \ k = 0, ..., k_0\} \tag{2.50}$$

where the integer k_0 will be specified in while. Because we work with disturbance-free predictions, we can use (2.26) and (2.27) also in this case. Hence

$$\mathcal{V}(x) = \{g \in \mathcal{W}^\delta : \ TH_c\Phi^k x + TR_k^c g \leq q^k, \ k = 0, ..., k_0\} \tag{2.51}$$

Finally, the CG action computation consists of solving the following QP optimization problem

$$\begin{aligned}
g(t) &= \min_g (g - r(t))^T \Psi (g - r(t)) && (2.52) \\
&\text{subject to} && (2.53) \\
TH_c\Phi^k x(t) + TR_k^c g &\leq q^k, \ k = 0, ..., k_0 \\
T\left(H_c(I-\Phi)^{-1}G + L\right) g &\leq q^{k_\varepsilon} - (\varepsilon+\delta)[\sqrt{T_i^T T_i}]
\end{aligned}$$

The computation of the constraint horizon k_0 can be accomplished via the following procedure. Define

$$G_k(j) := \max_{x\in\mathbb{R}^n, g\in\mathcal{W}_\delta} \quad T_j^T \bar{c}(k,x,g) - q_j^k \qquad (2.54)$$
$$\text{subject to}$$
$$T_j^T \bar{c}(i,x,g) \le q_j^i, \quad i = 0,...,k-1$$

Then, the following procedure can be proved to converge to k_0:

Algorithm 2.3.2 (k_0 computation)

1. $k = 1$
2. *if* $G_k(j) \le 0,\ \forall j = 1,...,z$ *then* $\bar{k} = k$ *and stop*;
3. $k = k + 1$, *goto 2*

2.4 Conclusion

In this chapter the Standard Command Governor design problem and its solution have been recalled in order to make clear what the starting point was to the novel techniques introduced in the next chapters. The key points of the standard CG approach are as follows: 1) the use of an inner-loop aimed at compensating the given plant and ensuring desirable control properties when the constraints are not active (linear regimes); 2) the design of an outer-loop involving the CG unit in charge to enforce input and state-related constraints by modifying, whenever necessary on the basis of the actual state, the reference signal.

The concept of "virtual" command sequence has been shown to be instrumental for synthesizing a CG having the required properties. This has been achieved by: first, parameterizing the virtual command sequence by a suitable vector of free parameters; second, choosing at each sampling time such a free parameter vector as the one minimizing a constrained quadratic selection index.

2.5 Appendix

2.5.1 P-difference

Given two sets $\mathcal{A}$ and $\mathcal{B}$ of $\mathbb{R}^n$ we define the P-difference between $\mathcal{A}$ and $\mathcal{B}$, and denotes it as $\mathcal{A} \sim \mathcal{B}$, the following set

$$\mathcal{A} \sim \mathcal{B} := \{x \in \mathbb{R}^n \ ;\ x + b \in \mathcal{A},\ \forall b \in \mathcal{B}\}$$

Whenever the set $\mathcal{A} \sim \mathcal{B}$ is not empty, it enjoys the following properties.

Properties

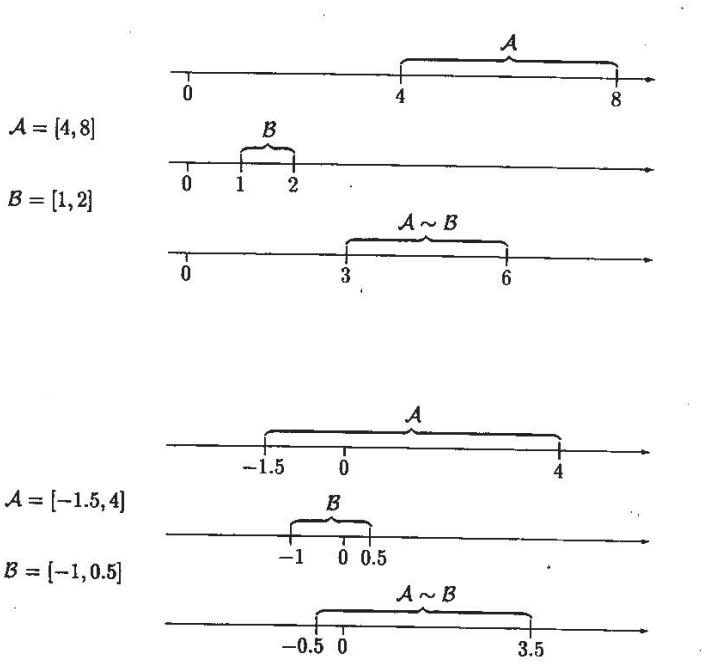

Fig. 2.3. P-difference of intervals

(*i*) $\mathcal{A} \sim \mathcal{B} = \cap_{b \in \mathcal{B}} (\mathcal{A} - b)$

(*ii*) $(\mathcal{A} \sim \mathcal{B}) \oplus \mathcal{B} \subset \mathcal{A}$

(*iii*) If $0_n \in \mathcal{B}$, then $\mathcal{A} \sim \mathcal{B} \subset \mathcal{A}$

(*iv*) Let $\mathcal{A} = \mathcal{A}_1 \cap \mathcal{A}_2$, then $\mathcal{A} \sim \mathcal{B} = (\mathcal{A}_1 \sim \mathcal{B}) \cap (\mathcal{A}_2 \sim \mathcal{B})$

(*v*) Let $\mathcal{B} = \mathcal{B}_1 \oplus \mathcal{B}_2$, then $\mathcal{A} \sim \mathcal{B} = (\mathcal{A} \sim \mathcal{B}_1) \sim \mathcal{B}_2$

(*vi*) If $\mathcal{A}$ is bounded, closed, convex, then $\mathcal{A} \sim \mathcal{B}$ is bounded, closed, convex

When $\mathcal{A}$ and $\mathcal{B}$ are intervals, subset of $\mathbb{R}$, the above property (i) allows directly the determination of the interval $\mathcal{A} \sim \mathcal{B}$. When $\mathcal{A}$ is a subset of $\mathbb{R}^n$, the determination of $\mathcal{A} \sim \mathcal{B}$ can be done quite directly when $\mathcal{A}$ is a polyhedral set

$$\mathcal{A} = \{a \in \mathbb{R}^n \ : \ \alpha_i' a \leq b_i, \ i = 1, ..., N\} \tag{2.55}$$

To this end, we need to consider the *support function* of a convex and compact set $\mathcal{B}$

Def. - Let $\mathcal{B} \subset \mathbb{R}^n$. The support function of $\mathcal{B}$ evaluated in $\eta \in \mathbb{R}^n$ is defined as

$$h_{\mathcal{B}}(\eta) = \sup_{b \in \mathcal{B}} \eta' b$$

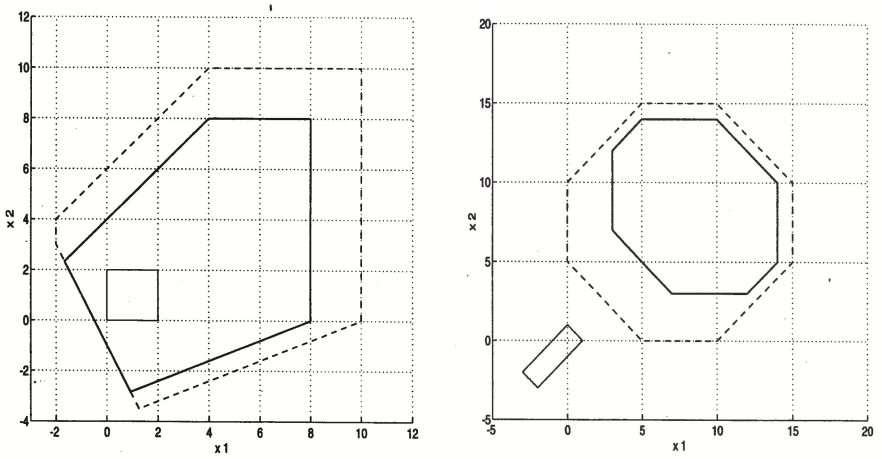

Fig. 2.4. P-difference of polyhedral: $\mathcal{A}$ (dash), $\mathcal{B}$ (thin continuous), $\mathcal{A} \sim \mathcal{B}$ (thick continuous)

Then, if $\mathcal{A} \subset \mathbb{R}^n$ is as in (2.55), one has that

$$\mathcal{A} \sim \mathcal{B} = \{a \in \mathbb{R}^n \ : \ \alpha_i' a \leq b_i - h_{\mathcal{B}}(\alpha_i), \ i = 1, ..., N\} \tag{2.56}$$

The numerical evaluation of the support function requires the use of LP programming when $\mathcal{B}$ is a polyhedral set. When, $\mathcal{B}$ is an ellipsoidal set, viz. $\mathcal{B} := \{b \in \mathbb{R}^n \ : \ b'Qb \leq 1\}$, with $Q = Q' > 0$, it can be shown that $h_{\mathcal{B}}(\eta) = \sqrt{\eta' Q^{-1} \eta}$. Observe further that the evaluation of $\mathcal{A} \sim M\mathcal{B}$, whenever M is a matrix of appropriate dimensions, is any longer more complex than evaluating $\mathcal{A} \sim \mathcal{B}$. In fact, one has that

$$h_{M\mathcal{B}}(\eta) = h_{\mathcal{B}}(M'\eta)$$

In next Figure 2.5.1, the P-difference for the following sets

$$\text{(Left)} \ \ \mathcal{A} = \begin{cases} -x_1 + x_2 \leq 6 \\ \frac{2}{5} x_1 - x_2 \leq 4 \\ -2x_1 - x_2 \leq 1 \\ -2 \leq x_1 \leq 10 \\ x_2 \leq 10 \end{cases} \quad \mathcal{B} = \begin{cases} 0 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq 2 \end{cases}$$

$$\text{(Right)} \ \ \mathcal{A} = \begin{cases} -10 \leq -x_1 + x_2 \leq 10 \\ 5 \leq x_1 + x_2 \leq 25 \\ 0 \leq x_1 \leq 15 \\ 0 \leq x_2 \leq 15 \end{cases} \quad \mathcal{B} = \begin{cases} -5 \leq x_1 + x_2 \leq 1 \\ -1 \leq -x_1 + x_2 \leq 1 \end{cases}$$

are reported.

2.5.2 Convergence results on sequences of compact sets

Consider a family of compact sets $\mathcal{F}_k$ defined as

$$\mathcal{F}_0 = \{0_n\} \tag{2.57}$$

$$\mathcal{F}_k = \sum_{i=0}^{k-1} \Phi^i G \mathcal{D},\ k \geq 1 \tag{2.58}$$

where Φ is an asymptotically stable matrix and $\mathcal{D}$ a compact set containing 0_d. Then, in the topology induced by the Hausedorff's metrics[1] it is possible to show that

Theorem - Let $\mathcal{D}$ be compact and Φ asymptotically stable. Then, there exists a compact set $\mathcal{F}$ such that

$$(i)\ \ \mathcal{F}_k \subset \mathcal{F}, \forall k \in \mathbb{Z}_+$$
$$(ii)\ \ \forall \varepsilon > 0, \exists k_\varepsilon\ :\ \mathcal{F} \subset \mathcal{F}_k + \mathcal{B}_\varepsilon.$$

[1] Hausedorff's metrics - Let ρ be a whatever bounded metrics in $\mathbb{R}^n$ and $\mathcal{R}$ the set of all compact subsets of $\mathbb{R}^n$. Let $\mathcal{A}$ and $\mathcal{B}$ arbitrary elements of $\mathcal{R}$ and consider the function $d_{\mathcal{H}}\ :\ \mathcal{R} \times \mathcal{R} \to \mathbb{R}$ defined as

$$d_{\mathcal{A}}(\mathcal{B}) := \sup_{x \in \mathcal{B}} \rho(A, x),\ \rho(A, x) := \inf_{y \in \mathcal{A}} \rho(x, y)$$
$$d_{\mathcal{H}}(\mathcal{A}, \mathcal{B}) := \max\{d_{\mathcal{A}}(\mathcal{B}), d_{\mathcal{B}}(\mathcal{A})\}$$

3

The Feed-Forward Command Governor Approach

This chapter presents a novel class of Command Governor (CG) strategies for input and state-related constrained discrete-time LTI systems subject to bounded disturbances in the absence of explicit state or output measurements.

While in traditional CG schemes the set-point manipulation is undertaken on the basis of either the actual measure of the state or its suitable estimation, it is shown here that the CG design problem can be solved, with limited performance degradation and with similar properties, also in the case that such an explicit measure is not available.

This approach, which will be referred to as the *Feed-Forward CG* (FF-CG) approach, may be a convenient alternative CG solution in all situations whereby the cost of measuring the state may be a severe limitation, e.g. in distributed or decentralized applications. This chapter and the preceding are mandatory for the reader in order to understand distributed implementations of the CG scheme. The Chapter is organized as follows. In Section 3.1, the FF-CG scheme is introduced and its main properties investigated. In Section 3.2, some of the technical details underlying the derivation of the FF-CG scheme are discussed for the sake of clarity. Finally, in Section 3.3 numerical details are presented and applied to a simple numerical example in order to give a complete picture of the numerical complexity underlying the FF-CG implementation.

3.1 Problem Formulation

In this section the goal is to present a different approach to the CG design problem for systems of the form 2.1-2.3 which satisfy also assumptions **A1-A2** which enables us to deal with the case that no state measurements are available to the CG unit. To this end, a FF-CG action of the form

$$g(t) = \underline{g}(r(t), g(t-1), \rho(t)) \tag{3.1}$$

will be considered where $g(t-1)$ is the last applied command, $\rho(t)$ is a scalar whose meaning will be clarified later and $g(t)$ is the best approximation of $r(t)$, to be computed so as to ensure constraints satisfaction along the system virtual evolutions.

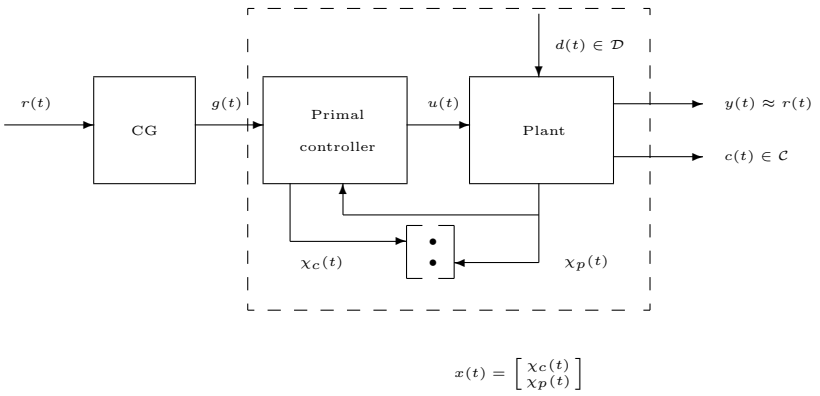

Fig. 3.1. Feedforward CG structure (without state measurement)

The idea underlying such an approach is that any admissible variation of the manipulated reference $g(\cdot)$ always produces a guaranteed bounded perturbation on the actual constraints vector c around a suitable feasible steady-state value. Such a property can be ensured by properly bounding the CG action variations by means of the following technical expedients depicted in Figure 3.1:

1. the computation of a new FF-CG action $g(t)$ is performed every τ steps, being τ a suitable integer to be determined, rather than at each time instant t as in the standard CG approach. Moreover, each new FF-CG command is applied for exactly τ steps;
2. the displacement between the new FF-CG command $g(t)$ and the previous one $g(t-\tau)$ is explicitly bounded during the FF-CG computation, i.e. $g(t) - g(t-\tau) \in \Delta\mathcal{G}(g(t-\tau), \rho(t-\tau))$,

where the integer $\tau > 0$ and the closed and convex set $\Delta\mathcal{G}(g, \rho) \subset \mathbb{R}^m$ are determined from the outset whereas $\rho(t)$ is a time-varying scalar parameter selected on-line. The definitions and meanings of τ, $\Delta\mathcal{G}(g, \rho)$ and $\rho(\cdot)$ will be introduced and discussed later.

The first difference with the standard CG design method is relies on the disturbance effects and their management by the proposed scheme. In fact, while in the standard CG approach the *virtual evolutions* $\bar{c}(k, x, g)$ in (2.8) are computed on the basis of the measured state $x(t)$ and the effect of disturbances

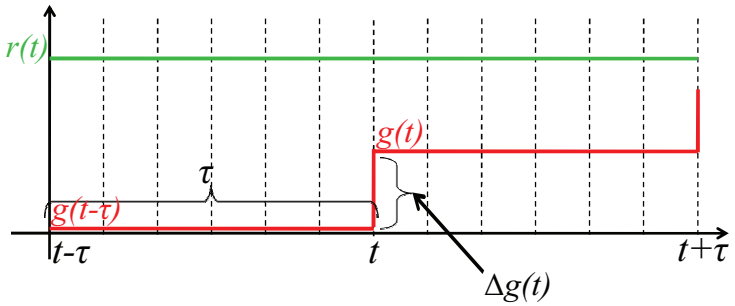

Fig. 3.2. Graphical representation of technical expedients adopted.

on the predictions can be exactly characterized along the prediction horizon, such an information is not here available and the disturbances need to be taken into account as if they were acting on each element of the sequence $\bar{c}(k, x, g)$ starting from a remote time instant. In order to deal with this problem, denote

$$x(t) = \hat{x}(t) + \tilde{x}(t) \tag{3.2}$$

where $\hat{x}$ is the disturbance-free component of the state and $\tilde{x}$ depending only on the disturbances. Then, $\bar{c}(k, x, g)$ can be rewritten as the sum of three terms

$$c(k, x, g, d(\cdot)) = \bar{c}(k, \hat{x}, g) + \tilde{c}(k, d(\cdot)) + H_c \Phi^k \tilde{x}. \tag{3.3}$$

Note that if no state measure is available, $\tilde{x}$ is not identified as a single vector but is a set instead depending on all possible disturbance history $d(\cdot) \in \mathcal{D}$. It can be characterized as

$$H_c \Phi^k \tilde{x} = \bigcup_{d(\cdot)\in\mathcal{D}} \left\{ H_c \Phi^k \sum_{i=-\infty}^{0} \Phi^{-i} G_d d(i-1) \right\} \subseteq \sum_{i=k}^{\infty} H_c \Phi^i G_d \mathcal{D}. \tag{3.4}$$

By recalling equations (2.9), we know that

$$\tilde{c}(k, d(\cdot)) \subseteq L_d \mathcal{D} \oplus \sum_{i=0}^{k-1} H_c \Phi^i G_d \mathcal{D}$$

which implies that the following inclusion holds true

$$\tilde{c}(k, d(\cdot)) + H_c \Phi^k \tilde{x} \subseteq \Delta_\infty. \tag{3.5}$$

It finally results that $\bar{c}(k,\hat{x},g) \in \mathcal{C}_\infty,\ \forall k \in \mathbb{Z}_+ \Longrightarrow c(k,x,g,d(\cdot)) \subset \mathcal{C},\ \forall k \in \mathbb{Z}_+$. As a consequence, in the present context, the constraints fulfilment is obtained by ensuring that

$$\bar{c}(k,\hat{x},g) \in \mathcal{C}_\infty, \forall k \in \mathbb{Z}_+ . \tag{3.6}$$

Such a condition can be further simplified by manipulating the *virtual evolutions* $\bar{c}(k,x,g)$ as follows

$$\bar{c}(k,\hat{x},g) = c_g + H_c \Phi^k (\hat{x} - x_g) \tag{3.7}$$

where x_g and c_g represent the steady-state values of the state and constrained vector already defined in (2.12)-(2.13) and $H_c\Phi^k(\hat{x}-x_g)$ the constrained vector transient evolution. Like in the standard CG solution, we will restrict our

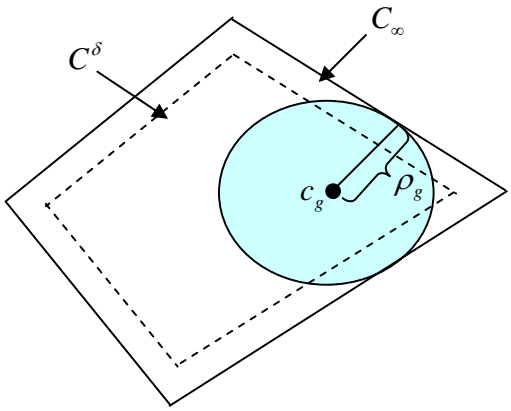

Fig. 3.3. Geometrical representation of condition (3.8) for $c \in \mathbb{R}^2$

attention to virtual commands g contained in $\mathcal{W}_\delta$. Then, the steady-state component of the virtual evolutions will always belong to $\mathcal{C}_\delta$, viz. $c_g \in \mathcal{C}_\delta$. As depicted in Figure 3.3, a sufficient condition to ensure that the constraints will be satisfied, although in a quite arbitrary and conservative way, is that of ensuring that the vanishing transient component of $\bar{c}$ is confined into a ball of radius ρ_g

$$\|H_c \Phi^k (\hat{x} - x_g)\| \leq \rho_g, \forall k \geq 0 \tag{3.8}$$

where ρ_g represents the minimum distance between c_g and the border of $\mathcal{C}_\infty$. Such a quantity can be computed by solving the following simple optimization problem

$$\begin{array}{c} \rho_g := \arg\max_\rho \rho \\ \text{subject to } \mathcal{B}_\rho(c_g) \subseteq \mathcal{C}_\infty. \end{array} \tag{3.9}$$

where $\mathcal{B}_\rho(c_g)$ represents the ball of radius ρ centered in c_g. Notice that, by construction, $\rho_g \geq \delta, \forall g \in \mathcal{W}_\delta$. Details on the solution of (3.9) are reported

in next Subsection 3.2.2. Then, the key idea behind the construction of an effective FF-CG algorithm is as follows: the FF-CG modifies command signal only every τ steps under the assumption that, at time $t-\tau$, a command $g(t-\tau) \in \mathcal{W}_\delta$ has been computed such that the transient component of $\bar{c}(k, \hat{x}(t-\tau), g(t-\tau))$ is confined in a ball of known radius $\rho(t-\tau)$ (which is also contained into the ball of radius $\rho_{g(t-\tau)}$). This makes the condition (3.8) to hold true, i.e.

$$\| H_c \Phi^k (\hat{x}(t-\tau) - x_{g(t-\tau)}) \| \leq \rho(t-\tau) \leq \rho_{g(t-\tau)}, \ \forall k \geq 0. \tag{3.10}$$

If we build an algorithm which, on the basis of the above information, were able to select at time t a new command $g(t) \in \mathcal{W}_\delta$ and a scalar $\rho(t) \geq 0$ such that the transient components of $\bar{c}(k, \hat{x}(t), g(t))$ were confined within a ball of radius $\rho(t) \leq \rho_{g(t)}$, then the constraints would be again satisfied, i.e.

$$\| H_c \Phi^k (\hat{x}(t) - x_{g(t)}) \| \leq \rho(t) \leq \rho_{g(t)}, \forall k \geq 0 \tag{3.11}$$

and, by induction, a FF-CG command $g(t)$ would be proved to exist at each time instant $t \in \mathbb{Z}_+$ provided that it would exist at time $t = 0$. It is worth noticing that $\rho(t)$ and ρ_g represent two distinct terms in the sense that $\rho(t)$ represents the actual "radius" of a ball that contains the future transient evolutions of $\bar{c}$, while ρ_g is the maximum value (depending on g) that such a radius may assume without violating the constraints. It is important to separate the two concepts because in many cases $\rho(t) << \rho_g$.

The term $\| H_c \Phi^k (\hat{x}(t) - x_{g(t)}) \|$ can be bounded. By observing that, if we wait for a sufficient long time after the application of a new FF-CG command, the transient contribution decreases and its 2-norm reduces of a certain fraction of its initial value. The following notion of *Generalized Settling Time* can be stated:

Definition 3.1. ***(Generalized Settling Time) -*** *The integer $\tau > 0$ is said to be a* Generalized Settling Time with parameter γ, *with $0 < \gamma < 1$, for the pair (H_c, Φ), if*

$$\begin{array}{c} \|H_c \Phi^k x\| \leq M(x), \quad \forall k = 0, 1, ..., \tau - 1 \\ \Downarrow \\ \|H_c \Phi^{\tau+k} x\| \leq \gamma M(x), \forall k \geq 0 \end{array} \tag{3.12}$$

holds true for each $x \in \mathbb{R}^n$, with the real $M(x) > 0$ any upper-bound to $\|H_c \Phi^k x\|, \forall k \geq 0$.

As a consequence, if the time interval between two command variations τ were chosen to be a generalized settling time with parameter $\gamma \in (0, 1)$ and $g(t-\tau) = g(t-\tau+1) = ... = g(t-1)$, the disturbance free c-transient would be bounded as follows

$$\| H_c\Phi^k(\hat{x}(t) - x_{g(t-\tau)}) \| \leq \gamma\rho(t-\tau), \quad \forall k \geq 0, \tag{3.13}$$

because $\Phi^\tau(\hat{x}(t-\tau) - x_{g(t-\tau)}) = (\hat{x}(t) - x_{g(t-\tau)})$. The latter observation can be used to characterize the set of all possible feasible commands which may be applied at time t complying with the constraint (3.11). To this end, let us parameterize the generic, non necessarily feasible, command $g \in \mathcal{W}_\delta$ to be applied to the system at time t as the sum of the previously applied command $g(t-\tau)$ and of a command increment Δg to be determined by the FF-CG, i.e. $g = g(t-\tau) + \Delta g$. By noticing that $x_{\Delta g} = x_g - x_{g(t-\tau)}$ and by exploiting the triangular inequality, we can easily bound the transient component of $\bar{c}(k, \hat{x}(t), g)$ as follows

$$\| H_c\Phi^k(\hat{x}(t) - x_g) \| = \| H_c\Phi^k(\hat{x}(t) - x_{g(t-\tau)}) - H_c\Phi^k x_{\Delta g} \| \tag{3.14}$$

$$\leq \quad \|H_c\Phi^k(\hat{x}(t) - x_{g(t-\tau)})\| + \| H_c\Phi^k x_{\Delta g}\| \tag{3.15}$$

$$\leq \quad \gamma\rho(t-\tau) + \|H_c\Phi^k x_{\Delta g} \| \leq \rho(t), \ \forall k \geq 0. \tag{3.16}$$

Then, a possible way to define the scalar $\rho(t)$ at time t complying with (3.16) is:

$$\rho(t) := \gamma\rho(t-\tau) + max_{k\geq 0} \| H_c\Phi^k x_{\Delta g} \| \tag{3.17}$$

where such a bound, as will be clearer in the next Section, is finitely determinable w.r.t. k. Moreover, note also that it does not depend on the state but only on $\rho(t-\tau) \leq \rho_{g(t-\tau)}$ and on the free command increment Δg, which has to be determined so that (3.11) holds true. By direct examination, the latter requirement simply consists of selecting Δg such that $\rho(t) \leq \rho_g$. In fact, the last condition can be explicitly rewritten as follows

$$\| H_c\Phi^k x_{\Delta g} \| \leq \rho_{g(t-\tau)+\Delta g} - \gamma\rho(t-\tau), \forall k \geq 0. \tag{3.18}$$

Finally, by recalling the static map $x_{\Delta g} = (I - \Phi)^{-1} G\Delta g$, the set of all feasible FF-CG commands at time t can be characterized as

$$\begin{cases} g \in \mathcal{W}_\delta \\ (g - g(t-\tau)) \in \Delta\mathcal{G}(g(t-\tau), \rho(t-\tau)) \end{cases} \tag{3.19}$$

where $\Delta\mathcal{G}(g, \rho)$ is defined as the set of all τ-step incremental commands from $g(t-\tau)$ ensuring inequality (3.18) to hold true:

$$\Delta\mathcal{G}(g, \rho) := \left\{\Delta g : \|H_c\Phi^k(I-\Phi)^{-1}G\Delta g\| \leq \rho_{g+\Delta g} - \gamma\rho, \forall k \geq 0\right\}. \tag{3.20}$$

The properties of above set will be detailed in next Section 3.2 where it will be shown that it is finitely determinable ((3.18) have to be explicitly checked only for $0 \leq k \leq k_0$, where k_0 is a certain integer $k_0 \leq \tau$) convex and closed set. Moreover, it will be proved that it is nonempty because it always contains a ball of radius $\eta^* > 0, \Delta\mathcal{G}(g, \rho) \supseteq \mathcal{B}_{\eta^*}$ for any possible command $g \in \mathcal{W}_\delta$ and for any scalar $\rho : 0 \leq \rho \leq \rho_g$. Finally, by using the same selection index of the standard CG approach, we can formulate the Feed-Forward CG algorithm as follows

Algorithm 3.1.1 ***The FF-CG Algorithm***
AT EACH TIME $t=k\tau, k=1,2\ldots$

1.1 SOLVE

$$g(t) = \arg\min_{g} \parallel g - r(t) \parallel^2_{\Psi} \tag{3.21}$$

SUBJECT TO (3.19)
1.2 APPLY $g(t)$ FOR THE NEXT τ STEPS
1.3 UPDATE $\rho(t) = \gamma\rho(t-\tau) + max_{k\geq 0} \parallel H_c\Phi^k(I-\Phi)^{-1}G\Delta g(t) \parallel$.

It is possible to prove that, under the assumption that a feasible couple $(g(0), \rho(0))$ complying with (3.11) is known at time $t = 0$ (which also implies constraints specification and disturbance set $\mathcal{D}$ which make $\mathcal{W}_\delta$ and $\mathcal{G}(g(0), \rho(0))$ non-empty), the above FF-CG scheme enjoys the following properties:

Theorem 3.2. - *Let assumptions* **A1-A2** *be fulfilled. Consider system (2.1) along with the* ***FF-CG*** *selection rule and let an admissible command signal $g(0) \in \mathcal{W}_\delta$ be applied at $t = 0$ and a scalar $\rho(0)$ exist such that $\parallel H_c\Phi^k(x(0) - x_{g(0)}) \parallel \leq \rho(0) \leq \rho_{g(0)},\ \forall k \geq 0$. Then:*

1. *At each istant $t=k\tau, k \in \mathbb{Z}_+$, the minimizer in (3.21) uniquely exists and can be obtained by solving a convex constrained optimization problem;*
2. *The system acted by the FF-CG never violates the constraints, i.e. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$ regardless of any possible admissible disturbance realization $d(\cdot) \in \mathcal{D}$. Namely at each time step such a vector will satisfy:*

$$c(t) \in \left(c_{g(t)} \oplus \mathcal{B}_{\rho(t)}\right) \oplus \Delta_\infty, \forall t \in \mathbb{Z}_+ \tag{3.22}$$

where the term $\left(c_{g(t)} + \mathcal{B}_{\rho(t)}\right)$ is a ball of center $c_{g(t)}$ and radius $\rho(t) < \rho_{g(t)}$, $g(t) = g\left(\left\lfloor \frac{t}{\tau} \right\rfloor \tau\right)$ and $\rho(t) = \rho\left(\left\lfloor \frac{t}{\tau} \right\rfloor \tau\right)$. $\lfloor\cdot\rfloor$ denotes the standard floor operator;

3. *The disturbance-free state $\hat{x}(t)$ lies into a convex and closed set centered in the steady-state value $x_{g(t-1)}$, i.e.*

$$\hat{x}(t) \in \left(x_{g(t-1)} \oplus \mathcal{X}_{\rho(t)}, \forall t \in \mathbb{Z}_+\right) \tag{3.23}$$

where

$$\mathcal{X}_\rho = \left\{x \in \mathbb{R}^n \middle| ||H_c\Phi^k x|| \leq \rho,\ k = 0, ..., \infty\right\}. \tag{3.24}$$

Moreover, if (H_c, Φ) is observable, $\mathcal{X}_\rho$ is compact.

4. *Whenever $r(t) \equiv r$, with r a constant set-point, the sequence of $g(t)$'s converges in finite time either to r or to its best admissible steady-state approximation $\hat{r}$, i.e. $\exists t' > 0$ such that*

$$g(t) = \hat{r} := \arg\min_{g \in \mathcal{W}_\delta} \|g - r\|^2_{\Psi}, \forall t \geq t' \tag{3.25}$$

and moreover

$$\lim_{t\to\infty} \hat{x}(t) = x_{\hat{r}}, \lim_{t\to\infty} \hat{y}(t) = y_{\hat{r}} = \hat{r}. \tag{3.26}$$

Proof

1) The existence of an admissible solution at each time instant $k\tau$ can be proved by simply remarking that $g(t) = g(t-\tau)$, to be chosen along with $\rho(t) = \gamma\rho(t-\tau)$, is always an admissible, although not necessarily the optimal, solution for the prescribed problem at time t. Moreover, being the admissible region (3.19) the intersection of two convex regions $\mathcal{W}_\delta$ and $\Delta\mathcal{G}(g,\rho)$ it results to be a convex set as well. Then, a minimizer for the constrained optimization problem (3.21) subject to (3.19) always exists and its uniqueness follows from the strict convexity of its cost function and admissible region.
2) At each time instant $t = k\tau$, with $k \in \mathbb{Z}_+$, a command $g(k\tau)$ complying with (3.19) is applied to the plant. By construction, the latter implies that the set-valued virtual predictions along the virtual time i defined in (2.7) satisfy $c(i, x(k\tau), g(k\tau), d(\cdot)) \subseteq \mathcal{C}, \forall d(\cdot) \in \mathcal{D}, \forall i \in \mathbb{Z}_+$. Then, the statement is proved by simply noticing that the inclusion $c(t) \in c(i, x(k\tau), g(k\tau), d(\cdot)) \subseteq \mathcal{C}, \forall d(\cdot) \in \mathcal{D}$ holds true for all time instants $t = k\tau + i,\ i \in \{0, 1, ..., \tau-1\}$ and by repeating the same argument for all $k \in \mathbb{Z}_+$. Moreover, it can be observed that, under the FF-CG action $g(k\tau)$, $\hat{c}(t) = \bar{c}(i, \hat{x}(k\tau), g(k\tau))$, for $t = k\tau + i, \forall i \in \{0, 1, ..., \tau-1\}$. Then, the fulfillment of conditions (3.19) implies, via equation (3.11), that $\|\bar{c}(i, \hat{x}(k\tau), g(k\tau)) - c_{g(k\tau)}\| \leq \rho(k\tau), \forall i \in \mathbb{Z}_+$, which, by combining the latter with (3.3) and (3.5), implies (3.22).
3) Conditions (3.23) and (3.24) simply follow by noting that, thanks to (3.11), the following bound holds true

$$\begin{aligned} \|H_c\Phi^{k+i}\left(\hat{x}(t)-x_{g(t)}\right)\| = \|H_c\Phi^k\left(\hat{x}(t+i)-x_{g(t)}\right)\| \leq \rho(t), \\ i=0,...,\tau-1, k=0,...,\infty \end{aligned} \tag{3.27}$$

The convexity of the set $\mathcal{X}(\rho)$ can be proved by means of classical triangular inequalities arguments. In fact, because

$$\begin{aligned} \left\|H_c\Phi^k\left(\lambda x' + (1-\lambda)x''\right)\right\| = \left\|\lambda H_c\Phi^k x' + (1-\lambda)H_c\Phi^k x''\right\| \leq \\ \leq \lambda\left\|H_c\Phi^k x'\right\| + (1-\lambda)\left\|H_c\Phi^k x''\right\| \end{aligned} \tag{3.28}$$

one can state that if $x', x'' \in \mathcal{X}(\rho)$ for a fixed value of ρ, then $(\lambda x' + (1-\lambda)x'') \in \mathcal{X}(\rho), \forall\lambda \in [0\ 1]$. Finally, consider the first n conditions $\|H_c\Phi^k x\| \leq \rho$, $k = 0, 1, ..., n-1$ of (3.24). They define a compact set inside the image of the linear map $c_x = \Theta x$ with Θ the Observability matrix

$$\Theta := [(H_c)^T\ \ (H_c\Phi)^T\ \ (H_c\Phi^2)^T\ \ \ldots\ \ (H_c\Phi^{n-1})^T]^T$$

and

$$c_x := [(H_c x)^T, (H_c\Phi x)^T, ..., (H_c\Phi^{n-1}x)^T]^T \tag{3.29}$$

Whenever the pair (H_c, Φ) is completely observable, the rank of Θ is full and coincides with its column-rank n. In this case, the linear map

between x and c_x is injective and admits the left-inverse operator $x = (\Theta^T\Theta)^{-1}\Theta^T c_x$. As a consequence, any compact set inside the image of Θ has a compact pre-image on x.

4) Consider $r(t) = r, \forall t \geq t^*$ and assume, without loss of generality, $\Psi = I$. Let $g(t)$ be the FF-CG action at time t, solution of the optimization problem (3.21) with optimal cost $V^2(t) = \| g(t) - r \|^2$. Clearly, $V(t)$ corresponds to the minimal Euclidean distance between $g(t)$ and r compatible with the constraints. As already stated, at time $t + \tau$, $g(t)$ is still an admissible, though not necessarily the optimal, solution at time $t + 1$. Hence

$$V(t + \tau) = \| g(t + \tau) - r \| \leq \| g(t) - r \| = V(t). \tag{3.30}$$

Thus $V(t)$, for constant set-points $r(t) \equiv r$, is a monotonically not increasing sequence. In order to prove the finite-time convergence of this succession to $\hat{r}$, it is enough to evaluate the two possible exclusive situations:

1 - if the optimal $g(t + \tau)$ is such that $\Delta g(t + \tau)$ belongs to the interior of $\Delta\mathcal{G}(g(t), \rho(t))$, i.e. $\Delta g(t + \tau) \in In[\Delta\mathcal{G}(g(t), \rho(t))]$, then no points $g \in \mathcal{W}_\delta$ belonging to any sufficiently small neighborhood $g \in \Xi$ of $g(t + \tau)$ would give rise to a lower cost $\| g - r \| < \| g(t + \tau) - r \|$, for all $g \in \Xi$. This implies that, because of convexity of the objective function (3.21), $g(t + \tau) = \hat{r}$.

2 - if $g(t+\tau)$ is such that $\Delta g(t+\tau)$ belongs to the border of $\Delta\mathcal{G}(g(t), \rho(t))$, then, because of Proposition 3.5's results, the distance between the old command $g(t)$ and the new command $g(t + \tau)$ is at least η^*. This implies that $V(t + \tau)$ will at least decrease by either η^*, in the case $g(t+\tau) \in In[\mathcal{W}_\delta]$, or a fixed and computable amount $\Delta V_{min}(r) > 0$ in the case $g(t + \tau)$ belongs also to the border of $\mathcal{W}_\delta$.

The latter allows one to conclude that for any $r \in \mathbb{R}^m$ there exists an integer

$$t' \leq \tau \left\lceil \frac{V(0) - \hat{V}_r}{\Delta V_{min}(r)} \right\rceil \tag{3.31}$$

with $\lceil \cdot \rceil$ denoting the ceiling operator and $\hat{V}_r := \|\hat{r} - r\|$. Finally, conditions (3.26) follow directly from **A1-A2**.

3.2 Solvability and Computability

In this section, a complete characterization of the main properties of the generalized settling time τ and of the convex set $\Delta\mathcal{G}(g, \rho)$ will be provided along with a description of the underlying computational aspects. Moreover, in the last subsection we will show how proposed general schemes can be properly tailored for the special (but very common) box constraints case.

3.2.1 Generalized Settling Time τ

In order to build an efficient FF-CG algorithm is important to determine a good estimate of the minimum generalized settling time τ for a given factor value γ. As a first step, let us recall the following result proved in [1]:

Lemma 3.3. *Let Φ asymptotically stable. Then, for any $x \in \mathbb{R}^n$ there exists an integer $k_0 \in \mathcal{Z}_+$ such that $\|H_c\Phi^k x\| \leq M, k = 0, \ldots, k_0$ implies $\|H_c\Phi^k x\| \leq M, k = k_0 + 1, ..., \infty$ for a whatever sufficiently large positive real $M \in \mathbb{R}_+$.*

As already noticed, such a result makes many of the quantities (e.g. $\mathcal{G}(g, \rho)$ in (3.20)) of the above FF-CG scheme finitely determinable once the integer k_0 is determined, e.g. via the algorithm provided in Section 2.3. Interestingly enough such an integer k_0 can be seen as a particular case of a generalized settling time when $\gamma = 1$. The following result shows, for the general case of $\gamma < 1$, the existence of an analytically determinable upper-bound to the minimum generalized settling time:

Lemma 3.4. *Let the pair (H_c^o, Φ^o) be the observable subsystem obtained via a canonical Kalman observability decomposition of (H_c, Φ) and Φ be asymptotically stable. Then, the integer*

$$\bar{\tau} = \left\lceil \log_\lambda \left(\frac{\gamma}{\bar{\sigma}(H_c^o) B \bar{\sigma} \left\{ \left((\Theta^o)^T \Theta^o \right)^{-1} (\Theta^o)^T \right\} \sqrt{n}} \right) \right\rceil \tag{3.32}$$

provides an upper-bound to the minimum generalized settling time τ with parameter γ *for (H_c, Φ), where $\Theta^o := \left[(H_c^o)^T, (H_c^o \Phi^o)^T, \ldots, (H_c^o (\Phi^o)^{n-1})^T \right]^T$ and B and λ two scalars such that $||(\Phi^o)^k|| \leq B\lambda^k$.*

Proof - First observe that the constraints are influenced only by the observable part of the state x^o, with $x^o = P^o x$ and P^o accounting for the change of state coordinates defined by the Kalman observability decomposition. In fact, one has that $H_c\Phi^k x = H_c^o (\Phi^o)^k x^o, \quad \forall k \in \mathbb{Z}_+$. Moreover, the asymptotical stability of Φ (and hence of Φ^o) implies the exponential convergence of its modes. Then, there exist two computable positive scalars $B > 0, \lambda \in (0, 1)$ such that $\|(\Phi^o)^k\| \leq B\lambda^k$. Our goal is to find an integer τ such that if

$$||H_c^o (\Phi^o)^k x^o|| \leq M(x^o), k = 0, ..., \infty \tag{3.33}$$

then $||H_c^o (\Phi^o)^{k+\tau} x^o|| \leq \gamma M(x^o), k = 0, ..., \infty$ follows. By resorting to the notation and results of Theorem 3.2, we can prove that

$$\|x^o\| = \left\| \left((\Theta^o)^T \Theta^o \right)^{-1} (\Theta^o)^T c_x \right\| \leq \bar{\sigma} \left\{ \left((\Theta^o)^T \Theta^o \right)^{-1} (\Theta^o)^T \right\} \|c_x\|, \tag{3.34}$$

where c_x is the vector defined in (3.29) and $\bar{\sigma} \left\{ \left((\Theta^o)^T \Theta^o \right)^{-1} (\Theta^o)^T \right\}$ denotes the largest singular value of $\left((\Theta^o)^T \Theta^o \right)^{-1} (\Theta^o)^T$. Under the hypothesis (3.33) holding true, we can easily show that

$$\|c_x\|^2 = ||H_c^o x^o||^2 + ||H_c^o \Phi^o x^o||^2 + \ldots + ||H_c^o (\Phi^o)^{n-1} x^o||^2 \leq nM^2(x^o) \quad (3.35)$$

and, because $\|c_x\| \leq \sqrt{n}M(x^o)$, we can bound $||x^o||$ as

$$\|x^o\| \leq \bar{\sigma}\left\{\left((\Theta^o)^T\Theta^o\right)^{-1}(\Theta^o)^T\right\}\sqrt{n}M(x^o). \quad (3.36)$$

Then, by exploiting the exponential stability of Φ^o, the term $\|H_c^o(\Phi^o)^k x^o\|$ can be bounded as

$$\|H_c^o(\Phi^o)^k x^o\| \leq \bar{\sigma}(H_c^o)B\lambda^k\bar{\sigma}\left\{\left((\Theta^o)^T\Theta^o\right)^{-1}(\Theta^o)^T\right\}\sqrt{n}M(x^o). \quad (3.37)$$

and an upper-bound to the minimum generalized settling time can be determined by looking for the minimum integer τ such that

$$\bar{\sigma}(H_c^o)B\lambda^\tau\bar{\sigma}\left\{\left((\Theta^o)^T\Theta^o\right)^{-1}(\Theta^o)^T\right\}M(x^o)\sqrt{n} \leq \gamma M(x^o). \quad (3.38)$$

Finally, (3.32) is obtained by resorting to the logarithms' properties.

The settling time value obtained by means of the previous result might be quite conservative. Hereafter we propose a method to find tighter value of τ complying with (3.12). Note first that, for our goals, we do not need to verify (3.12) for all $x \in \mathbb{R}^n$, but we can restrict our attention to a convex and closed set described by a finite number of constraints:

$$\begin{aligned}\Omega &:= \left\{x \in \mathbb{R}^n : \|H_c\Phi^k x\| \leq \rho_{max}, k = 0, ..., \infty\right\} = \\ &= \left\{x \in \mathbb{R}^n : \|H_c\Phi^k x\| \leq \rho_{max}, k = 0, ..., k_0\right\}\end{aligned}$$

with k_0 complying with Lemma 3.3 as computed in Section 2.3 and $\rho_{max} = max_{g\in\mathcal{W}_\delta}\rho_g$ denoting the maximum distance ρ_g between the steady-state value c_g and $\mathcal{C}_\infty$ frontier obtainable for any admissible input vector $g \in \mathcal{W}_\delta$. Then, our task is to find an integer τ ensuring that the satisfaction of

$$\| H_c\Phi^k x \| \leq M(x), \ \forall x \in \Omega, \forall k = 0, \ldots, \tau - 1 \quad (3.39)$$

for some $M(x)$ implying that $\| H_c\Phi^{\tau+i}x \| \leq \gamma M(x), \forall x \in \Omega, \forall i = 0, \ldots, k_0$ are satisfied as well. Consider now the function $M'(x) := \max_{k=0,\ldots,\tau-1} \|H_c\Phi^k x\|$. By construction, such a function is a lower-bound for any possible $M(x)$ complying with (3.39). Because $M'(x) \leq M(x)$, we can recast the computation of the generalized settling time as the problem of finding an integer τ such that

$$\| H_c\Phi^{\tau+i}x \| \leq \gamma \max_{k=0,\ldots,\tau-1} \|H_c\Phi^k x\|, \forall x \in \Omega, \forall i = 0, \ldots, k_0.$$

A necessary and sufficient condition for the latter is that there exists an integer $k \in \{0, ..., \tau - 1\}$ such that $\gamma \| H_c\Phi^k x \| - \| H_c\Phi^{\tau+i}x \| \geq 0, \forall x \in \Omega, \forall i = 0, \ldots, k_0$. Then, our goal can be rewritten as the problem of finding a τ such that there exists a $k \in \{0, ..., \tau - 1\}$ ensuring

$$\min_{x\in\Omega}\left(\gamma \| H_c\Phi^k x \| - \| H_c\Phi^{\tau+i}x \|\right) \geq 0, \forall i = 0, \ldots, k_0. \quad (3.40)$$

The latter reformulation can be used to build up an algorithm for the determination of the minimum τ by performing a binary search in the range $[1, \bar{\tau}]$ as follows

Algorithm 3.2.1 ***Generalized Settling Time Algorithm***

1.1 SET $\underline{l} := 1$
1.2 SET $\bar{l} := \bar{\tau}$
1.3 REPEAT UNTIL $\bar{l} - \underline{l} > 1$
  1.3.1 SET $\tau := \lceil (\underline{l} + \bar{l})/2$
  1.3.2 IF τ IS SUCH THAT IT EXISTS $k \in \{1, ..., \tau - 1\}$ ENSURING *(3.40)*
    1.3.2.1 SET $\bar{l} := \tau$
    ELSE
    1.3.2.2 SET $\underline{l} := \tau$
  1.3.3 GO TO (3.1)

Note that even such an algorithm is heavy from a computational point of view, it is not unrealistic to off-line solve it from the outset. Furthermore, it is worth to remark that even if problem (3.40) is non-convex, it relies in the special class of the Difference of Convex functions programming (a.k.a. DC programming) problems and can be thus solved in an efficient way (see the Appendix).

3.2.2 Computation and properties of $\Delta\mathcal{G}(g, \rho)$

The goal of this subsection is to completely characterize the set $\Delta\mathcal{G}(g, \rho)$ when $\mathcal{C}_\infty$ consists of polyhedral constraints, which are described as a collection of linear inequalities to computed by following the procedure (2.37-2.46)

$$\mathcal{C}_\infty := \{c \in \mathbb{R}^{n_c} \ : \ Tc \leq q^{k_\varepsilon}\} \tag{3.41}$$

where $T = [T_1, ..., T_{n_v}]^T \in \mathbb{R}^{n_v \times n_c}$, $q^{k_\varepsilon} = \left[q_1^{k_\varepsilon}, ..., q_{n_v}^{k_\varepsilon}\right]^T \in \mathbb{R}^{n_v}$ and the operator $\leq$ acts component-wise. By resorting to its definition in (3.20) and by exploiting Lemma 3.3, the set $\Delta\mathcal{G}(g, \rho)$ consists of all vectors Δg such that inequalities $||H_c \Phi^k (I - \Phi)^{-1} G \Delta g|| \leq \rho_{g+\Delta g} - \gamma\rho$ hold true for $k = 0, ..., k_0$.

The first step in order to understand the meaning of the above inequalities is to make clear what the term $\rho_{g+\Delta g}$ is. By definition (3.9), the latter represents the minimum Euclidean distance between $c_{g+\Delta g}$ and the border of the admissible set $\mathcal{C}_\infty$. Being the admissible set a polyhedron, such a distance can be computed as the minimum of the Euclidean distances between the point $c_{g+\Delta g}$ and each of the hyperplanes $T_i^T c = q_i^{k_\varepsilon}, \ i = 1, \ldots, n_v$ defining the half-spaces $T_i^T c \leq q_i^{k_\varepsilon}, \ i = 1, \ldots, n_v$. By using standard geometrical results, such a distance can be computed as the solution of the following linear programming problem

$$\rho_{g+\Delta g} = max_{\bar{\rho}}\bar{\rho} \tag{3.42}$$

$$subject\ \ to: 0 \leq \bar{\rho} \leq \frac{q_i^{k_\varepsilon} - T_i^T c_{g+\Delta g}}{||T_i||},\ \ i = 1, ..., n_v \tag{3.43}$$

where the scalar $\bar{\rho}$ is a slack variable and the right-most term in (3.43) describes the standard unsigned Euclidean distance between $c_{g+\Delta g}$ and the i-th hyperplane describing the boundaries of the polyhedral set $\mathcal{C}_\infty$. Because $c_{g+\Delta g} \in \mathcal{C}_\infty$, such a term is always non-negative. The latter expression allows one to describe the set $\Delta\mathcal{G}(g,\rho)$ as the set of vectors Δg such that a positive scalar $\bar{\rho} \geq \gamma\rho$ exists ensuring

$$\begin{aligned} &\| H_c\Phi^k(I-\Phi)^{-1}G\Delta g\| \leq \bar{\rho} - \gamma\rho, k = 0, ..., k_0 \\ &\bar{\rho} \leq \frac{q_i^{k_\varepsilon} - T_i^T\left[H_c(I-\Phi)^{-1}G(g+\Delta g)+L(g+\Delta g)\right]}{||T_i||},\ i = 1,..,n_v \end{aligned} \tag{3.44}$$

By means of the previous recasting and by taking into account the evaluation of set $\mathcal{W}_\delta$ in (2.49), the problem (3.21) at the generic time instant t, involving conditions (3.19) can be translated as

$$g(t) = \arg\min_g (g - r(t))^T\Psi(g - r(t))$$

$$s.t. \begin{cases} T\left(H_c(I-\Phi)^{-1}G + L\right)g \leq q^{k_\varepsilon} - (\varepsilon+\delta)[\sqrt{T_i^TT_i}] \\ \| T_i \|\| H_c\Phi^k(I-\Phi)^{-1}G(g - g(t-\tau)) \| \\ \quad + T_i^T(H_c(I-\Phi)^{-1}G + L)g \leq q_i^{k_\varepsilon} - (\varepsilon + \gamma\rho(t-\tau))\sqrt{T_i^TT_i}, \\ \qquad\qquad\qquad\qquad\qquad\qquad i = 1, ..., n_v,\ k = 0, ..., k_0 \end{cases} \tag{3.45}$$

where $[\sqrt{T_i^TT_i}]$ is defined in (2.24), k_0 is computed by means of Algorithm 2.3.2 and τ for the considered γ is evaluated according to Algorithm 3.2.1 Moreover, formulation (3.44) allows one to prove the following properties.

Proposition 3.5. - *Let the set $\Delta\mathcal{G}(g,\rho)$ be characterized by (3.44)-(3.45) for any $g \in \mathcal{W}_\delta$ and $\rho \in \mathbb{R}$ such that, $0 \leq \rho \leq \rho_g$. Then, such a set results:*

1. *convex, closed and finitely determinable;*
2. *nonempty. In particular, there exists a scalar $\eta^* > 0$ such that the set $\Delta\mathcal{G}(g,\rho)$ contains a ball of radius η^*, i.e. $\exists\eta^* > 0 : \Delta\mathcal{G}(g,\rho) \supseteq \mathcal{B}_{\eta^*},\ \forall g \in \mathcal{W}_\delta, \forall\rho \leq \rho_g, \rho \geq 0$;*
3. *compact, whenever $rank\{G^o\} = m$ where the tuple (H_c^o, Φ^o, G^o) is the observable subsystem obtained via a canonical Kalman observability decomposition of (H_c, Φ, G).*

Proof - 1) Closure and finite constraints cardinality follow by the fact that the set $\Delta\mathcal{G}(g,\rho)$ may be computed by means of the finite set of non-strict inequalities (3.44)-(3.45). For what regards convexity, let us assume two vectors $[\Delta g_1^T, \bar{\rho}_1]^T$ and $[\Delta g_2^T, \bar{\rho}_2]^T$ are given complying with (3.44)-(3.45). We need to

prove that, for any $\lambda \in [0,1]$, the vector $[\lambda \Delta g_1^T + (1-\lambda)\Delta g_2^T, \lambda\bar{\rho}_1 + (1-\lambda)\bar{\rho}_2]^T$ still complies with (3.44)-(3.45). This may be trivially verified for what regards the linear inequalities (3.45). Let us focus on the left hand side (3.44). By the triangular inequality, we obtain

$$\begin{aligned}&\|H_c\Phi^k(I-\Phi)^{-1}G(\lambda\Delta g_1+(1-\lambda)\Delta g_2)\|\\ =&\|\lambda H_c\Phi^k(I-\Phi)^{-1}G\Delta g_1+(1-\lambda)H_c\Phi^k(I-\Phi)^{-1}G\Delta g_2\|\\ \leq&\lambda\,\|H_c\Phi^k(I-\Phi)^{-1}G\Delta g_1\|+(1-\lambda)\,\|H_c\Phi^k(I-\Phi)^{-1}G\Delta g_2\|,\\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k=0,...,k_0\end{aligned}$$

Then, because $[\Delta g_1^T, \bar{\rho}_1]^T$ and $[\Delta g_2^T, \bar{\rho}_2]^T$ satisfy (3.44), for $\lambda \geq 0, (1-\lambda) \geq 0$ we obtain

$$\begin{aligned}&\|H_c\Phi^k(I-\Phi)^{-1}G(\lambda\Delta g_1+(1-\lambda)\Delta g_2)\|\\ &\leq \lambda(\bar{\rho}_1-\gamma\rho)+(1-\lambda)(\bar{\rho}_2-\gamma\rho) = (\lambda\bar{\rho}_1+(1-\lambda)\bar{\rho}_2)-\gamma\rho,\end{aligned}$$

for all $k = 0, ..., k_0$ which finally ensures

$$\begin{aligned}&\|H_c\Phi^k(I-\Phi)^{-1}G(\lambda\Delta g_1+(1-\lambda)\Delta g_2)\|\\ &\leq (\lambda\bar{\rho}_1+(1-\lambda)\bar{\rho}_2)-\gamma\rho, \quad k=0,...,k_0.\end{aligned}$$

2) If we denote the Euclidean distance between the point c_g and the i-th support hyperplane of $\mathcal{C}_\infty$ as $\rho_{g,i} = \frac{q_i^{k_\varepsilon} - \left(T_i^T\left(H_c(I-\Phi)^{-1}G+L\right)g\right)}{\|T_i\|}, i = 1, .., n_v$ we can rewrite the inequality (3.45) as

$$\bar{\rho} \leq \rho_{g,i} - \frac{T_i^T\left(H_c(I-\Phi)^{-1}G+L\right)\Delta g}{\|\,T_i\,\|}, i = 1, \ldots, n_v \tag{3.46}$$

By combining (3.46) with (3.44), we can also rewrite $\Delta\mathcal{G}(g,\rho)$ as the set of all vectors Δg such that

$$\|\,H_c\Phi^k(I-\Phi)^{-1}G\Delta g\,\| \leq \rho_{g,i} - \frac{T_i^T\left(H_c(I-\Phi)^{-1}G+L\right)\Delta g}{\|T_i\|} - \gamma\rho, k \geq 0, i = 1, \ldots, n_v. \tag{3.47}$$

Then, because $\rho_g \geq \rho$, a sufficient condition ensuring (3.47) holding true is given by

$$\|\,H_c\Phi^k(I-\Phi)^{-1}G\Delta g\,\| \leq -\frac{T_i^T\left(H_c(I-\Phi)^{-1}G+L\right)\Delta g}{\|T_i\|} + \rho_{g,i} - \gamma\rho_g, k \geq 0, i = 1, \ldots, n_v. \tag{3.48}$$

Let us focus now on the term $\rho_{g,i} - \gamma\rho_g$. By construction $\rho_g \leq \rho_{g,i}$. Then, $\rho_{g,i} - \gamma\rho_g \geq (1-\gamma)\rho_g \geq 0$. Moreover, $g \in \mathcal{W}_\delta$ implies that $\rho_g \geq \delta$. Then, $\rho_{g,i} - \gamma\rho_g \geq (1-\gamma)\delta$ and the condition (3.48) can be rewritten as

$$\|\,H_c\Phi^k(I-\Phi)^{-1}G\Delta g\,\| + \frac{\left|T_i^T\left(H_c(I-\Phi)^{-1}G+L\right)\Delta g\right|}{\|T_i\|} \leq (1-\gamma)\delta, k \geq 0, i = 1, \ldots, n_v. \tag{3.49}$$

Now, if we decompose the FF-CG action increment $\Delta g \in \mathbb{R}^m$ as the product of an arbitrary unitary vector $\hat{v} \in \mathbb{R}^m$ ($||\hat{v}|| = 1$) and of scalar $\eta > 0$, such that $\Delta g = \hat{v}\eta$, condition (3.49) becomes

$$\eta\left(\| H_c\Phi^k(I-\Phi)^{-1}G\hat{v}\| + \frac{\left|T_i^T\left(H_c(I-\Phi)^{-1}G+L\right)\hat{v}\right|}{\|T_i\|}\right) \leq (1-\gamma)\delta, k \geq 0, i = 1, \ldots, n_v. \tag{3.50}$$

To prove the claim it is enough to note that, for each possible $k \geq 0$, $i = 1, ..., n_v$ and for any possible direction $\hat{v}$, the term $\| H_c\Phi^k(I-\Phi)^{-1}G\hat{v} \| + \frac{\left|T_i^T\left(H_c(I-\Phi)^{-1}G+L\right)\hat{v}\right|}{\|T_i\|}$ is bounded. Then, there exists a sufficiently small value η^* such that inequality (3.50) holds true for any possible direction vector $\hat{v}$ and for any $\eta \leq \eta^*$.

3) In order to prove compactness consider that, being C_∞ a compact set, the scalar $\rho_{max} \geq \rho_g, \forall g \in \mathcal{W}_\delta$ is always bounded. Then, a necessary condition for (3.44) being satisfied is

$$\| H_c\Phi^k(I-\Phi)^{-1}G\Delta g\| \leq \rho_{max}, k \geq 0, \tag{3.51}$$

which is equivalent to

$$\| H_c^o(\Phi^o)^k(I-\Phi^o)^{-1}G^o\Delta g\| \leq \rho_{max}, k \geq 0. \tag{3.52}$$

By resorting to the same arguments used in Theorem 1, the latter implies that for any admissible $\Delta g \in \Delta\mathcal{G}(g(t-\tau), \rho(t-\tau))$ the vector $v = (I-\Phi^o)^{-1}G^o\Delta g$ belongs to set of observable states which can be shown to be compact with the same arguments used in the proof of item 2 of Theorem 3.2. This finally implies that all Δg inverse-images of such v have to belong to a compact set as well if the matrix $(I-\Phi^o)^{-1}G^o$ is full column rank, which requires $rank\{G^o\} = m$, being $(I-\Phi^o)^{-1}$ invertible by assumption.

3.2.3 The special case of box constraints

In many practical applications the constraints have a box structure

$$\mathcal{C} := \{c \in \mathbb{R}^{n^c} \ : \ \underline{q}_i \leq c_i \leq \bar{q}_i, i = 1, \ldots, n_c.\} \tag{3.53}$$

with $\underline{q}_i \in \mathbb{R}$ and $\bar{q}_i \in \mathbb{R}$ representing respectively the maximum and the minimum value that the i-th component of the constrained vector c may assume. By exploiting this particular constraints structure it is possible to simplify the design of the presented FF-CG and reduce its conservativeness. The main insight is that we can easily decompose the constraint set $\mathcal{C}_\infty$ into the Cartesian product of n_c sets of scalar intervals:

$$\mathcal{C} = \mathcal{C}^{(1)} \times \mathcal{C}^{(2)} \times ... \times \mathcal{C}^{(n_c)} \tag{3.54}$$

where

$$\mathcal{C}^{(i)} := \left\{c_i \in \mathbb{R} \,\middle|\, \underline{q}_i \leq c_i \leq \bar{q}_i\right\} \tag{3.55}$$

or equivalently

$$\mathcal{C}^{(i)} := \left\{c_i \in \mathbb{R} \,\middle|\, \begin{bmatrix} c_i \\ -c_i \end{bmatrix} \leq \begin{bmatrix} \bar{q}_i \\ -\underline{q}_i \end{bmatrix}\right\}. \tag{3.56}$$

As consequence, the set $\mathcal{C}_\infty$ results decomposed in n_c decoupled subsets as well as $\mathcal{C}^{(i)}$

$$\mathcal{C}_\infty = \mathcal{C}_\infty^{(1)} \times \mathcal{C}_\infty^{(2)} \times ... \times \mathcal{C}_\infty^{(n_c)} \tag{3.57}$$

where, by exploiting procedure (2.37-2.46), each set $\mathcal{C}_\infty^{(i)}$ has the following formulation

$$\mathcal{C}_\infty^{(i)} := \left\{ c_i \in \mathbb{R} \left| \begin{bmatrix} c_i \\ -c_i \end{bmatrix} \leq \begin{bmatrix} \bar{q}_i^{k_\epsilon} - (\varepsilon + \delta) \\ -\underline{q}_i^{k_\epsilon} - (\varepsilon + \delta) \end{bmatrix} \right. \right\} \tag{3.58}$$

While this structure does not give any benefit in the computation of the set $\mathcal{W}_\delta$, it may give advantages in the computation of the command increment Δg. The idea is that, if we choose Δg_i such that $\hat{c}_i(k, \hat{x}(t), g_i(t)) \in \mathcal{C}^{(i)}, k \geq 0$ for all $i = 1, ..., n_c$, then $\bar{c}(k, \hat{x}(t), g(t)) \in \mathcal{C}_\infty, k \geq 0$. This can be easily exploited by means of the following algorithm:

Algorithm 3.2.2 ***The FF-CG Algorithm (Box constraints)***
AT EACH TIME $t = \kappa\tau, \kappa = 1, 2 \ldots$

1.1 SOLVE

$$g(t) = \arg\min_g \parallel g - r(t) \parallel_\Psi^2 \tag{3.59}$$

SUBJECT TO

$$\begin{cases} g \in \mathcal{W}_\delta \\ (g - g(t-\tau)) \in \Delta\mathcal{G}^{(i)}(g(t-\tau), \rho^{(i)}(t-\tau)), i = 1, ..., n_c \end{cases} \tag{3.60}$$

1.2 APPLY $g(t)$
1.3 UPDATE $\rho^{(i)}(t) = \gamma\rho^{(i)}(t-\tau) + max_{k\geq 0}\{|H_{c,i}\Phi^k(I-\Phi)^{-1}G\Delta g|\}, i = 1, \ldots, n_c$

where $H_{c,i}$ represents the i-th line of H_c and sets $\Delta\mathcal{G}^{(i)}(g, \rho^{(i)})$ are defined as

$$\Delta\mathcal{G}^{(i)}(g, \rho^{(i)}) := \left\{ \Delta g \in \mathbb{R}^n : |H_{c,i}\Phi^k(I-\Phi)^{-1}G\Delta g| \leq \rho^{(i)}_{g+\Delta g} - \gamma\rho^{(i)}, k \geq 0 \right\}. \tag{3.61}$$

Note that, being the Euclidean distance defined as

$$\rho^{(i)}_{g+\Delta g} = min \left\{ c^i_{g+\Delta g} - \underline{q}_i^{k_\varepsilon}, \bar{q}_i^{k_\varepsilon} - c^i_{g+\Delta g} \right\} \tag{3.62}$$

those sets are very easily computable. In fact, in this case (3.59)-(3.60) can be recast in the following QP problem with linear constraints

$$g(t) = \arg\min_g (g - r(t))^T \Psi (g - r(t))$$
$$s.t. \begin{cases} (S_{c,i})\, g \leq \bar{q}_i^{k_\varepsilon} - \delta \\ (-S_{c,i})\, g \leq -\underline{q}_i^{k_\varepsilon} - \delta \\ (R^k_{c,i} + S_{c,i})g - R^k_{c,i}g(t-\tau) \leq \bar{q}_i^{k_\varepsilon} - (\varepsilon + \delta) - \gamma\rho^{(i)}(t-\tau) \\ (-R^k_{c,i} + S_{c,i})g + R^k_{c,i}g(t-\tau) \leq \bar{q}_i^{k_\varepsilon} - (\varepsilon + \delta) - \gamma\rho^{(i)}(t-\tau) \\ (R^k_{c,i} - S_{c,i})g - R^k_{c,i}g(t-\tau) \leq -\underline{q}_i^{k_\varepsilon} - (\varepsilon + \delta) - \gamma\rho^{(i)}(t-\tau) \\ (-R^k_{c,i} - S_{c,i})g + R^k_{c,i}g(t-\tau) \leq -\underline{q}_i^{k_\varepsilon} - (\varepsilon + \delta) - \gamma\rho^{(i)}(t-\tau), \\ \qquad\qquad i = 1, ..., n_c, \;\; k = 0, ..., k_0 \end{cases} \tag{3.63}$$

where

$$R^k_{c,i} := H_{c,i}\Phi^k(I-\Phi)^{-1}G \tag{3.64}$$

$$S_{c,i} := H_{c,i}(I-\Phi)^{-1}G + L. \tag{3.65}$$

Note that also the computation of the generalized settling time with parameter γ can be simplified in this case. In fact, it can be computed as the largest of all settling times obtained for each single constrained variable, i.e. $\tau = max_{i=1,\dots,n_c}\{\tau^{(i)}\}$ where $\tau^{(i)}$ is the settling time with parameter γ for the pair $(\Phi, H_{c,i})$.

3.3 Simulation Studies

In this section, we investigate the effect of different choices of the free parameters involved in the FF-CG design. A simple example will be used for that purposes and also to make comparisons with the classical CG solution. The simulation results presented hereafter have been achieved by using Matlab 7.12 + Simulink 7.7 on a Core 2 Quad personal computer. The standard Matlab FMINCON.M routine was used for the quadratic optimization. Consider the following linear time invariant system

$$\begin{cases} x(t+1) = \Phi x(t) + Gg(t) \\ y(t) \quad = H_y x(t) \\ c(t) \quad = H_c x(t) \end{cases} \tag{3.66}$$

where

$$\begin{gathered} \Phi = \begin{bmatrix} 1.5402 & -0.6703 \\ 1 & 0 \end{bmatrix},\ H_y = \begin{bmatrix} -0.8935 & 1.0237 \end{bmatrix}, \\ G = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ H_c = \begin{bmatrix} -0.8935 & 1.0237 \\ 1 & 0 \end{bmatrix} \end{gathered} \tag{3.67}$$

The CG task is that of ensuring reference tracking while maintaining the output c inside the following set

$$\mathcal{C} := \{c \in \mathbb{R}^2 : -0.4 \le c_1 \le 0.4 \wedge -3 \le c_2 \le 3\} \tag{3.68}$$

Note that $y(t) = c_1(t)$. The (3.66)-(3.67) constraints vector time evolution with initial conditions $x(0) = [-1.5, -1.5]^T$ to a constant reference $g(t) = r(t) = 0.38$ is depicted in Figure 3.4. When neither CG nor FF-CG units are used, the system responses clearly violate the prescribed constraints.

The FF-CG free parameters are δ, γ and Ψ. Actually, because the reference g is scalar, the choice of Ψ has no influence on the optimization problem to solve. Hence, we only take care about the parameters δ and γ which represent the available *design knobs* of the FF-CG scheme. In particular, δ affects the number of constraints involved in the on-line optimization problem by having a strict relationship with the constraint horizon k_0. On the contrary, the choice

of both δ and γ influence the generalized settling time τ, which indicates the amount of sampling steps over which the FF-CG has to constantly apply a previous computed action.

Figure 3.5 shows τ as a function of δ and γ. As expected, for fixed δ and increasing values of γ, the parameter τ becomes increasingly larger. On the contrary, by fixing γ and for decreasingly values of δ, which are the ones of practical interest, τ is increasingly larger. Thus, in this respect, it is no convenient to select both δ and γ very small.

Figure 3.6 shows the system responses $y(t)$ supervised by a FF-CG unit for different values of γ and τ, when $\delta = 0.05$. Although different values of these parameters do not give rise to very different tracking performance, it is suggested to select δ and γ in order to work with small value of τ. This can be a good design choice if the desired reference $r(t)$ changes quickly because the FF-CG scheme will update its action more often.

In the following simulations we used $\delta = 0.05$ and $\gamma = 0.7$, which involve the use of $k_0 = 5$ and $\tau = 7$. The initial supposed uncertainty was $\rho(0) = 0.2$ and the initial applied command was $g(0) = -0.2$. Both FF-CG strategies presented respectively in Algorithms (3.1.1) and (3.2.2) (Box constraints formulation) are considered.

3.3.1 FF-CG Numerical problem

The next on-line optimization problem underlying the FF-CG formulation follows directly from (3.45)

$$g(t) = \arg\min_g (g - 0.38)^T \Psi (g - 0.38)$$

$$s.t. \begin{cases} T\left(H_c(I-\Phi)^{-1}G\right) g \leq q - (0.05)[\sqrt{T_i^T T_i}] \\ \parallel T_i \parallel\parallel H_c\Phi^k(I-\Phi)^{-1}G(g - g(t-7)) \parallel \\ \quad + T_i^T(H_c(I-\Phi)^{-1}G)g \leq q - (0.7\rho(t-7))\sqrt{T_i^T T_i}, \\ \qquad\qquad\qquad\qquad\qquad\qquad i = 1,...,4,\ k = 0,...,5 \end{cases} \tag{3.69}$$

where $\rho(t)$, by recalling Algorithm (3.1.1), is

$$\rho(t) = 0.7\rho(t-\tau) + max_{k\in\{1,...,5\}} \parallel H_c\Phi^k(I-\Phi)^{-1}G(g(t) - g(t-7)) \parallel$$

and the matrices appearing in this formulation take the following values

$$H_c(I-\Phi)^{-1}G = \begin{bmatrix} 1.0008 \\ 7.6864 \end{bmatrix}$$

$$H_c\Phi^0(I-\Phi)^{-1}G = \begin{bmatrix} 1.0008 \\ 7.6864 \end{bmatrix},\ H_c\Phi^1(I-\Phi)^{-1}G = \begin{bmatrix} 1.8943 \\ 6.6864 \end{bmatrix},$$
$$H_c\Phi^2(I-\Phi)^{-1}G = \begin{bmatrix} 2.2467 \\ 5.1462 \end{bmatrix},\ H_c\Phi^3(I-\Phi)^{-1}G = \begin{bmatrix} 2.1907 \\ 3.4443 \end{bmatrix},$$
$$H_c\Phi^4(I-\Phi)^{-1}G = \begin{bmatrix} 1.8681 \\ 1.8554 \end{bmatrix},\ H_c\Phi^5(I-\Phi)^{-1}G = \begin{bmatrix} 1.4089 \\ 0.5490 \end{bmatrix}.$$

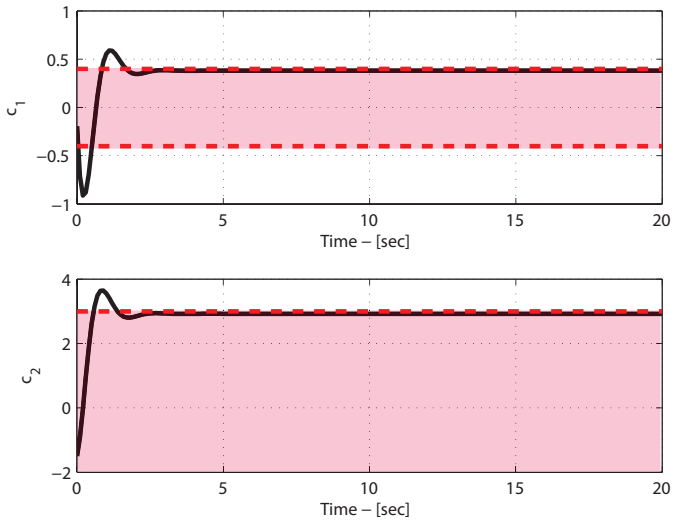

Fig. 3.4. Constrained variables evolution related to a constant reference $g(t) = r(t) = 0.38$ with no CG installed.

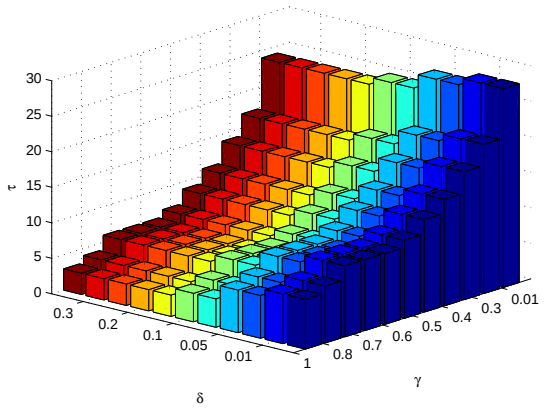

Fig. 3.5. Generalized settling time τ computed via (3.2.1) as a function of δ and γ. The values corresponding to $\gamma = 1$ represents the computed horizon k_0 as a function of δ.

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \; q = \begin{bmatrix} 0.4 \\ 3 \\ 0.4 \\ 3 \end{bmatrix}.$$

3.3.2 FF-CG(Box) Numerical problem

In this case the optimization problem to solve on-line comes from Algorithm (FFCGBoxalg) and problem (3.63). For this example, it results to be given by

$$g(t) = \arg\min_g (g - r(t))^T \Psi (g - r(t))$$
$$s.t. \begin{cases} \begin{bmatrix} S_{c,i} \\ -S_{c,i} \end{bmatrix} g \leq \begin{bmatrix} \bar{q}_i \\ \underline{q}_i \end{bmatrix} - \delta \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ \begin{bmatrix} R^k_{c,i} + S_{c,i} \\ -R^k_{c,i} + S_{c,i} \\ R^k_{c,i} - S_{c,i} \\ -R^k_{c,i} - S_{c,i} \end{bmatrix} g \leq \begin{bmatrix} \bar{q}_i \\ \bar{q}_i \\ -\underline{q}_i \\ -\underline{q}_i \end{bmatrix} - \left(\delta + 0.7\rho^{(i)}(t-7) + R^k_{c,i} g(t-7)\right) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, 2, \; k = 0, ..., 5 \end{cases} \tag{3.70}$$

where each $\rho^{(i)}(t), i = 1, 2$ is computed according to (3.62)

$$\rho^{(i)}(t) = 0.7\rho^{(i)}(t-7) + max_{k \in \{1,...,5\}} \{|R^k_{c,i}(g(t) - g(t-7))|\}, i = 1, 2$$

and matrices and vectors appearing in the latter two equations have the following values

$$S_{c,1} = 1.0008, \; S_{c,2} = 7.6864$$

$$R^0_{c,1} = 1.0008, \; R^1_{c,1} = 1.8943, \; R^2_{c,1} = 2.2467,$$
$$R^3_{c,1} = 2.1907, \; R^4_{c,1} = 1.8681, \; R^5_{c,1} = 1.4089$$

$$R^0_{c,2} = 7.6864, \; R^1_{c,2} = 6.6864, \; R^2_{c,2} = 5.1462,$$
$$R^3_{c,2} = 3.4443, \; R^4_{c,2} = 1.8554, \; R^5_{c,2} = 0.5490.$$

$$\bar{q}_1 = 0.4, \; \bar{q}_2 = 3, \; \underline{q}_1 = -0.4, \; \underline{q}_2 = -3.$$

3.3.3 Simulation Results

The results of simulations of the CG and FF-CG governed system are reported in Figure 3.7, where the output $y(t) = c_1(t)$ (upper) and the computed CG commands $g(t)$ (lower) are depicted for the same set-point $r(t) = 0.38$. In this figure, the classical CG solution (CG in the figures) and both FF-CG schemes, presented respectively in the Algorithm (3.1.1) and the FF-CG(Box) scheme related to the Algorithm (3.2.2), are considered.

It is fair commenting that both the CG and FF-CG(Box) algorithms have been built up so as to exploit the fact that the system is subject to box constraints whereas the FF-CG method does not enjoy this capability and, in turn, it expected to behave more conservatively. In fact, the performance of the CG and FF-CG(Box) algorithms are quite similar while the FF-CG has a slight worse behavior. This fact can be explained by looking at the Figure 3.9 where the *uncertainty* sequences $\rho(t)$ computed by the FF-CG and FF-CG(Box) schemes are depicted. As shown in that figure, the uncertainty related to FF-CG(Box), represented by $\rho_1(t)$ and $\rho_2(t)$, decreases faster than in the FF-CG case.

In Figure 3.8, the constrained variables are depicted: in this case they always satisfy the respective constraints. It is worth noticing that the trajectories of the system acted by the standard CG and the FF-CG(Box) schemes are very close each other and both are much closer to the constraint boundaries than the ones resulting for the FF-CG scheme.

Figure 3.10 shows the system evolution in the $c_1 - c_2$ plane and corroborates the theory presented above, that is based on the idea that the FF-CG based strategies maintain the variable c in a suitable neighborhood of the feasible steady-state set, in the figure depicted by the dotted line. Finally, the on-line computational burdens per step of all schemes are reported in the following Table 3.3.3. Please note that, as expected, FF-CG(Box) is more

	CG	FF-CG(Box)	FF-CG
CPU Time per step [ms]	30.0	27.2	42.5

Table 3.1. On-line phase - CPU time per step

efficient than FF-CG because it has to solve a simpler and less demanding convex optimization problem than the one solved, in general, by FF-CG.

3.4 Conclusions

In this Chapter, a novel Feed-Forward CG approach has been proposed which, thanks to the asymptotic stability of the pre-compensated system, doesn't base its action computation on the explicit measure of the state. On the

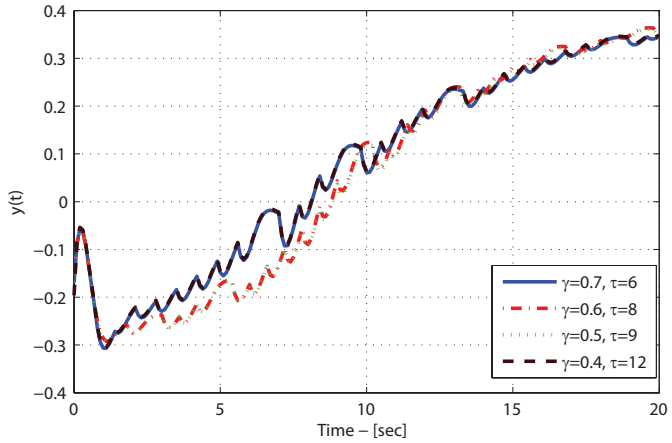

Fig. 3.6. Response of the system subject to FFCG action related to different values of γ and τ

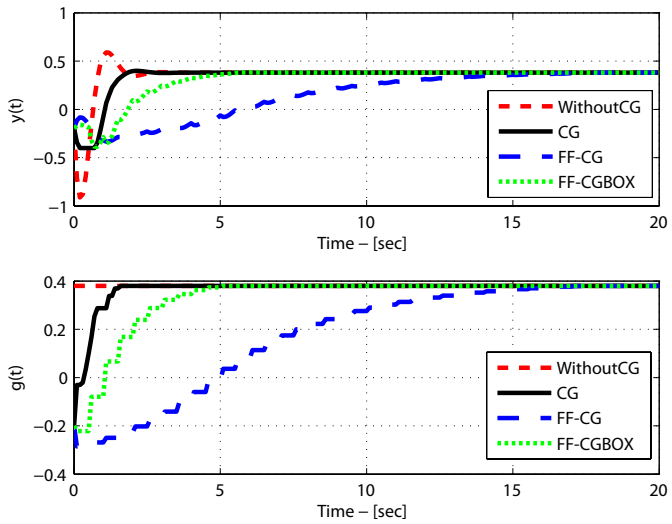

Fig. 3.7. a)[upper] Response of the system with different CG action. b)[down] Computed references

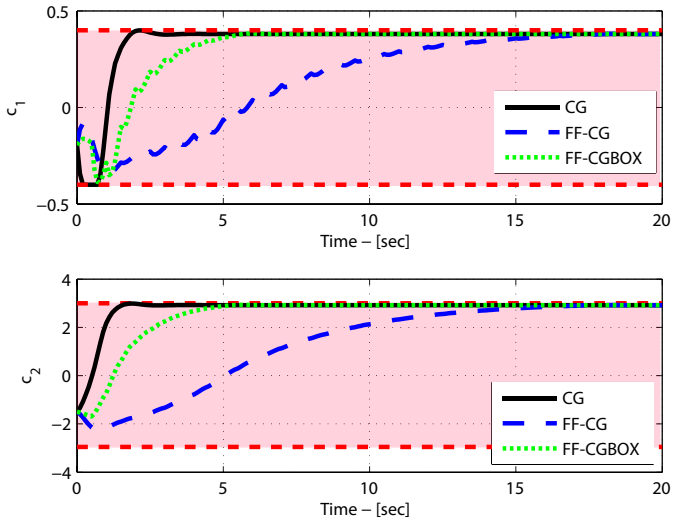

Fig. 3.8. Constrained output

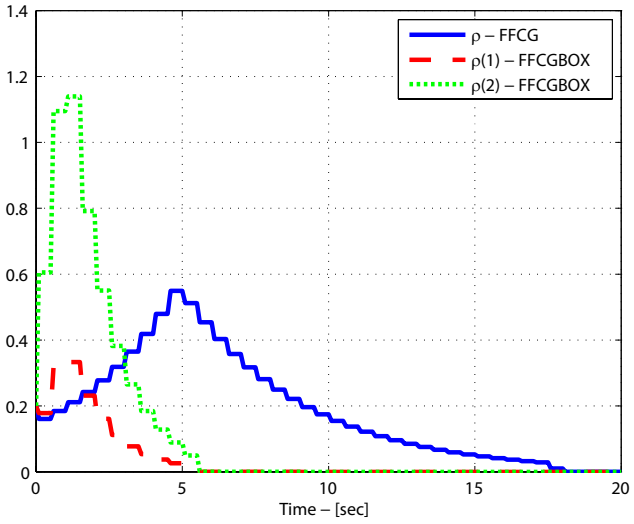

Fig. 3.9. Evolution of $\rho(t)$ (blue line) for FFCG, and $\rho_1(t)$ and $\rho_2(t)$ for FFCGBox

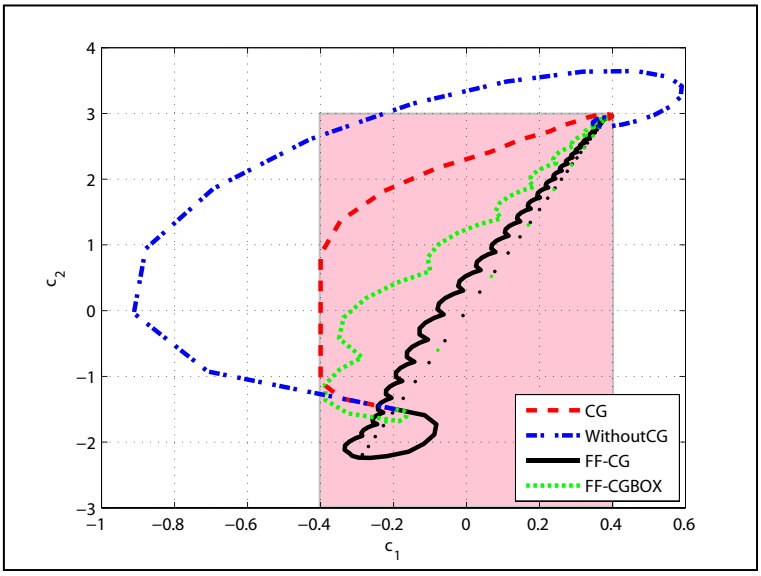

Fig. 3.10. Geometrical representation of condition (3.8) for $c \in \mathbb{R}^2$

contrary, by suitably limiting its action variations it is able to always maintain the constrained variable trajectory "not too far" from the set of admissible steady-state equilibria.

The properties of the proposed algorithm have been fully analyzed and the differences with the standard Feedback CG approach pointed out. Comparisons with the classical CG solution have been presented and commented in the final illustrative example. As discussed in the next chapters of this thesis, this class of solutions may be of interest in all applications where either the measure or the estimation of the whole state is problematic, e.g. in decentralized or distributed networked applications where the cost to make all the entire aggregate state known to all agents at each time instant could be unrealistic or require unrealistic communication infrastructures.

3.5 Appendix

The algorithm proposed in [52] is here reported in order to solve problem (3.40). Consider the DC programming problem of the form

$$\min\{f(x) = g(x) - h(x) : x \in X\} \tag{3.71}$$

where g and h are finite convex functions and X is a convex subset of $\mathbb{R}^n$. Moreover, we assume that the convex set X is compact with a nonempty interior, that is a point $y^0 \in \text{int}\{X\}$ exists. It is proved in [52] that a point

$x^* \in X$ is an optimal solution to (3.71) if and only if there exists a $t^* \in \mathbb{R}$ such that

$$0 = \inf\{-h(x) + t : x \in X, t \in \mathbb{R}, g(x) - t \leq g(x^*) - t^*\}. \tag{3.72}$$

We present an algorithm for solving DC programming problems of the form (3.71). Since the interior of int$\{X\}$ is not empty, it can be packed into an n-simplex $S^0 \subset \mathbb{R}^n$. The main idea of the algorithm is to generate a sequence of feasible points $\{y^0, y1...\} \subset X$, until a feasible point y^k is achieved which fulfills the condition (3.72). For each polytope P, we denote by $V(P)$ the set of its vertices.

Algorithm 3.5.1 ***Initialization.***

1.1 let $w^0 = g(y^0) - h(y^0)$ be the first upper-bound to the optimal value w^* of problem (3.71);
1.2 construct a polytope P^0 containing the set $\{(x,t) : x \in X, t \in \mathbb{R}^n, g(x) - t - w^* = 0\}$ and compute $V(P^0)$;
1.3 set $k = 0$.

***Iteration** k.*

1.1 Compute an optimal solution (x^k, t^k) to the problem $\min\{-h(x) + t : (x,t) \in V(P^k)\}$; if $-h(x^k) + t^k = 0$, then stop: y^k is an optimal solution to problem (3.71) with optimal value w^k; otherwise, compute $y^{k+1} \in X$ such that $w^{k+1} = g(y^{k+1}) - h(y^{k+1}) < w^k$;
1.2 construct a cutting plane, i.e., an affine function $l^k(x,t)$ such that

$$l^k(x^k, t^k) > 0, \tag{3.73}$$
$$l^k(x,t) \leq 0, \quad for\ x \in X, g(x) - t - w^{k+1} \leq 0; \tag{3.74}$$

1.3 set: $P^{k+1} = P^k \cap \{(x,t) : l^k(x,t) \leq 0\}$, and compute $V(P^{k+1})$;
1.4 set $k = k + 1$, and return to 1.1.

For the implementation of Algorithm 3.5.1, we have to specify the following main points:

a) the construction of a first polytope P^0 and its vertex set $V(S^0)$;
b) the choice of the point y^k for $k \geq 1$ and the construction of an affine function $l^k, k \geq 0$, satisfying condition (3.73);
c) the computation of the set $V(P^k),\ k \geq 1$.

In order to compute the vertex set of a polytope defined as the intersection of a polytope with a halfspaces [Task (c)], the methods discussed in [53] can be used. Tasks (a) and (b) are implemented as follows.

Construction of a First Polytope P^0 and Its Vertex Set.

Let

$$V(S^0) = \{v^1, ..., v^{n+1}\}$$

be the set of $n+1$ vertices of the n-simplex S^0, and let s be a subgradient of the convex function g at the point y^0. Further, define the affine function

$$l(x) = (x - y^0)s + g(y^0). \tag{3.75}$$

Then, $l(x)$ is an underestimation of the convex function $g(x)$, and we have

$$\begin{array}{l} \{(x,t) : x \in X, t \in \mathbb{R}^n, g(x) - t - w^* = 0\} \\ \subset \{(x,t) : x \in X, t \in \mathbb{R}^n, g(x) - t - w^0 \leq 0\} \\ \subset \{(x,t) : x \in X, t \in \mathbb{R}^n, l(x) - t - w^0 \leq 0\} \end{array}$$

Next, let $\bar{w}$ and t be real numbers satisfying

$$\bar{w} = min\{l(x) : x \in V(S^0)\} - \max\{h(x) : x \in V(S^0)\},$$

$$\bar{t} > \max\{g(x) : x \in V(S^0)\} - \bar{w}.$$

Then, obviously we have $\bar{w} < w^*$ and

$$\begin{array}{l} \{(x,t) : x \in X, t \in \mathbb{R}^n, g(x) - t - w^* = 0\} \\ \subset \{(x,t) : x \in X, t \in \mathbb{R}^n, g(x) - t - \bar{w} \geq 0\} \\ \subset \{(x,t) : x \in X, t \in \mathbb{R}^n, t \leq \bar{t}\}. \end{array}$$

Thus, a first polytope P^0 containing the set

$$\{(x,t) : x \in X, t \in \mathbb{R}^n, g(x) - t - w^* = 0\}$$

can be defined by

$$P^0 := \{(x,t) : x \in S^0, t \leq \bar{t}, l(x) - t - w^0 \leq 0\}. \tag{3.76}$$

The set $V(P^0)$ consists of the $2(n+1)$ points

$$\begin{array}{ll} (v^i, \bar{t}), & i = 1, ..., n+1, \\ (v^i, l(v^i) - w^0), & i = 1, ..., n+1. \end{array} \tag{3.77}$$

Moreover, by construction, the following property holds:

$$g(y^0) - \bar{t} - w < 0, \quad \forall w \geq w^* \tag{3.78}$$

Choice of y^k and Construction of l^k.

In what follows, we assume that the convex set X is given by

$$X = \{x \in \mathbb{R}^n : a(x) \leq 0\}, \tag{3.79}$$

where a is a convex function defined on $\mathbb{R}^n$. Moreover, at Iteration k of the algorithm, we define a convex function $\beta^k : \mathbb{R}^{n+1} \to \mathbb{R}$ by

$$\beta^k(x,t) = \max\{a(x); g(x) - t - w^k\}. \tag{3.80}$$

Because $y^0 \in \text{int}\{X\}$, it follows from the latter that

$$\beta^k(y^0, \bar{t}) < 0, \quad \forall k \geq 0. \tag{3.81}$$

At the k-th iteration, consider the point (x^k, t^k). Notice that, whenever the algorithm does not terminate at this iteration, we have $-h(x^k) + t^k < 0$. If $x^k \in X$, then set

$$w^{k+1} = min\{w^k; g(x^k) - h(x^k)\};$$

choose y^{k+1} such that

$$g(y^{k+1}) - h(y^{k+1}) = w^{k+1}.$$

An affine function $l^k(x,t)$ is defined by

$$\begin{aligned} l^k(x,t) &= (x - x^k)s^k + g(x^k) - w^{k+1} - t \\ &= s^k x - t - (s^k x^k - g(x^k) + w^{k+1}), \end{aligned} \tag{3.82}$$

where s^k is a subgradient of g at x^k. Obviously, $l^k(x,t)$ satisfies condition (3.73). If $x^k \notin X$, then $\beta^k(x^k, t^k) > 0$. Compute the point (ξ^k, θ^k) in the line segment $[(x^k, t^k), (y^0, \bar{t})]$ satisfying $\beta(\xi^k, \theta^k) = 0$. Set

$$w^{k+1} = \min\{w^k; g(\xi^k) - h(\xi^k)\};$$

choose y^{k+1} such that

$$g(y^{k+1}) - h(y^{k+1}) = w^{k+1}.$$

An affine function $l^k(x,t)$ is defined as

$$l^k(x,t) = ((x,t) - (\xi^k, \theta^k))s^k + \beta^k(\xi^k, \theta^k), \tag{3.83}$$

where s^k is a subgradient of β at (ξ^k, θ^k). Obviously, $l^k(x,t)$ satisfies

$$\begin{aligned} &{}^k(x^k, t^k) > 0, \\ &l^k(x,t) \leq 0, \quad \forall x \in X, g(x) - t - w^k \leq 0. \end{aligned}$$

Therefore, it satisfies condition (3.73) because $w^{k+1} \leq w^k$. The convergence of Algorithm 3.5.1 is proved in [52].

4

An Improved FF-CG approach

This chapter presents an enhanced control algorithm belonging to the class of the *Feed-Forward* Command Governor (FF-CG) strategies. Although effective the previously proposed FF-CG has an inherent limitation when considering bounded disturbance because the CG output is instructed to stay at a constant level for a given number of sampling steps instants between two successive CG computations. The tracking performance of the proposed scheme is obviously affected when references with rapid variations are considered.

The aim of this chapter is to complete the analysis on the FF-CG approach and present a novel class of enhanced FF-CG strategies where such a drawback is completely overcome, in that in this improved algorithm the FF-CG action can now be computed and applied at each sampling time. Such a solution is achieved by observing that, under the same assumptions adopted in chapters 2 and 3, the uncertainty about the state evolution arising from the absence of measurements, involves only the dynamics related to the initial conditions. Such an uncertainty can be bounded and estimated and the resulting FF-CG strategy formulated as a standard state-feedback CG on the basis of a suitable feedback signal, under a restricted set of constraints to be fulfilled. Moreover, in the disturbance-free case it is proved that the performance of this particular FF-CG scheme asymptotically equals that of the standard state-based CG strategy.

The chapter is organized as follows. In Section 4.1, the improved FF-CG scheme is introduced and completely analyzed. In Section 4.2, the numerical details are presented along with several numerical simulations.

4.1 Problem Formulation and improved FF-CG approach

Consider the previously described closed-loop system (2.1)-(2.3) satisfying assumptions **A1-A2**.

Here we will address a slight different approach to the FF-CG design problem, characterized again by the absence of any state measurement in determin-

ing $g(t)$. In order to better introduce the key ideas, let's consider temporarily the disturbance-free ($d(t) \equiv 0_{n_d}$) case and adopt the following notations for the steady-state solutions of (2.1) to a constant command $g(t) \equiv g,\ \forall t$

$$x_g{:=}(I_n{-}\Phi)^{-1}Gg,\ y_g{:=}H_y(I_n{-}\Phi)^{-1}Gg,\ c_g{:=}H_cx_g{+}Lg. \tag{4.1}$$

The idea explicitly employed in the FF-CG scheme of the previous chapter is that, if the generation rate of the modified command $g(\cdot)$ is significantly greater w.r.t. the system dynamics, then, because of system stability (see **A1**), the constrained vector $c(t)$ can always be held within a certain known (and "small") distance $\rho(t) > 0$ from the closed-loop steady-state equilibrium c-vector $c_{g(t)}$

$$c(t) - c_{g(t)} \in \mathcal{B}_{\rho(t)} \tag{4.2}$$

where $\mathcal{B}_{\rho(t)}$ represents the ball of radius $\rho(t)$ centered at the origin. This has been achieved in chapter 3 with strategies of the form

$$g(t) = \underline{g}(r(t), g(t-\tau), \rho(t-\tau)) \tag{4.3}$$

where $g(t)$ is computed every τ sampling steps and it is constantly applied between two successive CG action computations.

In this chapter, a less conservative approach is discussed. To this end, observe that, because $c_{g(t)}$ unambiguously depends on $g(t)$ and $\rho(t)$, it may be proven to be a function of its initial condition $\rho(0)$ and of the commands history up to time $t-1$

$$g^{t-1} := \{g(t-1), g(t-2), ..., g(0)\} \tag{4.4}$$

It is then possible conceive FF-CG schemes where, instead of considering the dependence on the measured state $x(t)$, decisions can be taken on the basis of $\rho(0)$ and of the past values of $g(t)$, that is

$$g(t) = \underline{g}(r(t), g^{t-1}, \rho(0)) \tag{4.5}$$

As it will be clear soon, the entire sequence g^{t-1} it is not necessary to be stored. In fact, a suitable aggregate expression equivalent to the knowledge of g^{t-1} for the computation of (4.5) can be found.

In order to make precise statements consider the constrained closed-loop system (2.1)-(2.3) satisfying assumptions **A1-A2**. As proved in Section 3.1, in spite of state unavailability,

$$\begin{gathered} \overline{c}(k, \hat{x}, g) \in \mathcal{C}_\infty,\ \forall k \in \mathbb{Z}_+ \\ \Downarrow \\ c(k, x, g, d(\cdot)){=}\overline{c}(k, \hat{x}, g){+}\ \tilde{c}(k, d(\cdot)){+}\ H_c\Phi^k\tilde{x}{\subset}\ \mathcal{C}, \forall k{\in}\mathbb{Z}_+ \end{gathered} \tag{4.6}$$

Thus, the constraints fulfilment can be ensured also in this case by only constraining the disturbance-free evolutions as follows

$$g \in \mathcal{W}^{\delta} \tag{4.7}$$

$$\bar{c}(k, \hat{x}, g) = c_g + H_c \Phi^k (\hat{x} - x_g) \in \mathcal{C}_{\infty} \tag{4.8}$$

where $\mathcal{W}_{\delta}$, $\mathcal{C}_{\infty}$ and $\bar{c}(k, \hat{x}, g)$ are defined respectively in (2.8), (2.11). The key idea used here for the construction of an effective FF-CG algorithm is as follows. Let us assume that at time $t = 0$ a command $g(0) \in \mathcal{W}_{\delta}$ has been applied to the system, the transient components of $\bar{c}(k, \hat{x}(0), g(0)), k \geq 0$, being confined into a ball of known radius $\rho(0)$ around $c_{g(0)}$ and the constraints are not violated, i.e. $c_{g(0)} \in \mathcal{C}_{\infty} \sim \mathcal{B}_{\rho(0)}$. The transient part of the predictions will be thus bounded as follows

$$\| H_c \Phi^k (\hat{x}(0) - x_{g(0)}) \| \leq \rho(0), \forall k \geq 0. \tag{4.9}$$

It can be noted that if a long time interval lasts after the application of a new FF-CG command, the transient contribution would decrease and could be bounded within a certain percentage of its initial bound $\rho(0)$. For the forthcoming discussion, the following definitions are in order:

Definition *(Guaranteed Contraction Sequence)* - The sequence $\gamma(k|t) \leq 1, \forall k \geq 0$ is a Guaranteed Contraction Sequence for the pair (H_c, Φ) at time t if

$$\begin{array}{c} \|H_c \Phi^k x\| \leq M(x), \qquad k = 0, 1, ..., t-1 \\ \Downarrow \\ \|H_c \Phi^{t+k} x\| \leq \gamma(k|t) M(x),\ k = 0, 1, ..., \infty \end{array} \tag{4.10}$$

holds true for each $x \in \mathbb{R}^n$, with the real $M(x) > 0$ any upper-bound to $\|H_c \Phi^k x\|, \forall k \geq 0$.

Definition 4.1. ***(Maximal Guaranteed Contraction Sequence)* -** *The sequence $\gamma^*(k|t) \leq 1, \forall k \geq 0$ is a Maximal Guaranteed Contraction Sequence for the pair (H_c, Φ) at time t if*

i. $\gamma^*(k|t)$ is a Guaranteed Contraction Sequence for the pair (H_c, Φ) at time t.

ii. $\gamma^*(k|t) \leq \gamma(k|t), \quad \forall k$, for all Guaranteed Contraction Sequences $\gamma(k|t)$ for the pair (H_c, Φ) at t.

A direct consequence of the above definitions is that if the command $g(0)$ computed at time $t = 0$ were constantly applied for the subsequent t steps, i.e. $g(0) = g(1) = ... = g(t-1)$, then, given a Maximal Guaranteed Contraction Sequence $\gamma^*(\cdot|t), t \in \mathcal{Z}_+$, the disturbance-free c-transient would be bounded as

$$\| H_c \Phi^k (\hat{x}(t) - x_{g(0)}) \| \leq \gamma^*(k|t) \rho(0), \quad \forall k \geq 0 \tag{4.11}$$

because of the following equalities

$$\Phi^t (\hat{x}(0) - x_{g(0)}) = (\hat{x}(t) - x_{g(0)}) = \left(\Phi^t \hat{x}(0) + \sum_{i=0}^{t-1} \Phi^{t-i-1} G g(0) - x_{g(0)} \right) \tag{4.12}$$

holding true. In chapter 3, the latter idea has been exploited to build up a FF-CG scheme where the command signal $g(t)$ is modified only every τ^* steps, being τ^* a Generalized Settling Time (Definition 3.1).

In this Chapter, we will overcome such a limitation as follows: consider at time t the disturbance-free c-transient evolution along the virtual horizon k assuming that a generic sequence of inputs $g(0), g(1), ..., g(t)$ has been applied from time $t = 0$

$$\begin{aligned}\overline{c}(k, \hat{x}(t), g(t)) &= c_g + H_c \Phi^k \left(\hat{x}(t) - x_{g(t)}\right) \\ &= c_g + H_c \Phi^k \left(\Phi^t \hat{x}(0) + \sum_{i=0}^{t-1} \left(\Phi^{t-i-1} G g(i)\right) - x_{g(t)}\right)\end{aligned} \tag{4.13}$$

The latter, by introducing the *translated command*

$$\Delta g(t) := g(t) - g(0) \tag{4.14}$$

may be rewritten as

$$\begin{aligned}&\overline{c}(k, \hat{x}(t), g(t)) = c_{g(t)} + H_c \Phi^k \left(\hat{x}(t) - x_{g(t)}\right) \\ &= c_{g(t)} + H_c \Phi^k \left(\Phi^t \hat{x}(0) + \sum_{i=0}^{t-1} \left(\Phi^{t-i-1} G \left(g(0) + \Delta g(i)\right)\right) - x_{g(0)} - x_{\Delta g(t)}\right) \\ &= c_{g(t)} + H_c \Phi^k \left(\Phi^t \hat{x}(0) + \left(\sum_{i=0}^{t-1} \Phi^{t-i-1} G g(0)\right) - x_{g(0)}\right) \\ &+ H_c \Phi^k \left(\sum_{i=0}^{t-1} \left(\Phi^{t-i-1} G \Delta g(i)\right) - x_{\Delta g(t)}\right)\end{aligned} \tag{4.15}$$

By recalling (4.11), the term depending from the initial conditions may be embedded as follows

$$\left(H_c \Phi^k \left(\Phi^t \hat{x}(0) + \left(\sum_{i=0}^{t-1} \Phi^{t-i-1} G g(0)\right) - x_{g(0)}\right)\right) \in \mathcal{B}_{\rho(0)\gamma^*(k|t)} \tag{4.16}$$

where the quantity $\gamma^*(k|t)\rho(0)$ represents an upper-bound to the effects of the initial conditions on the dynamics at time t. By definition, $\gamma^*(k|t) = 1$ for all k, t such that $k + t < \tau'$ and $\gamma^*(k|t) < 1$ for all k such that $k + t \geq \tau'$, τ' being the Minimal Generalized Settling time for the system (2.1). Then, it follows that $\gamma^*(k|t)\rho(0) < \rho(0), \forall k$ when $t \geq \tau'$. This inequality allows us to say that a sufficient condition for (4.8) to hold true is that

$$\left(c_{g(t)} + H_c \Phi^k \left(\sum_{i=0}^{t-1} \left(\Phi^{t-i-1} G \Delta g(i)\right) - x_{\Delta g(t)}\right)\right) \in \mathcal{C}_\infty \sim \mathcal{B}_{\rho(0)\gamma^*(k|t)} \tag{4.17}$$

By introducing now the *translated state*

$$\Delta x(t) = \left(\sum_{i=0}^{t-1} \Phi^{t-i-1} G \Delta g(i)\right) \tag{4.18}$$

it can be seen that it satisfies

$$\Delta x(t+1) = \Phi \Delta x(t) + G \Delta g(t) \tag{4.19}$$

under the assumption $\Delta x(0) = 0$. By using such a definition and remembering that $c_{g(t)} = c_{g(0)+\Delta g(t)} = c_{g(0)} + c_{\Delta g(t)}$, one may rewrite the sufficient condition (4.17) as

$$c_{g(0)} + \bar{c}(k, \Delta x(t), \Delta g(t)) \in \mathcal{C}_\infty \sim \mathcal{B}_{\rho(0)\gamma^*(k|t)} \forall k \geq 0. \tag{4.20}$$

Finally, we can denote

$$\mathcal{V}(\Delta x, \rho(\cdot)) := \{g \in \mathcal{W}_\delta \colon c_{g(0)} + \bar{c}(k, \Delta x, g - g(0)) \in \mathcal{C}_\infty \sim \mathcal{B}_{\rho(k)}, k \geq 0\}. \tag{4.21}$$

as the set of all *admissible* FF-CG commands g for a given sequence $\rho(k)$. On-line, at each time t, this sequence is instantiated as $\rho(k|t) = \rho(0)\gamma^*(k|t)$. Because $\mathcal{C}_\infty \sim \mathcal{B}_{\rho(k)}$ is a convex set and the predictions are linear, the latter results to be a convex and compact set. Then, by using a quadratic selection index, we may formulate the FF-CG algorithm as follows.

Algorithm 4.1.1 (*The FF-CG Algorithm*) REPEAT AT EACH TIME t

1.1 SOLVE

$$g(t) = \arg \min_{g \in \mathcal{V}(\Delta x(t), \rho(0)\gamma^*(\cdot|t))} \| g - r(t) \|_\Psi^2, \quad \Psi = \Psi' > 0 \tag{4.22}$$

1.2 APPLY $g(t)$
1.3 UPDATE (4.19)

The above FF-CG scheme enjoys the following properties.

Theorem 4.2. - *Let assumptions* **A1-A2** *be fulfilled. Consider system (2.1) along with the* ***FF-CG*** *selection rule (4.22) and let an admissible command signal $g(0) \in \mathcal{W}_\delta$ such that $c_{g(0)} \in \mathcal{C}_\infty \sim \mathcal{B}_{\rho(0)}$ be applied at time $t = 0$ where $\rho(0)$ is a known scalar such that*

$$\| H_c \Phi^k (x(0) - x_{g(0)}) \| \leq \rho(0), \ \forall k \geq 0 \tag{4.23}$$

Then:

1. *At each decision time t, the minimizer in (4.22) uniquely exists and can be obtained by solving a convex constrained optimization problem;*
2. *The system supervised by the FF-CG never violates the constraints, i.e. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$ regardless of any possible admissible disturbance realization $d(\cdot) \in \mathcal{D}$;*
3. *Let $g^{CG}(t) = \arg\min_{g \in \mathcal{V}(x(t))} \| g - r(t) \|_\Psi^2$ denote the standard CG solution as described in Chapter 2 for the disturbance-free CG design problem ($d(t) \equiv 0, \forall t$) where*

$$\mathcal{V}(x) := \{g \in \mathcal{W}_\delta : \bar{c}(k, x, g) \in \mathcal{C}, k \geq 0\}. \tag{4.24}$$

is the state-dependent admissible region and $\bar{c}(k,x,g)$ *defined in (3.3). Then, the time-varying regions of admissible commands* $\mathcal{V}(\Delta x(t),\rho(\cdot|t))$*, achieved by applying the FF-CG actions* $g(t)$ *at each time instant, asymptotically converge to (4.24) and* $\lim_{t\to\infty}(g(t)-g^{CG}(t))=0_m$*, regardless of the reference sequence* $r(t)$*;*

4. *The sequence of* $g(t)$*'s is bounded for any arbitrary bounded reference sequence* $r(t)\in\mathbb{R}^m$*. Moreover, whenever* $r(t)\equiv r$*, with* r *a constant set-point, the sequence of* $g(t)$*'s converges in finite time either to* r *or to its best admissible steady-state approximation* $\hat{r}$*:*

$$\exists t_s>0 \;\; t.c. \;\; g(t)=\hat{r}:=\arg\min_{g\in\mathcal{W}_\delta}\|g-r\|^2_\Psi, \forall t\geq t_s \tag{4.25}$$

and

$$\lim_{t\to\infty}\hat{x}(t)=x_{\hat{r}}, \qquad \lim_{t\to\infty}\hat{y}(t)=y_{\hat{r}}=\hat{r}, \qquad \lim_{t\to\infty}\hat{c}(t)=c_{\hat{r}}. \tag{4.26}$$

Proof

1) The existence of an admissible solution at each time t can be proved by simply remarking that $g(t)=g(t-1)$ is always an admissible, although not necessarily the optimal, solution for the prescribed problem at time t. Moreover, being the admissible region (4.21) convex and the cost function (4.22) strictly quadratic the optimal solution is unique.
2) At each time $t\geq 0$ the command $g(t)\in\mathcal{V}(\Delta x(t),\rho(\cdot|t))$ is applied to the plant. Observe that the set $\mathcal{V}(\Delta x(t),\rho(\cdot|t))$ has been built up so as to ensure the fulfilment of (4.20). This, because of (2.10), implies also that $c(k,x(t),g(t),d(\cdot))\subseteq\mathcal{C}$, $\forall d(\cdot)\in\mathcal{D}$, $\forall k\geq 0,\forall t\geq 0$. The statement is thus proved by simply noticing that the latter inclusion, being true $\forall k\geq 0$ is true in particular for $k=0$. Then, $c(t)\in c(0,x(t),g(t),d(\cdot))\subseteq\mathcal{C}$, $\forall d(\cdot)\in\mathcal{D},\forall t\geq 0$.
3) The main observation here is that, in the absence of disturbances, by combining (4.15), (4.16) and (4.20) we have that at each time instant the following inclusion holds true

$$\bar{c}(k,x(t),g(t))=\bar{c}(k,\hat{x}(t),g(t))\in\bar{c}(k,\Delta x(t),\Delta g(t))\oplus c_{g(0)}\oplus\mathcal{B}_{\rho(\cdot|t)} \tag{4.27}$$

where $\rho(\cdot|t):=\rho(0)\gamma^*(k|t)$ and the $+$ operator denotes the Pontryagin sum. Moreover, consider the evolution of the sequence $\rho(k|t)$ in the set $\mathcal{V}(\Delta x(t),\rho(\cdot|t))$. Being $\gamma^*(k|t)$ a Maximal Guaranteed Contraction Sequence, there exists a finite time t' such that $\rho(k|t)=\rho(0)\gamma^*(k|t)\leq\alpha\delta,\forall t\geq t',\forall k\geq 0$, with $\alpha\in(0,1)$. Then, if $t>t'$, a non-empty admissible region $\mathcal{V}(\Delta x(t),\alpha\delta)$ exists such that

$$\mathcal{V}(\Delta x(t),\rho(\cdot|t))\supseteq\mathcal{V}(\Delta x(t),\alpha\delta),\forall t>t'. \tag{4.28}$$

In order to prove the statement, we have to note that $\rho(0)\gamma^*(k|t)\to 0$ for $t\to\infty$ under the action of the FF-CG action $g(t)$. Therefore, because of (4.27)

$$\bar{c}(k, x(t), g(t)) = \bar{c}(k, \Delta x(t), \Delta g(t)) + c_{g(0)} \tag{4.29}$$

and the equation (4.28) takes the following form

$$\begin{aligned}\lim_{t\to\infty} \mathcal{V}(\Delta x(t), \rho(\cdot|t)) &= \{g \in \mathcal{W}_\delta : \bar{c}(k, \Delta x(t), \Delta g(t)) + c_{g(0)} \in \mathcal{C}, k \geq 0\} \\ &= \{g \in \mathcal{W}_\delta : \bar{c}(k, x(t), g(t)) \in \mathcal{C}, k \geq 0\} \\ &= \mathcal{V}(\Delta x(t), 0) = \mathcal{V}(x(t)).\end{aligned} \tag{4.30}$$

4) The boundedness of $g(t), \forall t \leq 0$, follows by the coercivety and strict convexity of the cost function under bounded $r(t)$. In fact, for all $r(t)$ bounded, if $g(t) = \infty$ were the optimum it would imply $\|g(t) - r(t)\|_\Psi = \infty$. However, this is not possible because the existence of an admissible and bounded $g(0)$ guarantees at time t the existence of a bounded admissible, although not optimal, solution $g(t)$. In order to prove the finite-time convergence when $r(t)$ is constant it is worth recalling, from the above item 3, the existence of the set $\mathcal{V}(\Delta x(t), \alpha\delta)$, $\alpha \in (0, 1)$ ensuring (4.28) which allows one to approximate the FF-CG problem for all $t > t'$ as

$$g(t) = \arg \min_{g \in \mathcal{V}(\Delta x(t), \alpha\delta)} \| g - r(t) \|_\Psi^2 . \tag{4.31}$$

Furthermore, let us introduce the following properties holding true for linear and asymptotically stable systems: first, one has that

$$g(t) \to g \in \mathbb{R}^m \Rightarrow \lim_{t\to\infty} \bar{c}(t) = c_g, \tag{4.32}$$

second, $\forall x(0) \in \mathbb{R}^n$ when $g(t) \equiv g,\ \forall t, \forall \lambda > 0$ there exists a real $\beta(\lambda) > 0$ such that

$$\|c(0) - c_g\| \leq \beta(\lambda) \Rightarrow \|\bar{c}(t) - c_g\| \leq \lambda, \forall t \geq 0, \forall g \in \mathcal{W}^\delta. \tag{4.33}$$

Next, let $g(t)$ be the FF-CG solution at time t of (4.31) with $r(t) \equiv r$. As already discussed, $g(t)$ is still an admissible, though not necessarily the optimal, solution at time $t + 1$. Hence, the sequence of cost $||g(t) - r||_\Psi^2$ is non increasing, i.e. $\| g(t + 1) - r \|_\Psi^2 \leq \| g(t) - r \|_\Psi^2$. As consequence, the sequence of solutions $g(t)$ converges to g_∞ that is necessarily $g_\infty = \hat{r}$. In fact, if $g_\infty \neq \hat{r}$, because of strict convexity of $\| \cdot \|_\Psi^2$ it would be, for a certain $\epsilon \in (0, 1)$

$$\|g_\infty + \epsilon(\hat{r} - g_\infty) - r\|_\Psi^2 < \|g_\infty - r\|_\Psi^2. \tag{4.34}$$

By exploiting property (4.32) for system (2.1), for $t \to \infty$, $\|\bar{c}(t) - c_{g_\infty}\| = 0$ which would implies the sure existence of an $\epsilon > 0$ such that $\|\bar{c}(t) - c_{g_\infty + \epsilon(\hat{r} - g_\infty)}\| \leq (1 - \alpha)\delta$. Then the FF-CG would be capable to switch from g_∞ to the better command $g_\infty + \epsilon(\hat{r} - g_\infty)$ and g_∞ would not be a convergence point. Now let's define

$$\Delta c(t) := H_c \Delta x(t) + L \Delta g(t) \tag{4.35}$$

By taking into account properties (4.32) and (4.33) for system (4.19), and by notice that $g(t) \to \hat{r} \Rightarrow \Delta g(t) \to \hat{r} - g(0)$, we can state that it exists a finite time $t'' > t'$ such that $\|\Delta c(t'') - c_{\hat{r}-g(0)}\| < \beta((1-\alpha)\delta)$ which implies $\|\Delta c(t) - c_{\hat{r}-g(0)}\| \leq (1-\alpha)\delta, \forall t \geq t''$, or equivalently that

$$\Delta c(t) - c_{\hat{r}-g(0)} \in \mathcal{B}_{(1-\alpha)\delta}, \forall t \geq t''. \tag{4.36}$$

The latter indicates that predictions for $\bar{c}$ along virtual time k, starting from $\Delta x(t'')$, will satisfy

$$\bar{c}(k, \Delta x(t''), \hat{r} - g(0)) - c_{\hat{r}-g(0)} \in \mathcal{B}_{(1-\alpha)\delta}, \forall k \geq 0 \tag{4.37}$$

Since $c_{\hat{r}} \in \mathcal{C}_\infty \sim \mathcal{B}_\delta$, if we add $c_{\hat{r}}$ to the right side of the latter we will obtain

$$c_{g(0)} + \bar{c}(k, \Delta x(t''), \hat{r} - g(0)) \in \mathcal{C}_\infty \sim \mathcal{B}_\delta \oplus \mathcal{B}_{(1-\alpha)\delta}, \forall k \geq 0 \tag{4.38}$$

that becomes

$$c_{g(0)} + \bar{c}(k, \Delta x(t''), \hat{r} - g(0)) \in \mathcal{C}_\infty \sim \mathcal{B}_{\alpha\delta}, \forall k \geq 0. \tag{4.39}$$

Then, because $\hat{r} \in \mathcal{V}(\Delta x(t''), \alpha\delta)$, the FF-CG will select for sure $\hat{r}$ at finite time $t_s = t''$. Finally, conditions (4.26) simply follow by asymptotic stability of (2.1).

4.2 Computational details

4.2.1 Maximal Guaranteed Contraction Sequence $\gamma(\cdot|t)$

For what regards the Maximal Guaranteed Contraction Sequence, in principle, one should determine any possible sequence $\gamma^*(\cdot|t)$ for every t. However, interesting enough, the following recursive property holds true

$$\gamma^*(k|t) = \gamma^*(k+t|0), \forall k \tag{4.40}$$

and only $\gamma^*(k|0)$ has to be computed in practice. Moreover, by inheriting the technicalities introduced in Subsection 3.2.1 for the computation of the Generalized Settling Time, the computation of $\gamma^*(\cdot|t)$ may be performed as follows:

Algorithm 4.2.1 ***Maximal Guaranteed Contraction Sequence Algorithm***

1.1 SET $\gamma^*(0|0) = 1$;
1.2 SET $\epsilon > 0$;
1.3 REPEAT FOR EACH K>0
  1.3.1 SET $\bar{\gamma} = \gamma^*(k-1|0)$

*1.3.2*SET $\underline{\gamma} = 0$
*1.3.3*REPEAT UNTIL $\bar{\gamma} - \underline{\gamma} > \epsilon$
1.3.3.1 SET $\gamma^*(k|0) := (\underline{\gamma} + \bar{\gamma})/2$
1.3.3.2 FIND GENERALIZED SETTLING TIME τ WITH PARAMETER $\gamma^*(k|0)$ AS IN SUBSECTION 3.2.1
1.3.3.3 IF $\tau > k$
1.3.3.3.1 SET $\underline{\gamma} := \gamma^*(k|0)$
1.3.3.4 ELSE
1.3.3.4.1 SET $\bar{\gamma} := \gamma^*(k|0)$
1.3.3.5 GO TO (3.1)

Observe also that the computation should be done for any k. Then, from a practical point of view it is only possible to compute a finite sub-sequence of $\gamma^*(k|0)$. However, the proof Theorem 4.2 shows that any approximating Guaranteed Contraction Sequence $\gamma(k|0)$ such that $\lim_{k\to\infty} \gamma(k|0) = 0$ may be used in the place of $\gamma^*(k|0)$ at the price of introducing some conservativeness in the plant start-up phase of the algorithm but without affecting its feasibility properties. Then, an alternative practicable procedure is then that of computing offline and storing only the first k' samples of $\gamma^*(k|0)$ and approximating the tail with the exponentially decreasing sequence directly derived from (3.38)

$$\gamma(k|0) = \bar{\sigma}(H_c^o)B\lambda^k\bar{\sigma}\left\{\left((\Theta^o)^T\Theta^o\right)^{-1}(\Theta^o)^T\right\}M(x^o)\sqrt{n}, k > k'. \tag{4.41}$$

4.2.2 Linear constraints: The disturbance-free case

In this section we will present all computational details for the design and the implementation of the above FF-CG strategy in the disturbance-free case $d(t) = 0,\ \forall t$. Because, also in this case, linear constraints for $c(t) \in \mathcal{C}$ are assumed, they can be represented as in (2.18)-(2.21) of Section 2.3.1 from which we inherit the usual expression for $\mathcal{C}_\delta$ and $\mathcal{W}^\delta$

$$\mathcal{C}_\delta = \{c \in \mathbb{R}^{n_c} \ : \ T_i'c \le q_i - \delta\sqrt{T_i'T_i},\ i = 1, ..., n_{v_c}\} \tag{4.42}$$

$$\mathcal{W}_\delta = \{w \in \mathbb{R}^m : \ T\left(H_c(I - \Phi)^{-1}G + L\right)g \le q - \delta[\sqrt{T_i^TT_i}]\} \tag{4.43}$$

It remains to characterize the set $\mathcal{V}(\Delta x, \rho(\cdot))$ which, in the disturbance-free case, is defined as

$$\mathcal{V}(\Delta x, \rho(\cdot)) := \{g \in \mathcal{W}_\delta \colon c_{g(0)} + \bar{c}(k, \Delta x, g - g(0)) \in \mathcal{C} \sim \mathcal{B}_{\rho(k)}, k \ge 0\} \tag{4.44}$$

where the integer k_0 is computed according to Algorithm 2.3.1. It is also easy to understand that

$$\begin{aligned}\Delta\bar{x}(k,\Delta x,g-g(0)) &= \Phi^k\Delta x+\left(\sum_{i=0}^{k-1}\Phi^iG\right)(g-g(0))\\ &= \Phi^k\Delta x+R_k^x(g-g(0)) \qquad (4.45)\\ \bar{c}(k,x,g) &= H_c\Delta\bar{x}(k,\Delta x,g-g(0))+L(g-g(0))\\ &= H_c\Phi^k\Delta x+(H_cR_k+L)(g-g(0))\\ &= H_c\Phi^k\Delta x+R_k^c(g-g(0)) \qquad (4.46)\end{aligned}$$

and hence

$$\begin{aligned}\mathcal{V}(\Delta x,\rho(\cdot)) = \{g\in\mathcal{W}^\delta : T\left(H_c(I-\Phi)^{-1}G+L\right)g(0)+TH_c\Phi^k\Delta x+\\ +TR_k^c(g-g(0))\le q-\gamma(k|0)\rho(0)[\sqrt{T_i^TT_i}],\ k=0,...,k_0\}\end{aligned} \qquad (4.47)$$

Finally, the CG action computation consists of solving the following QP optimization problem

$$\begin{aligned}g(t) &= \min_g(g-r(t))^T\Psi(g-r(t))\\ &\text{subject to}\\ T\left(H_c(I-\Phi)^{-1}G+L\right)g(0)+TH_c\Phi^k\Delta x(t)+TR_k^c(g-g(0)) &\le q\\ &\quad-\gamma(k+t|0)\rho(0)[\sqrt{T_i^TT_i}],\ k=0,...,k_0\\ T\left(H_c(I-\Phi)^{-1}G+L\right)g &\le q-\delta[\sqrt{T_i'T_i}] \qquad (4.48)\end{aligned}$$

where the sequence $\gamma^*(k|0)$ can be computed according to the Algorithm 4.2.1.

4.2.3 Linear constraints: The disturbance acting case

We assume here the same characterization used in Section 2.3.2 for bounded disturbances $d(t)$. By means of recursions (2.37)-(2.46), we have the following approximation for $\mathcal{C}_\infty$

$$\mathcal{C}_\infty \simeq \{c\in\mathbb{R}^{n_c} \ : \ Tc\le q^{k_\varepsilon}\} \qquad (4.49)$$

In this case, $\mathcal{V}(x)$ can be characterized as

$$\begin{aligned}\mathcal{V}(\Delta x,\rho(\cdot)) = \{g\in\mathcal{W}^\delta : T\left(H_c(I-\Phi)^{-1}G+L\right)g(0)+TH_c\Phi^k\Delta x+\\ +TR_k^c(g-g(0))\le q^{k_\varepsilon}-\gamma(k+t|0)\rho(0)[\sqrt{T_i^TT_i}],\ k=0,...,k_0\}\end{aligned} \qquad (4.50)$$

Finally, the CG action computation consists of solving the following QP optimization problem

$$\begin{aligned}g(t) &= \min_g(g-r(t))^T\Psi(g-r(t))\\ &\text{subject to}\\ T\left(H_c(I-\Phi)^{-1}G+L\right)g(0)+TH_c\Phi^k\Delta x(t)+TR_k^c(g-g(0)) &\le q^{k_\varepsilon}\\ &\quad-(\varepsilon+\gamma(k|0)\rho(0))[\sqrt{T_i^TT_i}],\ k=0,...,k_0\\ T\left(H_c(I-\Phi)^{-1}G+L\right)g &\le q^{k_\varepsilon}-\delta[\sqrt{T_i'T_i}] \qquad (4.51)\end{aligned}$$

The computation of the constraint horizon k_0 can be accomplished via the Algorithm 2.3.2.

4.3 Simulation Studies

4.3.1 Example 1

In this subsection we show the effectiveness of the improved FF-CG presented in this chapter by comparing it with all other strategies on the same system (3.66)-(3.66) already analyzed in Section 3.3. In order to distinguish it from other FF-CG based techniques, the improved one will be referred hereafter as FF-CG2. Also in this case the standard Matlab FMINCON.M routine has proved sufficient to solve the resulting quadratic optimization. As before, the task of the CG is to ensure good set-point tracking for $r(t) = 0.38$ and maintain the constrained output c in the set (3.68).

The unique FF-CG2 free parameter is δ, that affects constraint horizon $k_0 - 1$ and consequently the evaluation of $\gamma(\cdot, t)$, or equivalently that of $\gamma(k, 0), k \geq 0$ as discussed in Section (4.2). Figure 4.1 shows $\gamma(k, 0), k \in \{0, ..., 22\}$ for different value of δ.

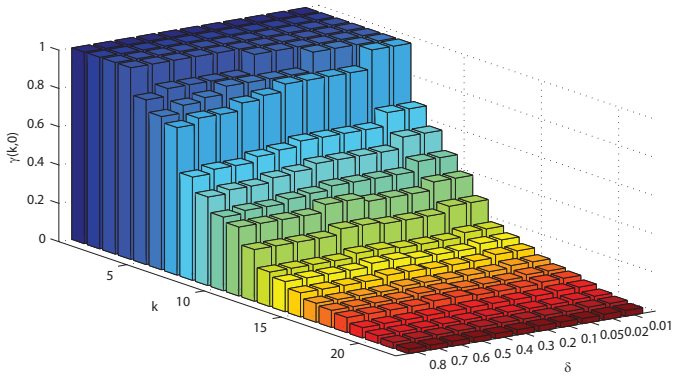

Fig. 4.1. Maximal Guaranteed Contraction Sequence $\gamma(k|0)$ computed via (4.2.1) as a function of δ and k.

As expected, for fixed δ and increasing values of k, $\gamma(k, 0)$ becomes increasingly lower. On the contrary, by fixing k and for increasingly smaller values for δ, $\gamma(k, 0)$ is increasingly larger. Thus, it is no convenient to select δ very small because we would be constrained to use a sequence $\gamma(k, 0)$ that equals 1 for many k after $k = 0$. In the following simulations, we undertook comparisons amongst the present FF-CG2 scheme, the standard CG and FF-CG related to Algorithm (3.2.2). In all these schemes we used the same parameters $\delta = 0.05$ and $\gamma = 0.7$, which involve the use of $\tau = 7$. The initial supposed uncertainty was $\rho(0) = 0.2$ and the initial applied command was $g(0) = -0.2$.

In this case the optimization problem to solve on-line is related with Algorithm (4.1.1) and problem (4.48). For this example, it results to be given by

$$\begin{aligned} & g(t) = \arg\min_g (g - r(t))^T \Psi (g - r(t)) \\ s.t. \begin{cases} T\left(H_c(I-\Phi)^{-1}G\right) g \leq q - 0.05[\sqrt{T_i' T_i}] \\ T\left(H_c(I-\Phi)^{-1}G\right) g(0) + T H_c \Phi^k \Delta x(t) + T R_k^c (g - g(0)) \leq q \\ \qquad\qquad - \gamma(k+t|0)\rho(0)[\sqrt{T_i^T T_i}],\ k = 0, ..., 5 \end{cases} \end{aligned} \tag{4.52}$$

where the matrices appearing in this formulation take the following values

$$H_c(I-\Phi)^{-1}G = \begin{bmatrix} 1.0008 \\ 7.6864 \end{bmatrix}$$

$$\begin{aligned} H_c\Phi^0(I-\Phi)^{-1}G &= \begin{bmatrix} 1.0008 \\ 7.6864 \end{bmatrix},\ H_c\Phi^1(I-\Phi)^{-1}G = \begin{bmatrix} 1.8943 \\ 6.6864 \end{bmatrix}, \\ H_c\Phi^2(I-\Phi)^{-1}G &= \begin{bmatrix} 2.2467 \\ 5.1462 \end{bmatrix},\ H_c\Phi^3(I-\Phi)^{-1}G = \begin{bmatrix} 2.1907 \\ 3.4443 \end{bmatrix}, \\ H_c\Phi^4(I-\Phi)^{-1}G &= \begin{bmatrix} 1.8681 \\ 1.8554 \end{bmatrix},\ H_c\Phi^5(I-\Phi)^{-1}G = \begin{bmatrix} 1.4089 \\ 0.5490 \end{bmatrix}, \end{aligned}$$

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix},\ q = \begin{bmatrix} 0.4 \\ 3 \\ 0.4 \\ 3 \end{bmatrix},$$

$$\begin{aligned} R_0^c &= \begin{bmatrix} -0.8935 \\ 1.0000 \end{bmatrix},\ R_1^c = \begin{bmatrix} 1.8943 \\ 6.6864 \end{bmatrix}, \\ R_2^c &= \begin{bmatrix} -1.2460 \\ 2.5402 \end{bmatrix},\ R_3^c = \begin{bmatrix} -1.1899 \\ 4.2421 \end{bmatrix}, \\ R_4^c &= \begin{bmatrix} -0.8674 \\ 5.8310 \end{bmatrix},\ R_5^c = \begin{bmatrix} -0.4081 \\ 7.1374 \end{bmatrix}. \end{aligned}$$

The simulations for the CG governed system are reported in Figure 4.2. In this figure, the classical CG solution (CG in the figures), the FF-CG2 Algorithm (4.1.1) and FF-CG Algorithm (3.2.2) are considered. The performance of the CG and FF-CG2 algorithms are almost identical while the FF-CG holds a certain degree of conservativeness. Moreover in Figure 4.4 the *uncertainty* $\rho(t)$ computed by FF-CG2 and FF-CG are depicted. As shown in that Figure, the uncertainty related to FF-CG2 never growths and quickly goes to zero w.r.t. uncertainty related to FF-CG. In Figure 4.3, the constrained variables are depicted: in this case they always satisfy the respective constraints.

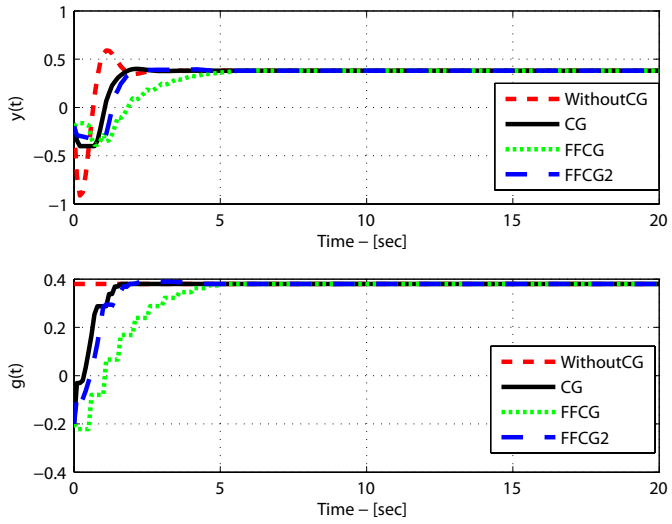

Fig. 4.2. (a)[upper] Response of the system with different CG action. (b)[down] Computed references

It is worth noticing that the trajectories of the system acted by standard CG and FFCG2 are very close each other and both are more active than those resulting under the FFCG scheme. Finally, in the following table

	CG	FF-CG	FF-CG2
CPU Time per step [ms]	30.0	27.2	28.1

the on-line computational burdens per step of all schemes are reported. The three strategies have a similar computational behavior.

4.3.2 Example 2: Position servomechanism

In this example further comparisons are reported with the aim of showing also the robusteness properties of the algorithms when exogenous disturbances are present. The proposed FF-CG schemes are applied to the position servomechanism schematically described in Figure 4.5. This consists of a DC-motor, a gear-box, an elastic shaft and an uncertain load. Technical specifications involve bounds on the shaft torsional torque T as well as on the input voltage V. System parameters are reported in Table 4.1.

Let θ_M and θ_L denote respectively the motor and the load angle and let

$$x_p = \left[\, \theta_L \;\; \dot{\theta}_L \;\; \theta_M \;\; \dot{\theta}_M \,\right]^T$$

be a suitable state vector. Then, the plant can be described by the following state-space model

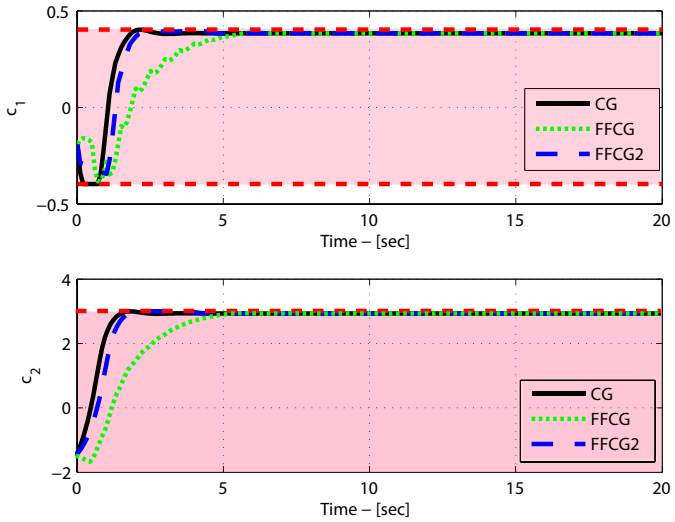

Fig. 4.3. Constrained output

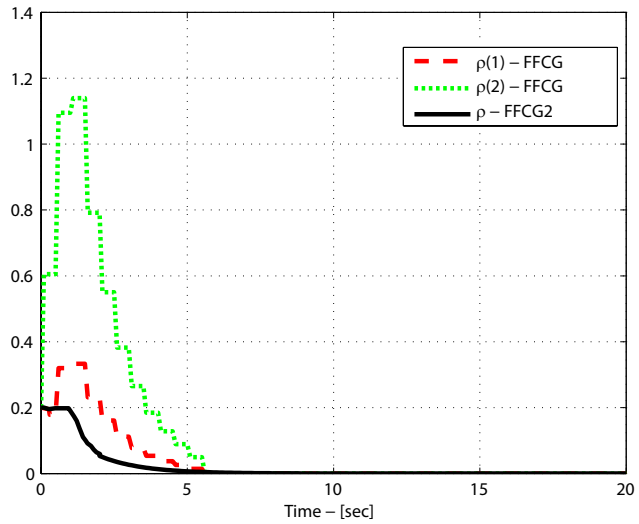

Fig. 4.4. Evolution of $\rho(t)$ (blue line) for FFCG, and $rho_1(t)$ and $\rho_2(t)$ for FFCGBox

$$\begin{cases} \dot{x}_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_\theta}{J_L} & -\frac{\beta_L}{J_L} & \frac{k_\theta}{\rho J_L} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_\theta}{\rho J_M} & 0 & -\frac{k_\theta}{\rho^2 J_M} & -\frac{\beta_M + k_T^2/R}{J_M} \end{bmatrix} x_p + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_T}{R J_M} \end{bmatrix} V \\ \theta_L = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x_p, \; T = \begin{bmatrix} k_\theta & 0 & -\frac{k_\theta}{\rho} & 0 \end{bmatrix} x_p \end{cases}$$

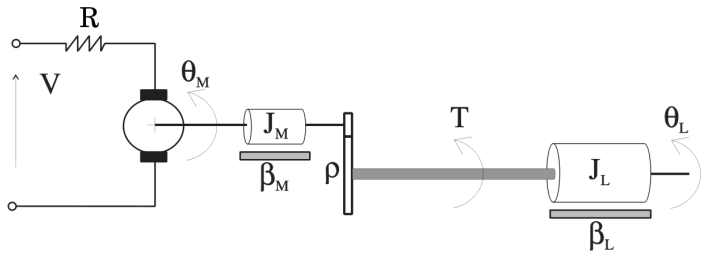

Fig. 4.5. Servomechanism model

Table 4.1. Model parameters

Symbol	Value (MKS)	Meaning
Symbol	Value (MKS)	Meaning
L_S	1.0	shaft length
d_S	0.02	shaft diameter
J_S	negligible	shaft inertia
J_M	0.5	motor inertia
β_M	0.1	motor viscous friction coefficient
R	20	resistance of armature
K_T	10	motor constant
ρ	20	gear ratio
k_θ	1280.2	torsional rigidity
J_L	$20J_M$	load inertia
β_L	25	load viscous friction coefficient
T_s	0.1	sampling time

Because the steel shaft has a finite shear strength, a maximum admissible shaft $\tau_{adm} = 50N/mm^2$ imposes the constraint $|T| \leq 78.5398\ Nm$ on the torsional torque. Moreover, the input DC voltage V has to be constrained within the range $|V| \leq 220\ V$. The model is discretized with a sampling step of $T_s = 0.1s$ and by using a zero-order holder on the input voltage terminal. It is assumed that the motor is provided with a control unit acting on the motor voltage and guaranteeing to guarantee assumptions **A1-A2**. It is also assumed that the closed-loop system state, input and output are not accessible to the CG purposes and only the manipulation of the set-point signal is allowed.

Simulations reveal that the pre-compensated system, when not governed by a CG unit, exhibits a very fast response but inadmissible input voltages and torsional torques for the references of interest, as shown in Figure. 4.6.b for a square-wave set-point with amplitude equal to $r = 60\ deg$ (solid line) and increasing frequency. On the contrary, when a FF-CG unit is used, the torque and voltage constraints happen to be fulfilled. Figure 4.7 shows the resulting system output (4.7.a) and the computed FF-CG action (4.7.b) for the same set-point of Figure 4.6. In these figures, the performance of the FF-CG and CG strategies can be compared. The comparison involves the more conservative FF-CG technique described in Algorithm 3.2.2, performed for a generalized

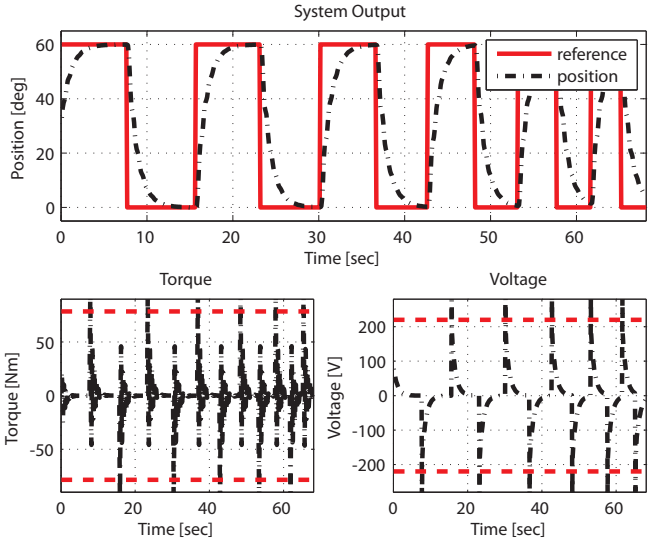

Fig. 4.6. System without CG unit: a) Position b) Constrained variables: Torsional Torque (left), voltage (Right)

settling time $\tau = 7$ denoted as "FF-CG" and the FF-CG technique related to Algorithm 4.1.1 here referred to "FF-CG2". One can observe that the level of conservativeness introduced by the FF-CG2 version is negligible after few istants when contrasted with the standard state-feedback CG approach. Moreover, FF-CG introduces a certain level of delay in the system response which is not present on the contrary in FF-CG2. In Figure 4.8, the constrained variables are depicted: in this case, the voltage inputs and torsional torques are admissible. It is worth pointing out that the trajectories of the system controlled by the standard CG and the proposed FF-CG2 strategies almost coincide. On the contrary, the trajectories produced by FF-CG are delayed of seven sample steps.

In order to show the effectiveness and performance robustness of FF-CG schemes when disturbances are present, we consider the severe case whereby $d(t)$ is a square-wave with amplitude $d_{max} = 0.007N$ (Figures 4.9-4.10). From the above figures it can be observed that both the FF-CG algorithms keep working adequately also if the state is brought far from the equilibrium by the disturbances. In particular, the constraints are always satisfied and the tracking performance is achieved. It is fair also pointing out that, in the simulations, the applied disturbance is unrealistically high, producing perturbation on the nominal torque as high as $40Nm$, when the maximum admissible torque is $78.5398Nm$. With smaller and smoother disturbance signals, the system controlled by both the FF-CG strategies presents a behavior similar to the one

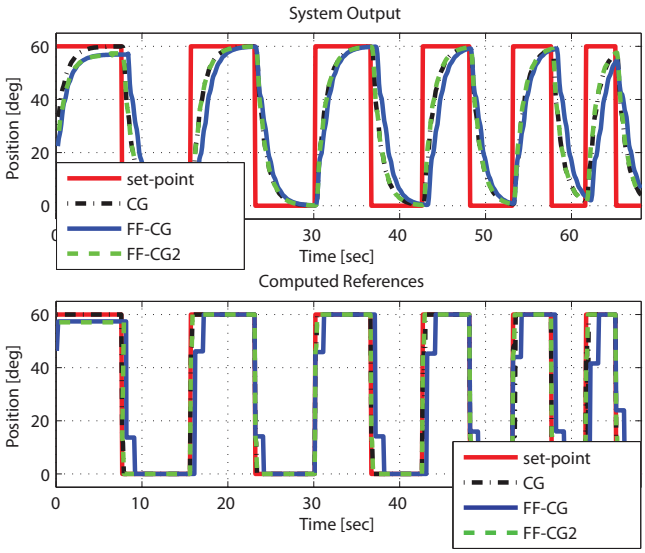

Fig. 4.7. System with CG unit. a) System Output: set-point (solid), CG (dash-dotted), FF-CG (dash) b) Computed Reference: set-point (solid), CG (dash-dotted), FF-CG (dash)

shown in Figures 4.7-4.8.

A further analysis on the performance robustness under disturbances for all FF-CG and CG schemes has been performed and the results have been depicted in Fig. 4.11-(upper), where performance-degradation vs disturbance-size plots are reported. In this respect, performance comparisons are expressed by the cost $J = \frac{1}{T}\sum_{k=0}^{T} |r(k) - \theta_L(k)|$ computed by executing simulations with square-wave disturbances characterized by different amplitudes d_{max}, with T the number of steps of the simulation (in this case T=400). As a result, the performance of the standard CG unit is only weakly affected by the disturbance size whereas the performance of the other two schemes monotonically degrades with the increment of the amplitude of the disturbance. Note also that for $d_{max} > 0.01N$ the standard CG design problem is unfeasible whereas the other algorithms are unfeasible for lower values: FF-CG is unfeasible for $d_{max} > 0.009N$ and FF-CG2 for $d_{max} > 0.01N$ respectively. A different way to see performance degradation due to disturbances is depicted in Fig. 4.11-(lower), where the output of the system acted by the FF-CG2 is reported for different values for d_{max}.

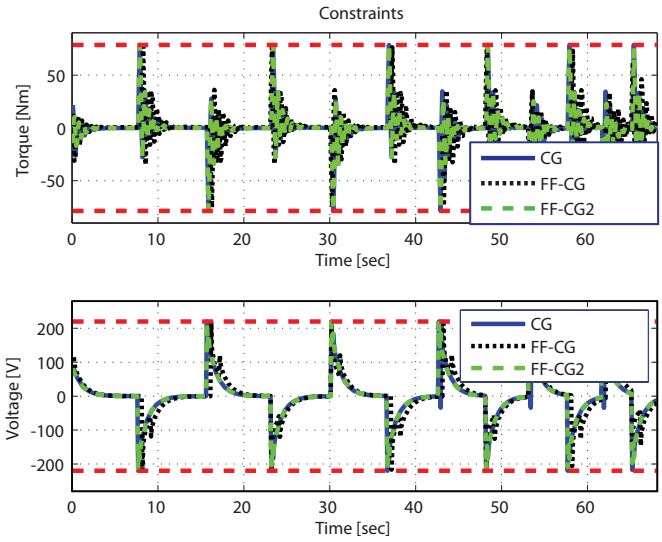

Fig. 4.8. Constrained variables. a) [Upper] Torque. b) [Lower] Voltage.

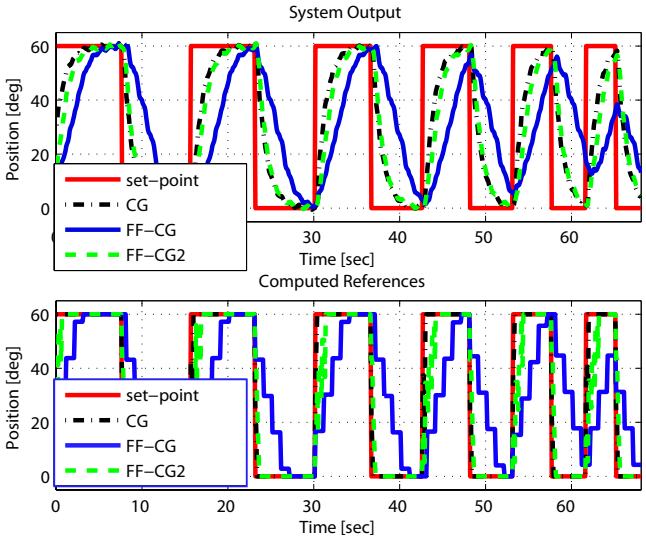

Fig. 4.9. System with CG unit. a) [Upper] System Output. b) [Lower] Computed Reference in the disturbance acting case.

4.4 Conclusions

In this Chapter, an novel more effective FF-CG scheme than the one presented in Chapter 3 is proposed which again does not make use of any measure of the

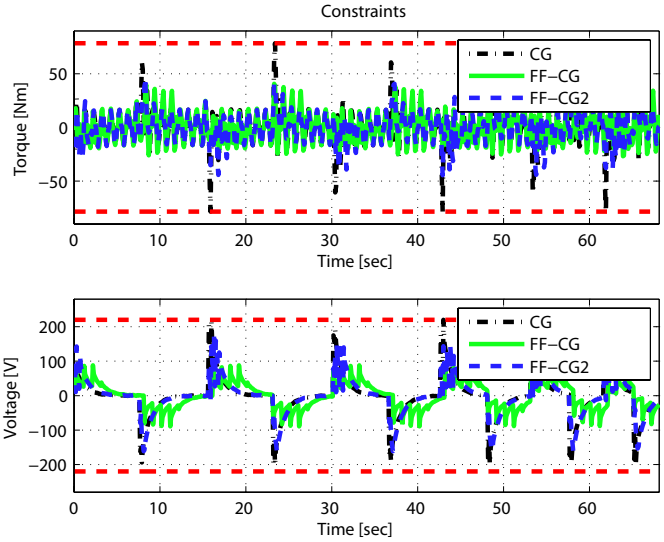

Fig. 4.10. Constrained variables in the disturbance acting case. a) [Upper] Torque. b) [Lower] Voltage.

state to govern the set-point manipulations. The main idea under its development was to limit the set-point variations in order to always maintain the state

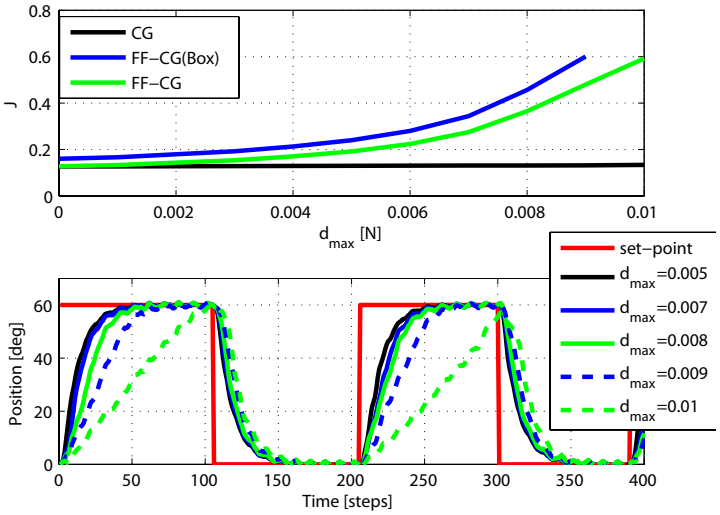

Fig. 4.11. a)[Upper]Performance degradation vs. disturbance sizes. b)[Lower] FF-CG2 performance degradation vs. disturbance sizes

trajectory "not too far" from the region of the steady-state admissible equilibria. The properties of the proposed algorithm have been carefully analyzed and the differences with standard CG approaches pointed out. Comparisons with classical CG and previously proposed FF-CG solutions have also been presented and discussed in the final examples.

Part II

Distributed FFCG and CG Schemes

5

Non-Iterative Feed-Forward Command Governor Distributed Schemes

The problem of interest here and in the next Chapters is the design of distributed supervision strategies based on Command Governor (CG) ideas for multi-agent systems where a centralized coordination unit cannot be used because of unrealistic or unavailable communication infrastructures. In particular, in this Chapter we discuss two distributed strategies to solve the above described supervision task in large scale applications based on the FF-CG ideas introduced in Chapter 3. It will be clear that the main advantages of the derived distributed FF-CG schemes consist in their low communication rates required for their implementation, remarkably lower than other distributed approaches - e.g those based on MPC predictive ideas and consensus mechanisms.

The first proposed scheme (Sequential-FFCG, S-FFCG) is a sequential algorithm in the sense that only one agent at a time is allowed to modify its own reference signal. This approach, although behaving increasingly slower for a rising number of agents, is anyway of interest in all situations whereby the coordination problem consists of few and slow set-point adjustments, e.g. in all small/medium-scale situations where the set-points change infrequently or slower than the system dynamics.

Such a method is also instrumental to build up a faster "parallel" version of the scheme (Parallel-FFCG, P-FFCG) where, whenever possible, all agents are allowed to modify their own reference signals simultaneously. The key point of this parallel scheme is the on-line determination of a suitable Cartesian inner approximation of the global constraint set [57], allowing the agents to optimize independently and simultaneously their reference signals and ensuring the fulfilment of the global constraints at the price of a slight optimality loss.

It is important to remark that the distributed implementation of the FF-CG scheme introduces additional technical challenges not present in the centralized solution which have to be carefully managed. In particular, as far as the sequential distributed scheme is concerned, verifiable sufficient conditions on the constraint set are proposed whose fulfilment ensures that the *liveliness* of the scheme is preserved. In fact, it will be shown that under gen-

eral constraint structures the proposed distributed supervising scheme may get stuck and fail to update the solution because, unlike in the centralized scheme, the agents are only allowed to update their commands one at the time and some directions in the solution space are then precluded during the minimization process. This restriction may lead to deadlock situations and prevent the agents from being able to modify their local set-points altogether. Such an analysis involves the verification of a particular geometrical property, hereafter referred as *Constraints Qualification* (CQ), for all points belonging to the boundary of the constrained region. To this end, a numerical procedure is also proposed and it is shown that it is possible to determine arbitrarily accurate inner approximations of the prescribed constrained region (via a multi-box approach) which, although only in single-input problems $g_i \in \mathbb{R}$, result *Constraints Qualified* (CD) by construction thus avoiding deadlock situations from occurring.

The Chapter is organized as follows: in Section 5.1 the system under consideration is described and the design problem formulated, in Section 5.2 the S-FFCG sequential strategy is fully described and analyzed, in Section 5.3 the geometrical characterization of the structure of the constraints set is exploited in order to understand its Constraint Qualification and establish if deadlocks may occur. In Section 5.4 the parallel distributed version of the FF-CG is described. Section 5.5 concludes the Chapter.

5.1 System description and Problem Formulation

Consider a set of N subsystems $\mathcal{A} = \{1, \ldots, N\}$, each one being a LTI closed-loop dynamical system regulated by a local controller which ensures stability and good closed-loop properties when the constraints are not active (small-signal regimes when the coordination is effective). Let the i-th closed-loop subsystem be described by the following discrete-time model

$$\begin{cases} x_i(t+1) = \Phi_{ii} x_i(t) + G_i g_i(t) + \sum_{j \in \mathcal{A}-\{i\}} \Phi_{ij} x_j(t) \\ y_i(t) \quad = H_i^y x_i(t) \\ c_i(t) \quad = H_i^c x(t) + L_i g(t) \end{cases} \tag{5.1}$$

where: $t \in \mathbb{Z}_+$, $x_i \in \mathbb{R}^{n_i}$ is the state vector (which includes the controller states under dynamic regulation), $g_i \in \mathbb{R}^{m_i}$ the manipulable reference vector which, if no constraints (and no CG) were present, would coincide with the desired reference $r_i \in \mathbb{R}^m$ and $y_i \in \mathbb{R}^{m_i}$ is the output vector which is required to track r_i. Finally, $c_i \in \mathbb{R}^{n_i^c}$ represents the local constrained vector which has to fulfill the set-membership constraint

$$c_i(t) \in \mathcal{C}_i, \ \forall t \in \mathbb{Z}_+, \tag{5.2}$$

$\mathcal{C}_i$ being a convex and compact polytopic set. It is worth pointing out that, in order to possibly characterize global (coupling) constraints amongst states

of different subsystems, the vector c_i in (5.1) is allowed to depend on the aggregate state and manipulable reference vectors $x = [x_1^T, \ldots, x_N^T]^T \in \mathbb{R}^n$, with $n = \sum_{i=1}^N n_i$, and $g = [g_1^T, \ldots, g_N^T]^T \in \mathbb{R}^m$, with $m = \sum_{i=1}^N m_i$. Moreover, we denote by $r = [r_1^T, \ldots, r_N^T]^T \in \mathbb{R}^m$, $y = [y_1^T, \ldots, y_N^T]^T \in \mathbb{R}^m$ and $c = [c_1^T, \ldots, c_N^T]^T \in \mathbb{R}^{n^c}$, with $n^c = \sum_{i=1}^N n_i^c$, the other relevant aggregate vectors. The overall system arising by the composition of the above N subsystems can be described as

$$\begin{cases} x(t+1) = \Phi x(t) + Gg(t) \\ y(t) \quad = H^y x(t) \\ c(t) \quad = H^c x(t) + Lg(t) \end{cases} \tag{5.3}$$

where

$$\Phi = \begin{pmatrix} \Phi_{11} & \ldots & \Phi_{1N} \\ \vdots & \ddots & \vdots \\ \Phi_{N1} & \ldots & \Phi_{NN} \end{pmatrix}, G = \begin{pmatrix} G_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & G_N \end{pmatrix}$$

$$H^y = \begin{pmatrix} H_1^y & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & H_N^y \end{pmatrix}, H^c = \begin{pmatrix} H_1^c \\ \ldots \\ H_N^c \end{pmatrix}, L = \begin{pmatrix} L_1 \\ \ldots \\ L_N \end{pmatrix}.$$

Notice that the system (5.3) has the same form of system (2.1) without any disturbance action considered. Actually, the strategies presented hereafter could be applied also in cases of disturbance occurences but for the sake of clarity this scenario is skipped. Observe however that Chapter 3 and Section 2.3.2 contain all the required arguments for any possible extension in this direction. As in the previous chapters, it is assumed that the system (5.3) enjoys the Assumptions **A1** and **A2** properties. Roughly speaking, the CG design problem we want to solve is that of locally determine, at each time step t and for each agent $i \in \mathcal{A}$, a suitable reference signal $g_i(t)$ which is the best approximation of $r_i(t)$ such that its application never produces constraints violation, i.e. $c_i(t) \in \mathcal{C}_i, \forall t \in \mathbb{Z}_+, i \in \mathcal{A}$.

The ideas exploited here are based on the centralized solutions to the FF-CG design problem (Chapter 3) that have been achieved by finding, at τ steps, a CG action $g(t)$ as a function of the current reference $r(t)$ and past applied command $g(t-\tau)$

$$g(t) = \underline{g}(r(t), g(t-\tau)) \tag{5.4}$$

such that $g(t)$ is the best approximation of $r(t)$ under the condition $c(t) \in \mathcal{C}$, where $\mathcal{C} \subseteq \{\mathcal{C}_1 \times ... \times \mathcal{C}_N\}$ is the global admissible region. The modified command $g(t)$ is kept constant between two subsequent updatings if the explicit knowledge of the state plant is not available.

5.2 Sequential Feed-Forward Command Governor Scheme (S-FFCG)

Here we introduce a distributed CG scheme based on the FF-CG approach presented in Chapter 3, inspired by the serial method presented in [41]. For the sake of comprehension, in what follows, a simplified variant of the FF-CG method will be considered by setting $\rho(t) = \rho_{g(t)}, \forall t > 0$ in (3.11). In this case, the step 3.1 of the FF-CG Algorithm is ruled out and the set $\Delta\mathcal{G}(g, \rho)$ in (3.20) depends only on the current command g

$$\Delta\mathcal{G}(g) := \{\Delta g : \|H^c \Phi^k (I - \Phi)^{-1} G \Delta g\| \leq \rho_{g+\Delta g} - \gamma \rho_g, \forall k \geq 0\}. \tag{5.5}$$

where ρ_g, in this case is determined by means of (3.9) with $\mathcal{C}$ in place of $\mathcal{C}_\infty$. The above choice, although conservative, leads to a simpler analysis and

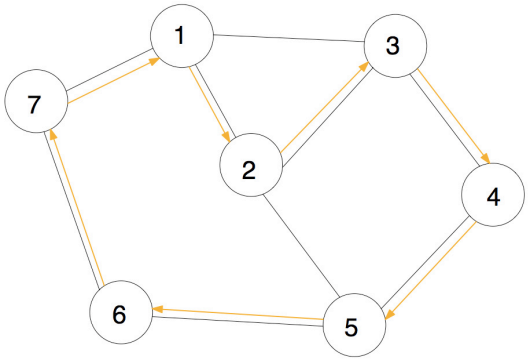

Fig. 5.1. Graph Γ and Hamiltonian cycle $\mathcal{H}$

has the merit that the agents (Master nodes in Fig. 1.6) do not need to communicate their $\rho(t)$ instances amongst them.

It is also worth pointing out that early FF-CG distributed schemes proposed in [55] were based on the computation of the following more conservative set $\Delta\mathcal{G}'(g)$

$$\Delta\mathcal{G}'(g) := \{\Delta g : \|H^c \Phi^k (I - \Phi)^{-1} G \Delta g\| \leq \delta - \gamma\delta, \forall k \geq 0\} \tag{5.6}$$

that is considerably smaller than $\Delta\mathcal{G}(g)$ and $\Delta\mathcal{G}(g, \rho)$.

In order to achieve a distributed supervising strategy based on the FFCG we assume that the agents are connected via a communication network. Such a network may be modeled by a *communication graph* defined as

Definition 5.1. (*Communication Graph*) *Let a set of N dynamically interconnected subsystems be given. Then, a Communication Graph is an undirected graph $\Gamma = (\mathcal{A}, \mathcal{B})$, where*

- $\mathcal{A}$ *denotes the set of the* N *subsystems*
- $\mathcal{B} \subset \mathcal{A} \times \mathcal{A}$ *the set of edges representing the existing communication links amongst agents in the Communication Graph, i.e. the edge* (i, j) *belongs to* $\mathcal{B}$ *if and only if the agents governing the* i*-th and the* j*-th subsystems are able to directly share information within one sampling time.*
- G *is assumed to be connected, i.e. for each couple of agents* $i \in \mathcal{A}, j \in \mathcal{A}$ *there exists at least one sequence of edges connecting* i *and* j*, with the minimum number of edges connecting the two agents denoted by* $d_{i,j}$*.*
- *The set of all agents with a direct connection with the* i*-th agent represents the* **Neighborhood of the** i**-th agent** $\mathcal{N}_i = \{j \in \mathcal{A} : d_{i,j} = 1\}$.

Let Γ be a Hamiltonian graph and assume, without loss of generality, that the sequence $\mathcal{H} = \{1, 2, ..., N-1, N\}$ is a Hamiltonian cycle. The idea behind the approach is that only one agent at a time instant is allowed to manipulate its local command signal $g_i(t)$ while all others are instructed to hold and keep applying their previous values. After each decision step, the agent in charge will update the global command received from the previous updating agent and will forward this new value to the next updating agent in the cycle. Such a polling policy implies that, eventually after a preliminary initialization cycle, at each time instant the "agent in charge" always knows the whole aggregate vector $g(t-\tau)$. By exploiting this observation we can define the following distributed FF-CG algorithm:

Algorithm 5.2.1 ***Sequential-FFCG Algorithm (S-FFCG)*** *- Agent* i

AT EACH TIME t

1.1 IF$(t == \kappa\tau, \kappa = 0, 1, \ldots) \&\& (\kappa \ mod \ N) == i$

1.1.1 RECEIVE $g(t-\tau)$ FROM THE PREVIOUS AGENT IN THE CYCLE $\mathcal{H}$

1.1.2 SOLVE

$$\begin{aligned} & g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel^2_{\Psi_i} \\ & subject\ to: \\ & \begin{cases} g(t) = [g_1^T(t-\tau), ..., g_i^T, ..., g_N^T(t-\tau)]^T \in \mathcal{W}_\delta \\ (g_i - g_i(t-\tau)) \in \Delta\mathcal{G}_i^0(g(t-\tau)) \end{cases} \end{aligned} \tag{5.7}$$

1.1.3 APPLY $g_i(t)$

1.1.4 UPDATE $g(t) = [g_1^T(t-\tau), ..., g_i^T(t), ..., g_N^T(t-\tau)]^T$

1.2 ELSE

1.2.1 APPLY $g_i(t) = g_i(t-1)$

1.3 TRANSMIT $g(t)$ TO THE NEXT AGENT IN $\mathcal{H}$

where $\Psi_i > 0$ is a weighting matrix, $\kappa \ mod \ N$ is the remainder of the integer division κ/N and

$$\Delta\mathcal{G}_i^0(g) := \left\{ \Delta g_i : [0_{m_1}^T, 0_{m_2}^T, \ldots, \Delta g_i^T, \ldots, 0_{m_N}^T]^T \in \Delta\mathcal{G}(g) \right\} \tag{5.8}$$

is the set of all possible command variations for g_i in the case that the commands of all other agents are frozen.

Notice that agents do not share the references $r_i(t)$and that, as a consequence, they are not aware on the actual costs minimized by other agents.

This design choice, although allowing one to split the original overall problem in simple local optimization problems, prevents the agents from reaching the optimal solution that the centralized FFCG would be able to compute under the same conditions. For this reason, in the following discussion we have to introduce a different notion of *optimality* along with further important notions and assumptions in order to understand the actual performance and properties of the above distributed algorithm.

Definition 5.2. ***(Admissible direction)*** *- Let $\mathcal{S} \subset \mathbb{R}^m$ be a convex set and let us consider an arbitrary point $g \in \mathcal{S}$. The vector $v \in \mathbb{R}^m$ represents an admissible direction for $g \in \mathcal{S}$ if there exists a fixed strictly positive scalar $\bar{\lambda} > 0$ such that $(g + \lambda v) \in \mathcal{S}, \lambda \in [0, \bar{\lambda}]$.*

Definition 5.3. ***(Decision Set of agent i)*** *- The Decision Set $\mathcal{V}_i^{\mathcal{S}}(g)$ of the agent i at a point $g \in \mathcal{S}$ represents the set of all admissible directions belonging to $\mathbb{R}_i^m$ that such an agent could move along in updating its action when all other agents held their commands unvaried, viz. $\mathcal{V}_i^{\mathcal{S}}(g) := \{d \in \mathbb{R}^{m_i} : [0_1^T, \ldots, 0_{i-1}^T, d^T, 0_{i+1}^T, \ldots, 0_N^T]^T$ is an admissible direction for $g \in \mathcal{S}\}$.*

Definition 5.4. ***(Viability property)*** *- A point $g \in \mathcal{S}$ is said to be "*viable*" if, for any admissible direction $v = [v_1^T, ..., v_N^T]^T \in \mathbb{R}^m$, $v_i \in \mathbb{R}^{m_i}$ with $\sum_{i=1}^N m_i = m$, at least one subvector $v_i \neq 0$ there exists such that $v_i \in \mathcal{V}_i^{\mathcal{S}}(g)$.*

Definition 5.5. ***(Pareto Optimal Solution)*** *- Let vectors r_i, $i = 1, 2, ..., N$ be given. Consider the following multi-objective problem:*

$$\begin{array}{l} \min_g [\| g_1 - r_1 \|_{\Psi_1}^2, \ldots, \| g_i - r_i \|_{\Psi_i}^2, \ldots, \| g_N - r_N \|_{\Psi_N}^2] \\ subject \quad to \; g = [g_1^T, .., g_i^T, ..., g_N^T]^T \in \mathcal{W}_\delta \end{array} \tag{5.9}$$

A solution $g^* \in \mathcal{W}_\delta$ is a Pareto Optimal solution of the optimization problem (5.9) if there not exist $g \in \mathcal{W}_\delta$, such that: $\| g_i - r_i \|_{\Psi_i}^2 \leq \| g_i^* - r_i \|_{\Psi_i}^2 \; \forall i \in \{1, \ldots, N\}$ and $\| g_j - r_j \|_{\Psi_j}^2 < \| g_i^* - r_i \|_{\Psi_i}^2, j \in \mathcal{A}$.

Definition 5.6. ***(Nash Equilibrium)*** *- Consider the multi-objective problem (5.9) for a certain reference $r = [r_1^T, ..., r_N^T]^T$. Then, a solution $g^n \in \mathcal{W}_\delta$ is a Nash equilibrium of the problem (5.9) if for each $i \in \mathcal{A}$ there not exist a subvector g_i such that $g_i - g_i^n \in \mathcal{V}_i^{\mathcal{W}_\delta}(g^n)$ that satisfies the following inequality*

$$\| g_i - r_i \|_{\Psi_i}^2 \leq \| g_i^* - r_i \|_{\Psi_i}^2 \; .$$

The above definitions are instrumental to characterize deadlock situations that, unlike the centralized solution, may exist in this distributed scheme when the same constraint set $\mathcal{W}_\delta$ of the centralized scheme is used. The rationale is that by acting one agent at a time, certain viable paths existing in the centralized scheme are precluded and the agents could get stuck on Nash Equilibria indefinitely without getting a Pareto Optimum. In order to clarify

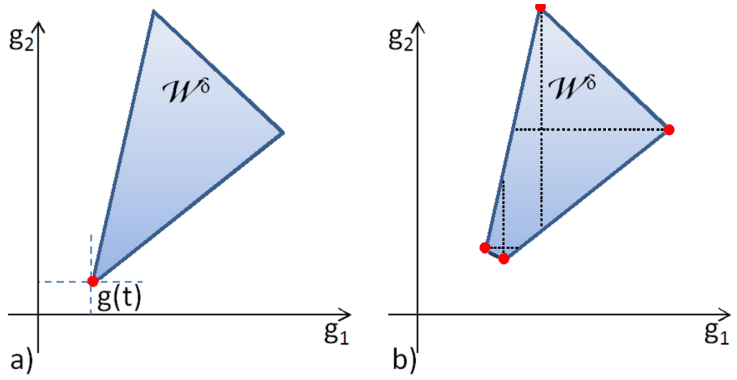

Fig. 5.2. Two-agent cases. Each agent selects its command along a different axis. a) No viable point situation: in this case the point $g(t)$ is not viable. In fact both agents cannot change their local command without violating boundaries; b)Viable points: in this case, for each red vertex at least one out of the two agents can move inside $\mathcal{W}_\delta$. In fact, each admissible direction $v = [v_1, v_2]^T$ at each vertex is such that either $\lambda_1[v_1, 0]^T, \lambda_1 > 0$ or $\lambda_2[0, v_2]^T, \lambda_2 > 0$ is admissible.

such matters, next Figg. 5.2-5.3 depict different viable and no-viable situations for points on the border of $\mathcal{W}_\delta$.

In order to avoid this deadlock situations and allow agents to get always a Pareto Optimum rather than a Nash Equilibrium, we have to introduce the following assumption for the points belonging to the border of $\mathcal{W}_\delta$

A3. Each point belonging to $\partial(\mathcal{W}_\delta)$ is viable, $\partial(\mathcal{W}_\delta)$ denoting the border of $\mathcal{W}_\delta$.

In the next section, a computable way to check if $\mathcal{W}_\delta$ satisfies **A3**, according to Definition (5.4) and a geometrical method allowing one to compute suitable inner approximations of $\mathcal{W}_\delta$ satisfying **A3** are presented.

Finally, the following properties can be shown to hold under **A3** for the above stated S-FFCG scheme

Theorem 5.7. *Let assumptions* **A1-A2-A3** *be fulfilled for the system composed by N subsystems in form (5.1). Let us consider the distributed* ***S-FFCG*** *selection rule (5.7) and let an admissible aggregate command signal $g(0) = [g_1^T(0), \ldots, g_N^T(0)]^T \in \mathcal{W}_\delta$ be applied at $t = 0$ such that (3.8) holds true. Then*

1) for each agent $i \in \mathcal{A}$, at each time $t = k\tau, k \in \mathbb{Z}_+$, the minimizer in (5.7) uniquely exists and can be obtained by locally solving a convex constrained optimization problem;
2) the overall system acted by the agents implementing the S-FFCG policy never violates the constraints, i.e. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$;

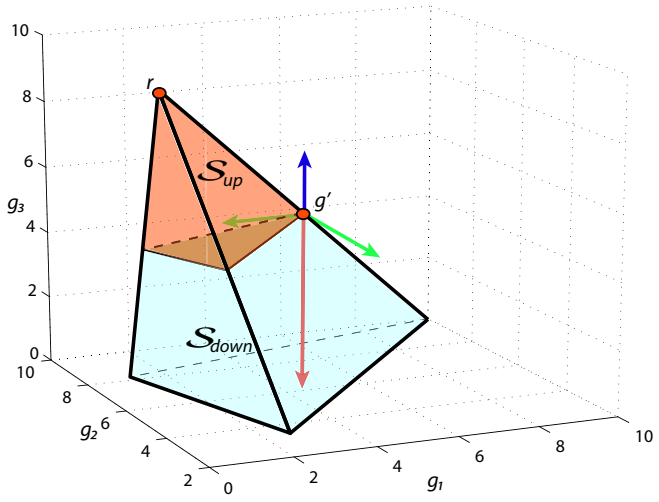

Fig. 5.3. A three-agent non viable case with agents $\mathcal{A} = \{1, 2, 3\}$ aimed at minimizing respectively $\|g_1 - r_1\|$, $\|g_2 - r_2\|$ and $\|g_3 - r_3\|$. At the point g', although agents 1 and 2 are locked (green arrows represent not admissible directions). However, no deadlock occurs because agent 3 could move along the red arrow in the $\mathcal{S}_{down}$ region. In the present situation g' represents a Nash equilibrium for problem (5.9) because the viable direction (red arrow) of agent 3 corresponds to increase its cost. In fact, its distance from the reference r_3 in $\mathcal{S}_{up}$ increases. Nevertheless g' is not a Pareto-Optimal solution because all points g^* lying over the segment connecting g' and r are such that $\|g_i^* - r_i\| < \|g_i' - r_i\|, i = 1, 2, 3..$

3) whenever $r(t) \equiv [\ r_1^T, \ldots, r_N^T\ ]^T, \forall t$, *with* r_i *a constant set-point, the sequence of solutions* $g(t) = [g_1^T(t), \ldots, g_N^T(t)]^T$ *asymptotically converges to a Pareto-Optimal stationary (constant) solution of (5.9), which is given by* r *whenever* $r \in \mathcal{W}_\delta$, *or by any other Pareto-Optimal solution* $\hat{r} \in \mathcal{W}_\delta$ *otherwise.*

Proof

1) The existence of an admissible solution for each agent at each time $k\tau$ can be proved by simply remarking that $g_i(t) = g_i(t - \tau)$, is always an admissible, although not necessarily the optimal, solution for the prescribed problem at time t.
2) At each time $t = k\tau$, with $k \in \mathbb{Z}_+$, from a centralized point of view, a command $g(k\tau)$ complying with (3.19) is applied to the overall plant. By construction, the latter implies that the set-valued virtual predictions along the virtual time i defined in (2.8) satisfy

$$\bar{c}(i, x(k\tau), g(k\tau)) \in \mathcal{C},\ \forall i \in \mathbb{Z}_+,$$

Then, the statement is proved by simply noticing that the following inclusion

$$c(t) = c(i, x(k\tau), g(k\tau)) \in \mathcal{C},$$

holds true for all time instants $t = k\tau + i,\ \ i \in \{0, 1, ..., \tau - 1\}$ and by repeating the same argument for all $k \in \mathbb{Z}_+$.

3) The stated convergence property follows simply because the sequences of solutions $g_i(t)$ makes the sequences of local costs $||g_i(t) - r_i||^2_{\Psi_i}$ non increasing for any $i = 1, ..., N$ under constant setpoints. In fact, it is not convenient for the agents to modify their actual optimal solutions if the costs cannot be decreased further on. To this end, let $g_i(t)$ be the S-FFCG action of the i-th agent at time t, solution of the optimization problem (5.7). As already discussed, $g_i(t)$ is still an admissible, though not necessarily the optimal, solution at time $t + \tau$. Hence, the sequences of costs $||g_i(t) - r_i||^2_{\Psi_i}$ are all non increasing, i.e.

$$\| g_i(t + \tau) - r_i \|^2_{\Psi_i} \leq \| g_i(t) - r_i \|^2_{\Psi_i} \tag{5.10}$$

Then, we want to show that any stationary optimal solution, viz. $g(t) = g(t + 1)\ \forall t$, is Pareto Optimal by proving that a solution is not stationary if is not Pareto-Optimal. To this end, let $g'(t) = [g_1'^T(t), ..., g_N'^T(t)]^T$ be the actual solution at time $t = k\tau, k \in \mathbb{Z}_+$ which is assumed to be not Pareto-Optimal. As a consequence, other different solutions exist which improve the costs. Supposedly, vectors $v = [v_1^T, ..., v_N^T]^T \in \mathbb{R}^m$ would exist with $g'(t) + v \in \mathcal{W}_\delta$, such that

$$||g_i'(t) + v_i - r_i||^2_{\Psi_i} - ||g_i'(t) - r_i||^2_{\Psi_i} \leq 0, \tag{5.11}$$

happens to hold for all $i \in \mathcal{A}' := \{i \in \mathcal{A} : v_i \neq 0\}$ with some of the above inequalities becoming strict for at least one index $i \in \mathcal{A}'$. Because of the strict convexity of the norm $||\cdot||^2_{\Psi_i}$, the following inequality happens to be true for all $\alpha \in (0, 1)$

$$\begin{aligned} ||(1 - \alpha)g_i' + \alpha(g_i'(t) + v_i) - r_i||^2_{\Psi_i} \\ < (1 - \alpha)||g_i'(t) - r_i||^2_{\Psi_i} + \alpha||g_i'(t) + v_i - r_i||^2_{\Psi_i}, \forall i \in \mathcal{A}' \end{aligned} \tag{5.12}$$

Therefore, by means of straightforward algebraic manipulations, one arrives to

$$\begin{aligned} ||g_i'(t) + \alpha v_i - r_i||^2_{\Psi_i} - ||g_i'(t) - r_i||^2_{\Psi_i} \\ < \alpha(||g_i'(t) + v_i - r_i||^2_{\Psi_i} - ||g_i'(t) - r_i||^2_{\Psi_i}), \forall i \in \mathcal{A}' \end{aligned} \tag{5.13}$$

for all $\alpha \in (0, 1)$. Because (5.11), the right-hand term in (5.13) is always negative. Then, one can state

$$||g_i'(t) + \alpha v_i - r_i||^2_{\Psi_i} - ||g_i'(t) - r_i||^2_{\Psi_i} < 0, \forall \alpha \in (0, 1), \forall i \in \mathcal{A}' \tag{5.14}$$

The latter may be interpreted as the fact that if the above admissible direction v did exist at $g'(t)$, **for each** agent $i \in \mathcal{A}'$ it would be strictly

convenient to move to $g_i'(t)+\alpha v_i$, for a suitable value of α, from its previous solution $g_i'(t)$.
Now we have to verify that at least one agent is allowed to move from $g_i'(t)$ along v_i because of constraints. To this end, because of **A3**, note that v_i belongs to $\mathcal{V}_i^{\mathcal{W}_\delta}(g'(t))$ for all agents corresponding to any not empty subset $\mathcal{A}_v'$. Hence, according to the sequential S-FFCG updating policy, if at time $t = k\tau$, the index $(k \bmod N) \in \mathcal{A}_v'$ then, because of (5.14), the agent $i' = k \bmod N$ will find convenient to move into $g_{i'}'(t) + \alpha v_{i'}, \alpha \in [0, \bar{\alpha}]$. In fact, because of viability of $g'(t)$ (see the above definition) $v_{i'} \in \mathcal{V}_{i'}^{\mathcal{W}_\delta}(g'(t))$ implies that a scalar $\bar{\alpha} \in (0, 1)$ exists ensuring $g' + [0_1^T, ..., \alpha v_{i'}^T, ..., 0_N^T]^T \in \mathcal{W}_\delta$ for all $\alpha \in (0, \bar{\alpha})$.
When no agents in $\mathcal{A}_v'$ are allowed to update their actions at $t = k\tau$, all of them constantly apply the most recently applied commands until one of them becomes the allowed agent. In fact, the condition $\mathcal{A}_v' \subseteq \mathcal{A}$ ensures that a future time $t' = (k + j')\tau$, $j' \in [1, N]$, surely will exist for the agent $((k+j') \bmod N) \in \mathcal{A}_v'$. Please notice also that $\mathcal{A}_v'$ does not change because $g((k+j)\tau) = g'(t)$, for all $j \in [1, j']$. Thus, if we are not at a Pareto-optimal solution at time t, at least one of the agents will move from it by $N\tau$ time steps.

Remark 5.8. The disturbance acting case may be taken into account by simply determining, in the above design procedure, $\mathcal{W}_\delta$, ρ_g and in turn $\Delta\mathcal{G}(g)$ w.r.t. to $\mathcal{C}_\infty$ (see (2.5)) instead of $\mathcal{C}$.

5.2.1 Computational Details

The goal of this subsection is to give a complete formulation of the Problem (5.7) when $\mathcal{C}$ consists of polyhedral constraints (see (2.18)-(2.21)). To this end it is sufficient to specialize the FF-CG centralized problem (3.45) to the case where the decision variable is a subvector g_i of the entire command g, related to the i-th agent. Then Problem (5.7) is rephrased as

$$g_i(t) = \arg\min_{g_i}(g_i - r_i(t))^T \Psi_i (g_i - r_i(t))$$

$$s.t. \begin{cases} T\big(H_c(I-\Phi)^{-1}G + L\big)\,[g_1^T(t-\tau), ..., g_i^T, ..., g_N^T(t-\tau)]^T \leq q - \delta[\sqrt{T_j^T T_j}] \\ \| T_j \| \| H_c\Phi^k(I-\Phi)^{-1}G(g_i - g_i(t-\tau)) \| \\ \qquad + T_j^T(H_c(I-\Phi)^{-1}G + L)[g_1^T(t-\tau), ..., g_i^T, ..., g_N^T(t-\tau)]^T \\ \qquad \leq q_j - \gamma\rho(t-\tau)\sqrt{T_j^T T_j}, \quad j = 1, ..., z, \ k = 0, ..., k_0 \end{cases} \tag{5.15}$$

where $[\sqrt{T_j^T T_j}]$ is defined in (2.24), k_0 is computed by means of Algorithm 2.3.1 and τ, for the considered γ, is evaluated according to Algorithm 3.2.1.

5.3 Constraints Qualification

In this section, some aspects of the viability property **A3** related to Definition (5.4) will be clarified and, in particular, a numerical procedure to check that this property is satisfied for all points of the boundaries of a given polyhedric set of constraints is presented.

First of all, it is worth pointing that, as far as the ability of agents to move towards a Pareto-Optimal solution is concerned, Assumption **A3** represents only a sufficient condition only. In fact, it is possible to observe in Fig. 5.3 that if the reference r would coincide with g', although the point g' is not viable according to Definition (5.4), the agents would be in a Pareto-Optimal solution. Moreover, in Fig. 5.4 a trickier situation is depicted, here the agents are at a point g' which does not comply with Assumption **A3**. Despite this fact, since such an assumption is only sufficient, the depicted agents are capable to reach a Pareto-Optimum without any difficulty. The observed behavior can be explained by considering that that Definition (5.4) and consequently Assumption **A3** do not directly account for the reference r and, in turn, the functional costs.

It is conjectured that the following more general definition of viability, that takes into account both the reference r and the notion of Cost-Descendant directions, could be more appropriated giving rise to necessary and sufficient conditions. Roughly speaking, this more general definition of viability states that a point is viable if, for all possible references, at least one agent can move decreasing its cost. Formally,

Definition 5.9. (*Cost-Descendent Direction*) *Let $\mathcal{S} \subset \mathbb{R}^m$ be a convex set and consider a point $g \in \mathcal{S}$. The admissible direction $v \in \mathbb{R}^m$ at g represents a Cost-Descendant (CD) direction for a given reference $r = [r_1^T, ..., r_N^T]^T$ with $\sum_{i=1}^{N} m_i = m$, if $\| g_i - r_i \|^2_{\Psi_i} \geq \| g_i + v_i\lambda - r_i \|^2_{\Psi_i}$ $\forall i \in \{1, \ldots, N\}, 0 < \lambda \leq \bar{\lambda}$ and $\| g_j - r_j \|^2_{\Psi_j} > \| g_j + \lambda v_j - r_j \|^2_{\Psi_i}, j \in \mathcal{A}, 0 < \lambda \leq \bar{\lambda}$,*

Definition 5.10. (*Reference Dependent Viability*) *A point $g \in \mathcal{S}$ is said to be "*viable*" if, for all desired references $r = [r_1^T, ..., r_N^T]^T \in \mathbb{R}^m$, $r_i \in \mathbb{R}^{m_i}$ with $\sum_{i=1}^{N} m_i = m$ and $r \neq g$, at least one subvector $v_i \neq 0$, $i \in \mathcal{A}$ exists such that $v_i \in \mathcal{V}_i^{\mathcal{S}}(g)$ and the direction $[0_1^T, ..., v_i, ..., 0_N^T]$ is a Cost-Descendant direction at g w.r.t r.*

One of the merits of the viability notion provided by Definition (5.4) is that simple numerical procedures can be devised to test Assumption **A3**. On the contrary, the notion of viability given in Definition (5.10) is expected to be more cumbersome to be verified and this aspect requires further investigations.

The numerical test, here presented, checks a finite number of points belonging to $\partial(\mathcal{W}_\delta)$ in order to verify Assumption **A3**. Such a test is necessary and sufficient to state if a point of $\partial(\mathcal{W}_\delta)$ is viable according to Definition

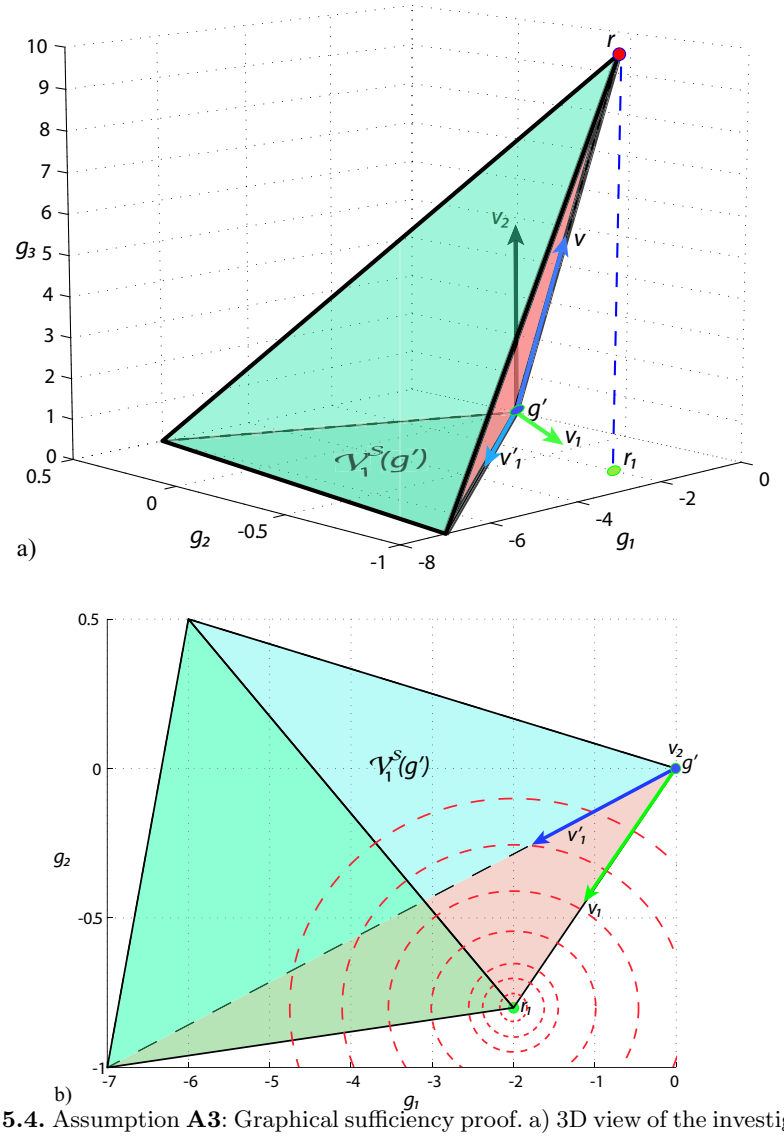

Fig. 5.4. Assumption **A3**: Graphical sufficiency proof. a) 3D view of the investigated polyhedron $\mathcal{S}$. b) $\mathcal{S}$ projection on the $g_1 - g_2$ plane. In this case, there are two agents $\mathcal{A} = \{1, 2\}$ minimizing respectively $\|g_1 - r_1\| + \|g_2 - r_2\|$ and $\|g_3 - r_3\|$. The point g' is not viable according to Definition (5.4) because the sub-vectors v_1 and v_2 of the admissible direction v do not belong to $\mathcal{V}_1^{\mathcal{S}}(g')$ and, respectively, $\mathcal{V}_2^{\mathcal{S}}(g')$. Then, Assumption **A3** is not satisfied. Nevertheless, the polyhedron $\mathcal{S}$ is viable in g' and no deadlocks occur. In fact, although agent 2 is locked, agent 1 can move along v_1' for improving its cost.

(5.4) if no deadlocks can take place also under the stated distributed S-FFCG policy.

Lemma 5.11. *Let the polyhedron $\mathcal{S} \subset \mathbb{R}^m$ be given and expressed as $Ag \leq b, A \in \mathbb{R}^{|\mathcal{J}|\times m}, b \in \mathbb{R}^{|\mathcal{J}|}$. Consider also a generic point $g' \in \mathcal{S}$ and the set $\mathcal{J}' := \{j \in \mathcal{J} : a_j^T g' = b_j\}$. Then, g' is viable iff the following test fails.*

Test - *Find, if there exists, a vector $w = [w_1^T, ..., w_N^T]^T \in \mathbb{R}^m$ such that*

$$\begin{cases} A(g'+w) \leq b \\ a_j^T[0_1^T, ..., w_i^T, ..., 0_N^T]^T > 0, \quad \textit{for all } i \in \mathcal{A}, \textit{ for at least one } j \in \mathcal{J}' \end{cases} \tag{5.16}$$

where a_j^T and b_j denote the rows of the matrix A and, respectively, vector b.

Proof

1) *Sufficiency* - ***Test*** failure $\Rightarrow$ viability of g' :
Assume that (5.16) has no solution. Then for all $w \in \mathbb{R}^m$ such that $A(g'+w) \leq b$ there exists at least one index $i \in \mathcal{A}$ satisfying

$$a_j^T[0_1^T, ..., w_i^T, ..., 0_N^T]^T \leq 0, \forall j \in \mathcal{J}' \tag{5.17}$$

The latter condition directly implies that for all $j \in \mathcal{J}'$

$$a_j^T(g' + [0_1^T, ..., w_i^T, ..., 0_N^T]^T) - b_j \leq 0 \tag{5.18}$$

Otherwise, for all $j \in \mathcal{J}$ such that $a_j^T g' < b_j$, two exclusive situations have to be considered
1) if $a_j^T[0_1^T, ..., w_i^T, ..., 0_N^T] \leq 0$, it is trivial to see that (5.18) still holds true;
2) if $a_j^T[0_1^T, ..., w_i^T, ..., 0_N^T] > 0$, one cannot state that (5.18) is true but, because $a_j^T g' < b_j$, there exists a scalar $\bar{\lambda} = \frac{b_j - a_j^T g'}{a_j^T[0_1^T, ..., w_i^T, ..., 0_N^T]} > 0$ such that $a_j^T(g' + \lambda[0_1^T, ..., w_i^T, ..., 0_N^T]^T) - b_j \leq 0, \forall \lambda \in [0, \bar{\lambda}]$;

In both cases, for all admissible $w \in \mathbb{R}^m$, at least one subvector $\bar{\lambda} w_i \neq 0$, with $\bar{\lambda} > 0$, is such that $A(g' + [0_1^T, ..., \bar{\lambda} w_i^T, ..., 0_N^T]) \leq b$, the latter being equivalent to state that $w_i \in \mathcal{V}_i^{\mathcal{S}}(g')$. Please notice that if g' is an internal point of $\mathcal{S}$, i.e. $Ag' < b$, for all admissible $w \in \mathbb{R}^m$, the set $\mathcal{J}'$ is empty, hence the ***Test*** fails and this is sufficient to state that all inner points in $\mathcal{S}$ are viable.

2) *Necessity* - ***Test*** failure $\Leftarrow$ viability of g':
We prove the Necessity of the Lemma by proving that if ***Test*** has success then g' is not viable. The success of ***Test*** implies that it exists a vector w' that satisfies

$$\begin{cases} A(g'+w') \leq b \\ a_j^T[0_1^T, ..., (w_i')^T, ..., 0_N^T]^T > 0 \\ \qquad \textit{for at least one } j \in \mathcal{J}', \forall i \in \mathcal{A}. \end{cases} \tag{5.19}$$

The existence of an index $j \in \mathcal{J}'$ such that $a_j^T[0_1^T, ..., (w_i')^T, ..., 0_N^T]^T > 0, \forall i \in \mathcal{A}$, implies that

$$a_j^T(g' + [0_1^T, ..., (w_i')^T, ..., 0_N^T]^T) - b_j > 0$$

Then, for all w_i' the vector $(g' + [0_1^T, ..., (w_i')^T, ..., 0_N^T]^T)$ does not satisfy at least one of the above inequalities. Hence, at g' there exists an admissible direction $w' \in \mathbb{R}^m$ for which all sub-vectors w_i' do not belong to $\mathcal{V}_i^{\mathcal{S}}(g')$, meaning that g' is not viable.

Because the direct application of the above conditions to check the viability of all points of $\partial(\mathcal{S})$ could give rise to a cumbersome numerical procedure, several geometrical results are here presented to prove that the viability of all points on the boundaries of $\mathcal{S}$, hereafter denoted as $\partial(\mathcal{S})$, can be established by only checking a finite number of points on $\partial(\mathcal{S})$. To this end, the usual notion of *face* of a polyhedron is recalled.

Definition 5.12. (*Face of a polyhedron*) - *Let a convex polyhedron $\mathcal{S} \subset \mathbb{R}^m$ expressed as a set of p linear inequalities $a_j^T g - b_j \leq 0, j \in \mathcal{J} := \{1, \ldots, p\}$ be given. Any region $\mathcal{P} := \{g \in \mathcal{S} : a_j^T g - b_j = 0, j \in \mathcal{J}' \subset \mathcal{J}\}$ is said to be a* face *of $\mathcal{S}$. Moreover, the quantity $m - |\mathcal{J}'|$ represents the* order *of $\mathcal{P}$.*

Based on the above definition, vertices are 0-order faces, facets are $(m-1)$-order faces, ridges are $(m-2)$-order faces and so on.

Lemma 5.13. *Let a convex polyhedron $\mathcal{S}$ and a $(m-k)$-order face $\mathcal{P}$ with $k \in \{1, ..., m-1\}$ be given. Then, $v \in \mathbb{R}^m$ is an admissible direction for all points of the interior of $\mathcal{P}$, say it $In(\mathcal{P})$, if it is an admissible direction for at a least one point g of $In(\mathcal{P})$.*

Proof - Consider a generic point $g \in In(\mathcal{P})$. Because it belongs to the interior of $\mathcal{P}$, only a subset of inequalities related to $\mathcal{S}$ are strictly satisfied, viz.

$$\left(a_j^T g - b_j = 0, \forall j \in \mathcal{J}' \subset \mathcal{J}\right) \wedge \left(a_j^T g - b_j < 0, \forall j \in \mathcal{J} \backslash \mathcal{J}'\right). \tag{5.20}$$

Next, assume $v \in \mathbb{R}^m$ be an admissible direction at $g \in In(\mathcal{P})$. It's also useful to remind that an admissible direction $v = [v_1^T, \ldots, v_N^T]^T$ at g implies that there exists a $\bar{\lambda} > 0$ which satisfies

$$a_j^T(g + \lambda v) - b_j \leq 0, \forall \lambda \in [0, \bar{\lambda}], \forall j \in \mathcal{J}. \tag{5.21}$$

Then, because (5.20) and (5.21), one has that

$$a_j^T v \leq 0, \forall j \in \mathcal{J}' \subset \mathcal{J}. \tag{5.22}$$

Now, consider a different internal point $g' \neq g$. Note that also for g' condition (5.20) holds true. As a consequence, v is still admissible because:

1) - for all $j \in \mathcal{J}'$, conditions (5.20) and (5.22) imply that $a_j^T g' + \lambda v \leq b_j, \lambda > 0$;
2) - for all $j \in \mathcal{J} \backslash \mathcal{J}'$, two exclusive cases can happen: if $a_j^T v \leq 0$, it is trivial to see that $a_j^T g' + \lambda v \leq b_j$ for all $\lambda > 0$; in the other case, when $a_j^T v > 0$, because $a_j^T g' < b_j$ there exists a scalar $\bar{\lambda} = \frac{b_j - a_j^T g'}{a_j^T v} > 0$ such that $a_j^T(g' + \lambda v) \leq b_j, \forall \lambda \in [0, \bar{\lambda}]$.

Based on the above Lemma, the following result which allows one to verify in an easy way the CD of $\mathcal{S}$ can be stated.

Lemma 5.14. *Let a convex polyhedron $\mathcal{S}$ be given and $\mathcal{P}_{\mathcal{S}}$ denote the set of all faces of $\mathcal{S}$. Then, each point of $\mathcal{S}$ is viable, that is the constraint set $\mathcal{S}$ is CD, if for each element $\mathcal{P} \in \mathcal{P}_{\mathcal{S}}$ there exists at least a viable point $g \in In(\mathcal{P})$.*

Proof - The viability of points belonging to $In(\mathcal{S})$ has been proved in Lemma 5.11, and we need then to prove only the viability of points on the faces $\mathcal{P}_{\mathcal{S}}$. We will resort to the mathematical induction principle. To this end, we assume the viability of points on 0-order faces (vertices) of $\mathcal{P}_{\mathcal{S}}$. It is then sufficient to show that all points on $(m-j)$-order faces are viable provided that all points on $(m-j-1)$-order faces are such. To this end, consider a $(m-j)$-order face $\mathcal{P}$ and assume that all points of $\partial(\mathcal{P})$ are viable, $\partial(\mathcal{P})$ consisting of $(m-j-1)$-order faces only. Because points on $\partial(\mathcal{P})$ have been assumed viable, we need only to investigate the viability of points belonging to $In(\mathcal{P})$ which, based on Lemma 5.13 results, are viable provided that a single viable inner point $g \in In(\mathcal{P})$ exists. Then, by taking into account the viability definition given in the previous section, one can be ensured that:

1) If an admissible $v = [v_1^T, ..., v_N^T]^T$ exists at g, then $v_i \in \mathcal{V}_i^{\mathcal{S}}(g)$ (i.e. $[0_1, ..., v_i, ..., 0_N]$ admissible at g) implies $v_i \in \mathcal{V}_i^{\mathcal{S}}(g')$ for all $g' \in In(\mathcal{P})$;
2) Since any point $g' \in In(\mathcal{P})$ shares the same admissible directions of g, the previous item holds true for all admissible $v = [v_1^T, ..., v_N^T]^T$ at g' .

The above results allow one to introduce the following numerical procedure to check the viability all points belonging to the boundaries of a polyhedron $\mathcal{S}$.

Constraints Qualification test for polyhedrons $\mathcal{S}$

1.1 COMPUTE $\mathcal{P}_{\mathcal{S}}$
1.2 SET $Pl := \emptyset$
1.3 FOR EACH $\mathcal{P} \in \mathcal{P}_{\mathcal{S}}$
  1.3.1 SELECT $g \in In(\mathcal{P})$
  1.3.3 APPEND g TO Pl
1.4 SET *check* := *viable*
1.5 FOR EACH $g \in Pl$
  1.5.1 PERFORM ***Test***
  1.5.2 IF ***Test*** FAILS
    1.5.2.1 SET *check* = *notviable*
    1.5.2.1 BREAK
1.6 RETURN *check*

Remark 5.15. A worst case application of the ***Test*** involves the checking of $|\mathcal{J}| \times N$ inequalities (5.16) for each point collected in Pl. The above algorithm may need then a huge number of iterations to finish. However, because this problem has to be solved off-line, the computational burden does not represent a significant obstacle..

5.3.1 Viable approximations

In this subsection we describe a method to find arbitrarily accurate viable multi-box inner-approximations of a no CD polyhedron in the case that all agents have mono-dimensional decision sets, viz. $m_i = 1, \forall i \in \mathcal{A}$ and $N = m$. To this end, the notion of *box* is recalled.

Definition 5.16. (*Box in* $\mathbb{R}^m$) *A box is a convex polytope with all the hyperplanes characterizing its boundaries parallel to the axes. More formally a box $\Omega(l, u)$ is defined as*

$$\Omega(l, u) := x \in \mathbb{R}^m : l \leq x \leq u, \tag{5.23}$$

where l and u are real vectors of $\mathbb{R}^m$ and the inequalities hold componentwise.

Let us consider, for a no viable polyhedron $\mathcal{S}$, a multi-box inner approximation $\mathcal{M}(\mathcal{S}) \subset \mathcal{S}$. That is, according to [54], a collection of full-dimensional boxes such that

1) the intersection between any two boxes is not full-dimensional;
2) the union of all boxes in $\mathcal{M}(\mathcal{S})$ is contained in $\mathcal{S}$;

The numerical method described in [54] can be used to find multi-box inner approximations $\mathcal{M}(\mathcal{S})$ of $\mathcal{S}$. We will show that for such a kind of approximation the convex hull of $\mathcal{M}(\mathcal{S})$, say it $\mathcal{S}' := \text{co}\{\mathcal{M}(\mathcal{S})\}$, is always CD. It is clear from the above discussion, that each vertex of $\mathcal{S}'$ is a vertex of a box contained in $\mathcal{M}(\mathcal{S})$. By exploiting this fact, the following preliminary results can be stated

Lemma 5.17. *Let $g' \in \mathbb{R}^m$ be a point of $\partial(\mathcal{S}')$ such that $g' \in \partial(\mathcal{M}(\mathcal{S}))$. Then, m scalars $v_i \in \{-1, 1\}, i = 1, ..., m$, there exist such that, for any $\bar{\lambda} > 0$, $g' + [0_1, \ldots, \lambda v_i, \ldots, 0_m]^T \in \mathcal{S}', \forall \lambda \in [0, \bar{\lambda}], \forall i \in \mathcal{A}$.*

Proof - Because g' belongs to the border of a box $\Omega \subset \mathcal{M}(\mathcal{S})$, for each $i \in \mathcal{A}$ there exists $v_i \in \{-1, 1\}$ and a $\bar{\lambda}_i > 0$ such that a generic point $g := g' + [0_1, \ldots, \lambda v_i, \ldots, 0_m]^T \in \Omega, \forall \lambda \in [0, \bar{\lambda}_i]$. Because $\Omega \subset \mathcal{S}'$, then $g \in \mathcal{S}'$ and the lemma's statement is proved for $\bar{\lambda} = \min_{i \in \mathcal{A}} \bar{\lambda}_i$.

Lemma 5.18. *Let $g' \in \mathbb{R}^m$ and $g'' \in \mathbb{R}^m$ be two points of $\mathcal{S}'$ such that $g' + [0_1, \ldots, \lambda v_i', \ldots, 0_m]^T \in \mathcal{S}'$ and $g'' + [0_1, \ldots, \lambda v_i'', \ldots, 0_m]^T \in \mathcal{S}', \lambda \in [0, \bar{\lambda}_i'], v_i \in \{-1, 1\}, \forall i \in \mathcal{A}$. Then, for each point belonging to the convex combination $g = \gamma g' + (1 - \gamma) g''$ there exists, for each $i \in \mathcal{A}$, a pair $(\hat{\lambda}_i, v_i)$, with $\hat{\lambda}_i > 0$ and $v_i \in \{-1, 1\}$, such that $g + [0, \ldots, \lambda v_i, \ldots, 0]^T \in \mathcal{S}', \forall \lambda \in [0, \hat{\lambda}_i], \forall h \in [1, \ldots, m]$.*

Proof - In order to prove the statement it is sufficient to consider the following admissible convex combination for a certain index $i \in \mathcal{A}$

$$\gamma(g' + [0_1, \ldots, \bar{\lambda}_i v_i', \ldots, 0_m]^T) + (1 - \gamma)(g'' + [0_1, \ldots, \bar{\lambda}_i v_i'', \ldots, 0_m]^T) \in \mathcal{S}' \tag{5.24}$$

which, when $v_i' = v_i''$, becomes

$$g + [0_1, \ldots, \bar{\lambda}_i v_i', \ldots, 0_m]^T \in \mathcal{S}' \tag{5.25}$$

Then, a pair $(\hat{\lambda}_i, v_i)$ satisfying the lemma's statement is given by $\hat{\lambda}_i = \bar{\lambda}_i$ and $v_i = v_i'$. Otherwise, when $v_i' = -v_i''$, (5.24) becomes

$$g + (2\gamma - 1)[0_1, \ldots, \bar{\lambda}_i v_i', \ldots, 0_m]^T \in \mathcal{S}' \tag{5.26}$$

In this case, a suitable pair is given by $(\hat{\lambda}_i, v_i)$, where $\hat{\lambda}_i = |(2\gamma - 1)\bar{\lambda}_i|$ and $v_i = \text{sign}\{(2\gamma - 1)v_i'\}$.

Lemma 5.19. *For each point g of the border of $\mathcal{S}'$ there exist m admissible directions aligned to the axes, i.e. $g + [0_1, \ldots, \lambda v_i, \ldots, 0_m]^T \in \mathcal{S}', \lambda \in [0, \bar{\lambda}], v_i \in \{-1, 1\}, \forall i \in \mathcal{A}, \forall g \in \partial(\mathcal{S}')$.*

Proof - It directly follows from Lemmas 5.18 and 5.19. In fact, because of Lemma 5.18, this lemma statement is satisfied by all vertices of $\mathcal{S}'$. Moreover, by considering that a point $g \in \partial(\mathcal{S}')$ can be expressed as a convex combination of a finite number of vertices of $\mathcal{S}'$, because Lemma 5.19, the proof is completed.

Lemma 5.20. *Let $\mathcal{S}'$ be expressed as the intersection of $|\mathcal{J}|$ inequalities $Ag \leq b$. Then, for all $g \in \partial(\mathcal{S}')$ and for all w such that $A(g + w) \leq b$ the following condition is satisfied*

$$a_j^T g + a_j^T [0_1^T, ..., w_i^T, ..., 0_N^T]^T \leq b_j, \quad \text{for all } i \in \mathcal{A}, \text{for at least one } j \in \mathcal{J}_g' \tag{5.27}$$

where $\mathcal{J}_g' := \{j \in \mathcal{J} : a_j^T g = b_j\}$.

Proof - The fact that $g + w$ is admissible allows one to state that, for each $j \in \mathcal{J}_g'$, necessarily there is an $i \in \mathcal{A}$

$$a_j^i w_i < 0. \tag{5.28}$$

Moreover, taking into account Lemma 5.19, one can assume that there exist m scalars $v_i \in \{1, 1\}$ such that

$$g + [0_1, ..., \lambda v_i, ..., 0_m]^T \in \mathcal{S}', \lambda \in [0, \bar{\lambda}], \forall g \in \partial(\mathcal{S}') \tag{5.29}$$

that implies

$$\lambda a_j^i v_i \leq 0, \lambda \in [0, \bar{\lambda}], \forall i \in \mathcal{A}, \forall j \in \mathcal{J}_g', \forall g \in \partial(\mathcal{S}') \tag{5.30}$$

As consequence, for each $g \in \partial(\mathcal{S}')$, if an $i' \in \mathcal{A}$ satisfies (5.28) for any $j' \in \mathcal{J}_g'$, necessarily w_i and v_i have the same sign, then, because of (5.30) for all $j \in \mathcal{J}_g' \backslash \{j'\}$

$$a_j^{i'} w_{i'} < 0 \tag{5.31}$$

that means

$$a_j^T (g + [0_1, ..., w_{i'}, ..., 0_m]^T) < b_j, \forall j \in \mathcal{J}_g'. \tag{5.32}$$

Finally the viability property of each point $g \in \partial(\mathcal{S}')$ is ensured by the next lemma.

Lemma 5.21. *For a given $g \in \partial(\mathcal{S}')$, let $\mathcal{V}_i^{\mathcal{S}'}(g)$ be the decision set for agent i acting at g. Then, g is viable.*

Proof - By exploiting Lemma 5.20 we have that $\forall w$ such that $A(g+w) \leq b$ there exists at least an index $i \in \mathcal{A}$, with related scalar w_i, such that

$$a_j^T(g + [0_1, ..., w_i, ..., 0_m]^T) < b_j, \forall j \in \mathcal{J}_g'. \tag{5.33}$$

As a consequence, $w_i \in \mathcal{V}_i^{\mathcal{S}'}(g)$ and w cannot satisfy system equations (5.16).

In conclusion, a no CD polyhedron $\mathcal{W}_\delta$ can be always approximated with a CD polyhedron $\mathcal{W}_\delta'$ and the S-FFCG problem recast as follows

$$\begin{array}{l} g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel_{\Psi_i}^2 \\ subject\ to: \\ \begin{cases} g(t) = [g_1^T(t-\tau), .., g_i^T, ..., g_N^T(t-\tau)]^T \in \mathcal{W}_\delta' \\ (g_i - g_i(t-\tau)) \in \Delta\mathcal{G}_i^0(g(t-\tau)) \end{cases} \end{array} \tag{5.34}$$

where the set $\mathcal{W}_\delta'$ is used in the place of $\mathcal{W}_\delta$. Each set $\Delta\mathcal{G}_i^0$ is not subject to modification because it represent a local constraint in the optimization problem (5.34), then its fulfillment does not depend on the global command vector g.

Remark 5.22. In order to improve the accuracy of the proposed method, the inner approximation could be performed more effectively by approximating only subsets of $\mathcal{W}_\delta$ that contains no viable points. This will be an issue for further researches.

Illustrative Example

In this section a short example is presented in order to show the effectiveness of proposed method. The two-dimensional polytopic constraint set $\mathcal{S}$ of Figure 5.5 is considered. It is characterized by the following five inequalities

$$Ag \leq b$$

where

$$A = \begin{pmatrix} -0.2693 & 0.9630 \\ 0.3288 & -0.9444 \\ -0.9874 & 0.1584 \\ 0.9877 & -0.1563 \\ 0.9837 & 0.1797 \end{pmatrix}, b = \begin{pmatrix} 6.1218 \\ -0.0681 \\ -1.7645 \\ 8.4683 \\ 10.7936 \end{pmatrix}$$

The CQ test presented in section V has been performed on it and the answer was achieved in 0.04 seconds by means of the Multi-Parametric Toolbox (MPT) (please see [56] for details) with MATLAB 7.12® installed on a Intel Core™2® Quad machine. The polytope $\mathcal{S}$ resulted no CD because of the presence of the vertex $g' = [1.908, 0.7355]^T$, which is not viable as illustrated in Figure 5.5. Then, $\mathcal{S}$ has been inner-approximated by 91 boxes (Figure 5.6) by using the algorithm presented in [54] which, on the same machine, took 37 seconds to terminate its execution. The resulting CD polytope $\mathcal{S}'$, that does not contain g' in its convex hull, is depicted in Figures 5.7-5.8.

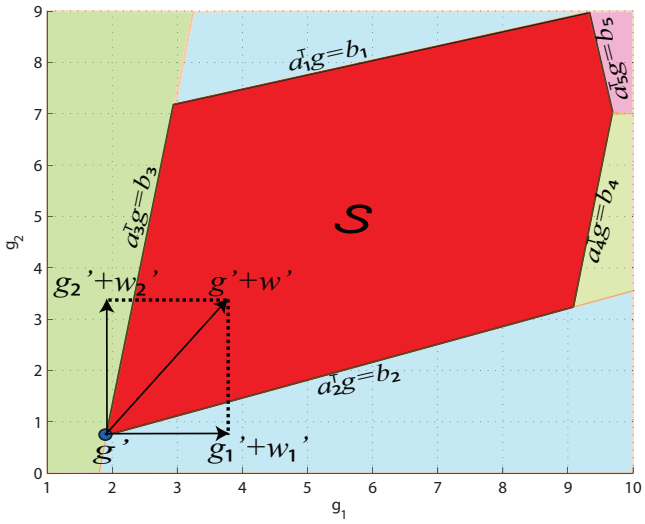

Fig. 5.5. The polytope $\mathcal{S}$. At g' $\mathcal{J}' = \{2, 3\} \subset \{1, 2, ..., 5\}$. Notice that the point g' is not viable because the admissible vector w' is such that $a_3^T[w_1', 0]^T > 0$ and $a_2^T[0, w_2']^T > 0$.

Analysis in multi-dimensional cases

We show here two interesting examples related to the use of the above presented multi-box inner approximation technique in the case of systems with multi-dimensional decision sets. The first is the above mentioned counterexample showing that the convex hull of the multi-box approximation set may fail to be viable in general multi-dimensional problems, when the viability notion of Definition (5.4) is adopted. The second is instead provided in order to remark the fact that this method may however be used as a "first attempt" to obtain an approximated viable set of a constrained region, because in many cases it is a successful approach. In this subsection we investigate if the above presented approximation could be used for problems with multi-dimensional decision set case without modification. Unfortunately, we noticed that when $m \geq 4$, problem instances may arise where the viability cannot be proved.

Example 1:

Consider a two-agent case each acting on 2-dimensional local space ($m_1 = m_2 = 2$) with $m = 4$. We are looking for two hyper-planes

$$\begin{cases} a_1^T g = b_1 \\ a_2^T g = b_2 \end{cases}$$

for which the Test of Lemma 5.11 successes. Because we assume that such hyper-planes arise from the multi-box approximating procedure, we con-

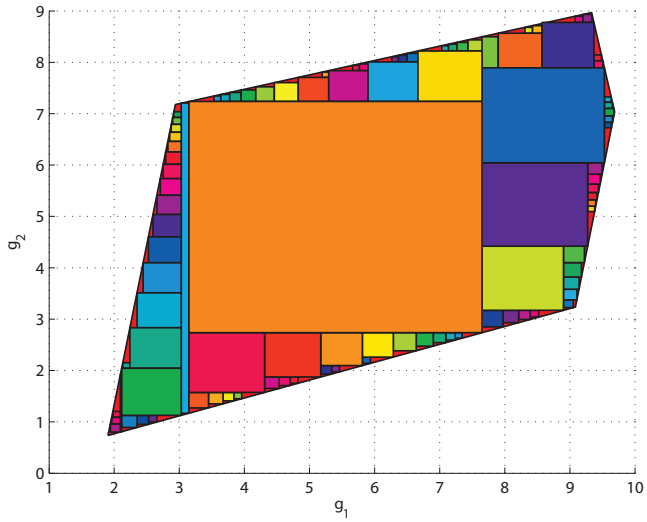

Fig. 5.6. Multi-box inner approximation $\mathcal{M}(\mathcal{S}) = \{\Omega_k\}_{k=1}^{91}$ of $\mathcal{S}$.

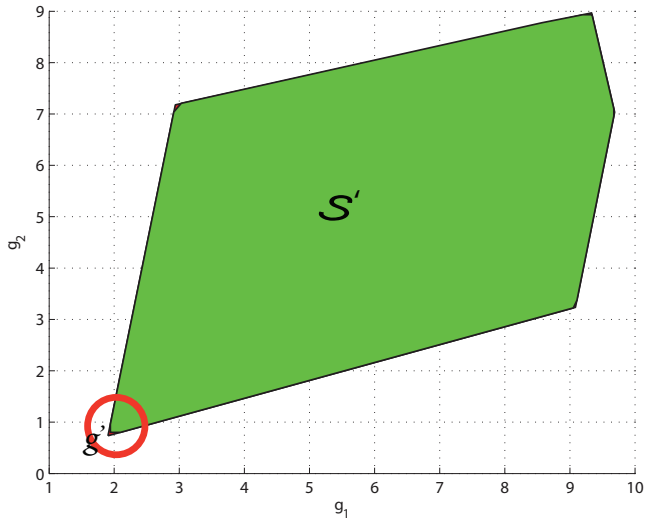

Fig. 5.7. Convex Hull of $\mathcal{M}(\mathcal{S})$.

sider that they represent faces of the convex hull of such an approximation which enjoy the properties stated in Lemmas 5.19-5.20. In order to simplify our search, without any loss of generality, we look for a direction w

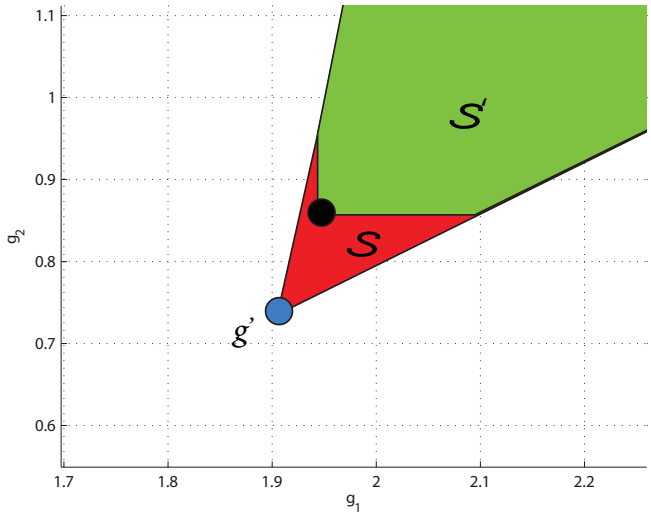

Fig. 5.8. Zoom on the no viable point g': the point g' (blue) is not inside $\mathcal{S}'$, having been substituted by a new viable vertex (black spot).

having same components in all axes, i.e. $w = [v, v, v, v]^T$ where $v \in \mathbb{R}$. The success of the Test suggests us that w and the coefficient vectors a_1 and a_2 have to jointly satisfy the following conditions

$$\begin{cases} (a_1^1 + a_1^2 + a_1^3 + a_1^4)v < 0 \\ (a_2^1 + a_2^2 + a_2^3 + a_2^4)v < 0 \\ (a_1^1 + a_1^2)v > 0 \\ (a_1^3 + a_1^4)v < 0 \\ (a_2^1 + a_2^2)v < 0 \\ (a_2^3 + a_2^4)v > 0 \end{cases} \tag{5.35}$$

In fact, according to Lemma 5.11, the latter indicates that any g lying on the considered hyperplane is not viable. Moreover, we have to add further constraints in order to impose that such hyperplane belongs to the convex hull of a multi-box inner approximation and then satisfies Lemmas 5.19-5.20. In particular, we impose that the first component of w satisfies Lemma 5.20. Then, we have to consider the following additional conditions

$$\begin{cases} a_1^1 v < 0 \\ a_2^1 v < 0 \end{cases} \tag{5.36}$$

Because the other components of w satisfy Lemma 5.19, the following constraints have to be included in our search

$$\begin{cases} (a_1^2 v < 0 \wedge a_2^2 v < 0) \vee (-a_1^2 v < 0 \wedge -a_2^2 v < 0) \\ (a_1^3 v < 0 \wedge a_2^3 v < 0) \vee (-a_1^3 v < 0 \wedge -a_2^3 v < 0) \\ (a_1^4 v < 0 \wedge a_2^4 v < 0) \vee (-a_1^4 v < 0 \wedge -a_2^4 v < 0) \end{cases} \tag{5.37}$$

Observe that several solutions for system of inequalities (5.35)-(5.37) exist and may be represented by the following compact form.

$$\begin{cases} a_1^1 < 0 \\ a_1^2 > -a_1^1 \\ \begin{pmatrix} a_1^3 < -a_1^1 - a_1^2 \wedge 0 < a_1^4 < -a_1^1 - a_1^2 - a_1^3 \wedge a_2^1 < 0 \wedge \\ 0 < a_2^2 < -a_2^3 \wedge a_2^3 < 0 \wedge -a_2^3 < a_2^4 < -a_2^1 - a_2^2 \\ \qquad\qquad - a_2^3 \wedge v > 0 \vee a_1^3 > 0 \wedge \\ a_1^4 < -a_1^1 - a_1^2 - a_1^3 \wedge a11 < 0 \wedge 0 < a_2^2 < -a_2^1 \wedge \\ (0 < a_2^3 \leq -a_2^1 - a_2^2 \wedge -a_2^3 < a_2^4 < 0 \wedge v > 0) \end{pmatrix} \end{cases} \tag{5.38}$$

where $\wedge$ and $\vee$ represents logic AND and, respectively, OR conditions. As a conclusion, from the success of the Test of Lemma 5.11 is not possible to guarantee that the multi-box approximation is viable in this simple multi-dimensional case.

Example 2:

As a second scenario, we consider the same problem of Example 1 with one of the two agents acting on a scalar decision space, i.e. $m_1 = 1$ and $m_2 = 3$. In this case we look for two hyper-planes and an admissible direction $w = [w_1, w_2],\ w_1 \in \mathbb{R}, w_2 \in \mathbb{R}^3$ that satisfy the Test of Lemma 5.11 or, equivalently, the following system of inequalities

$$\begin{cases} a_1^T w < 0 \\ a_2^T w < 0 \\ a_1^2 w_2^1 + a_1^3 w_2^2 + a_1^4 w_2^3 < 0 \\ a_2^2 w_2^1 + a_2^3 w_2^2 + a_2^4 w_2^3 > 0 \\ a_1^1 w_1 > 0 \\ a_2^1 w_1 < 0. \end{cases} \tag{5.39}$$

Furthermore, an additional constraint of type (5.36) has to be added for the satisfaction of Lemma 5.20, that is

$$\begin{cases} a_1^2 w_2^1 < 0 \\ a_2^2 w_2^1 < 0 \end{cases} \tag{5.40}$$

and further constraints of type (5.37) are needed in order to take into account Lemma 5.19

$$\begin{cases} (a_1^1 w_1 < 0 \wedge a_1^2 w_1 < 0) \vee (-a_2^1 w_1 < 0 \wedge -a_2^1 w_1 < 0) \\ (a_1^3 w_2^2 < 0 \wedge a_2^3 w_2^2 < 0) \vee (-a_1^3 w_2^2 < 0 \wedge -a_2^3 w_2^2 < 0) \\ (a_1^4 w_2^3 < 0 \wedge a_2^4 w_2^3 < 0) \vee (-a_1^4 w_2^3 < 0 \wedge -a_2^4 w_2^3 < 0) \end{cases} \tag{5.41}$$

Please note that any attempt to solve the system of inequalities fails because the first inequalities of (5.41) and the latter two inequalities of (5.39)

are incompatible. Then, in this case, two hyper-planes belonging to the convex hull of a multi-box region are always viable and the multi-box inner approximation succeeds in making the resulting approximated region viable.

5.4 Parallel Feed-Foward Command Governor Scheme (P-FFCG)

The main drawback of the S-FFCG algorithm is that, every τ time instants, only one agent at a time is allowed to modify its local command. This yields to a reduced capability to track the desired reference $r(t)$. In order to overcome such a limitation, any agent should be enabled to select its local command every τ time instants. Here we introduce a distributed CG scheme based on the FF-CG approach presented in Chapter 3 assuming, as in Section 5.2, that the agents are connected via the communication network represented by the graph $\Gamma = (\mathcal{A}, \mathcal{B})$, defined in (5.1) with the related Hamiltonian cycle $\mathcal{H} = \{1, ...N\}$. The two key points to be considered in building up such a kind of strategy are:

a) the definition of the information set available to each agent
b) the determination of a set of decentralized "selection rules" such that the composition of all feasible local commands satisfies global constraints (3.19).

With regards to the information available to each agent, we will assume that each agent acts as a gateway in redistributing at each time instant data amongst the other, no directly connected, agents. Then, at each time instant t, the most recent information on computed commands available locally to the i-th agent i represented by the following **Local Information** vector:

$$\xi_i(t) = \left[g_1^T \left(t - \left\lceil \frac{d_{i,1}}{\tau} \right\rceil \tau \right), \ldots, g_i^T(t-\tau), \ldots, g_N^T \left(t - \left\lceil \frac{d_{i,N}}{\tau} \right\rceil \tau \right) \right]^T \quad (5.42)$$

Where $\lceil \cdot \rceil$ represents the ceiling operator for a scalar. It results that the **Common Information** on the applied commands shared by all agents at each time t is given by the vector

$$\xi(t) = \left[g_1^T \left(t - \left\lceil \frac{d_1}{\tau} \right\rceil \tau \right), ..., g_N^T \left(t - \left\lceil \frac{d_N}{\tau} \right\rceil \tau \right) \right]^T \quad (5.43)$$

where d_i is the maximum amongst all distances $d_{i,j}$ from the i-th agent to any other in the graph, i.e. $d_i = max_{j \in \mathcal{A}} d_{i,j}$.

The main idea behind the proposed selection rule is that of generating, every τ steps and on the basis of the information shared by all the agents in the network, a set of decoupled alternative constraints (one for each agent) such that their local fulfillment implies the fulfilment of global constraints

(3.19). In other words, at each computation time step, we will substitute the admissible region for Δg given by conditions (3.19) with its *Set Cartesian Decomposition* ([57]).

If such a decomposition is properly accomplished, the problem decouples and each agent will have simply to fulfill the inclusion into a local set in the form

$$\Delta g_i(t) \in \Delta\mathcal{G}_i(t), i = 1, ..., N \tag{5.44}$$

with $\Delta\mathcal{G}_i(t) \subseteq \mathbb{R}^{m_i}, i = 1, ..., N$ convex and compact sets containing 0_{m_i} for all $t \geq 0$.

It remains to understand how to generate local decoupled constraints guaranteeing global constraints satisfaction. The first step is to observe that if constraints (5.44) are satisfied at each time step, then we can define the set of all possible feasible values for $g(t)$ computed on the basis of the common information vector $\xi(t)$ as follows

$$\Xi(t) = \{\xi(t)\} \oplus \left(\bigoplus_{i=1}^{d_1} \Delta\mathcal{G}_1(t - i\tau) \times \ldots \times \bigoplus_{i=1}^{d_N} \Delta\mathcal{G}_N(t - i\tau) \right) \tag{5.45}$$

where $\oplus$ denotes the Pontryagin set sum (Definition (2.2)).

On the basis of the above feasible values of $g(t-\tau)$, the set of all admissible aggregate **command variations** can be computed as follows

$$\Delta\mathcal{V}(\Xi) := \left\{ \Delta g \;\middle|\; \begin{array}{r} \Delta g \in \Delta\mathcal{G}(g), \forall g \in \Xi \\ (g + \Delta g) \in \mathcal{W}_\delta, \quad \forall g \in \Xi \end{array} \right\} \tag{5.46}$$

Finally, the (approximated) set Cartesian decomposition giving rise the agent-wise decoupled constraints (5.44) should satisfy the following set inclusion condition

$$\Delta\mathcal{G}_1(t) \times \ldots \times \Delta\mathcal{G}_N(t) \subseteq \Delta\mathcal{V}(\Xi(t)), \tag{5.47}$$

$$0_i \in In(\Delta\mathcal{G}_1(t)), \ldots, 0_N \in In(\Delta\mathcal{G}_N(t)) \tag{5.48}$$

Note also that because $\Xi(t)$ is a common information shared by all agents in the network, a proper index need to be defined for all agents to independently compute the same collection of sets $\Delta\mathcal{G}_i(t), i = 1, ..., N$. We can finally describe the Parallel FF-CG procedure to be performed. It involves also the usage of S-FFCG described in section 5.1, then the P-FFCG procedure can be seen as a mix between the above described parallel approach and the S-FFCG where agents behave in a sequential or in parallel way according to precise events that can occur. In order to model this procedure one can consider four possible *operative scenarios*: a parallel scenario (**PAR**), a sequential scenario (**SEQ**) and further two scenarios (**P2S** and **S2P**) that are related with the transition between sequential and parallel scenarios. Each scenario can be summarized as follows

- **PAR:**In this scenario agents go parallel and switch to sequential scenario if the distance between $\Xi(t)$ and the border of $\mathcal{W}_\delta$ is smaller than a prefixed threshold ϵ_P. In particular

  1.1 Each agent computes the minimum distance between $\Xi(t)$ and the border of $\mathcal{W}_\delta$ as

$$\mu(t) = \min_{g \in \Xi(t), g' \in \partial(\mathcal{W}_\delta)} \|g - g'\| \tag{5.49}$$

  2.1 If $\mu(t)$ is bigger than a predetermined threshold ϵ_P

  2.1.1 Each agent determines the collection of sets $\Delta\mathcal{G}_i(t), i = 1, ..., N$ as the solution of its instance of the following optimization problem

$$\begin{array}{c} \max \limits_{\Delta\mathcal{G}_i(t), i=1,...,N} V(\Delta\mathcal{G}_1(t) \times ... \times \Delta\mathcal{G}_N(t)) \\ subject \quad to \ (5.47), (5.48) \end{array} \tag{5.50}$$

  where $V(\cdot)$ denotes a possible measure of the volume of a set (to achieve good dynamical properties we want $(\Delta\mathcal{G}_1(t) \times ... \times \Delta\mathcal{G}_N(t))$ to be as large as possible)

  2.1.2 each agent chooses its own reference by solving the following convex optimization problem

$$\begin{array}{c} g_i(t) = arg \min \limits_{g_i} \|g_i - r_i(t)\|^2_{\Psi_i} \\ subject \quad to \ (g_i - g_i(t-\tau)) \in \Delta\mathcal{G}_i(t) \end{array} \tag{5.51}$$

  2.2 Otherwise, go to **P2S**

- **P2S:** This is a transition scenario between **PAR** and **SEQ**. In this case only one agent updates according to the Hamiltonian cycle $\mathcal{H}$ while all other non-active agents set their local set $\Delta\mathcal{G}_i = 0$. More formally

  1.1 The agent in charge computes the following set on the basis of its most updated local information (5.42)

$$\Xi_i(t) = \{\xi_i(t)\} \oplus \Big(\bigoplus_{i=1}^{d_{i,1}} \Delta\mathcal{G}_1(t - i\tau) \times \ldots \times \bigoplus_{i=1}^{d_{i,N}} \Delta\mathcal{G}_N(t - i\tau) \Big) \tag{5.52}$$

  2.1 If Ξ_i is not a singleton it computes $g_i(t)$ as

$$\begin{array}{l} g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel^2_{\Psi_i} \\ subject \quad to \ (g_i - g_i(t-\tau)) \in \Delta\mathcal{V}_i^0(\Xi_i(t)) \end{array} \tag{5.53}$$

  where

$$\Delta\mathcal{V}_i^0(\Xi_i) := \left\{ \Delta g_i \in \mathbb{R}^{m_i} \,|[0_1^T, ..., \Delta g_i^T, ..., 0_N^T]^T \in \Delta\mathcal{V}(\Xi_i) \right\}. \tag{5.54}$$

  2.2 Otherwise go to scenario **SEQ**

- **SEQ:**In this case agents perform the S-FFCG algorithm described in Section 5.1 and can recover to the **PAR** scenario when the distance between the global command $g(t-\tau)$ and the border of $\mathcal{W}_\delta$ is bigger than a prefixed threshold ϵ_S. When the trigger event occurs, only the agent in charge is capable of catching it because the actual applied global command $g(t-\tau)$, is known and, before switching to **S2P**, the agent sends an acknowledgment consisting in a number $\sigma_i(t) = d_{max}$ where $d_{max} = \max_j d_j$. More in detail the agent in charge
  1.1 computes
  $$\mu(t) = \min_{g \in \partial(\mathcal{W}_\delta)} \|g - g(t-\tau)\| \tag{5.55}$$
  2.1 If $\mu(t)$ is smaller than ϵ_S computes $g_i(t)$ by means of (5.7).
  3.1 Otherwise it sets $\sigma_i(t) = d_{max}$, sends $\sigma_i(t)$ to neighbors and goes to **S2P**.

  On the contrary the agent not in charge has to check if an acknowledgment $\sigma_j(t)$ arrives from one of its neighbors and, in this case, it sets $\sigma_i(t) = \sigma_j(t) - 1$, communicates $\sigma_i(t)$ to the neighbors and goes to **S2P**.
- **S2P:** This is a transition scenario between **SEQ** and **PAR** that represents an initialization phase for the parallel procedure. In this case agents decrements $\sigma_i(t)$, when $\sigma_i(t) = 0$ that means that all agents knows the last applied command vector $g(t)$ then they can switch to **PAR** and the parallel procedure can restart with $\Xi(t) = \{g(t-\tau)\}$.

Remark 5.23. It is worth to remark that in **P2S** scenario after d_{max} time steps $\Xi_i(t), \forall i \in \mathcal{A}$ reduces to a singleton, hence the transition to **SEQ** scenario is ensured.

Remark 5.24. For scenarios **PAR** and **SEQ** we have considered two different thresholds ϵ_P and ϵ_S that represent design knobs for the P-FFCG. In particular they have to be chosen so that $\epsilon_P < \epsilon_S$. This expedient avoids a possible huge number of switches between **PAR** and **SEQ** that, although do not prevent agents from reaching their objective, slows down the updating process of the command $g(t)$.

The algorithm that we want to present is implemented by means of the state-finite automata depicted in Figure 5.4. From an implementation point of view, the involvement of such a hybrid scenario results in the introduction of the logical variable $st_i \in \{"PAR", "SEQ", "S2P", "P2S"\}$ that indicates in which scenario the i-th agent is operating. More formally, it can be presented by the following "parallel" version of distributed FF-CG algorithm:

Algorithm 5.4.1 ***Parallel-FFCG Algorithm (P-FFCG)*** *- Agent i*
AT EACH TIME t
1.1 RECEIVE $\xi_j(t), \sigma_j(t)$ FROM ALL NEIGHBORS $j \in \mathcal{N}_i$

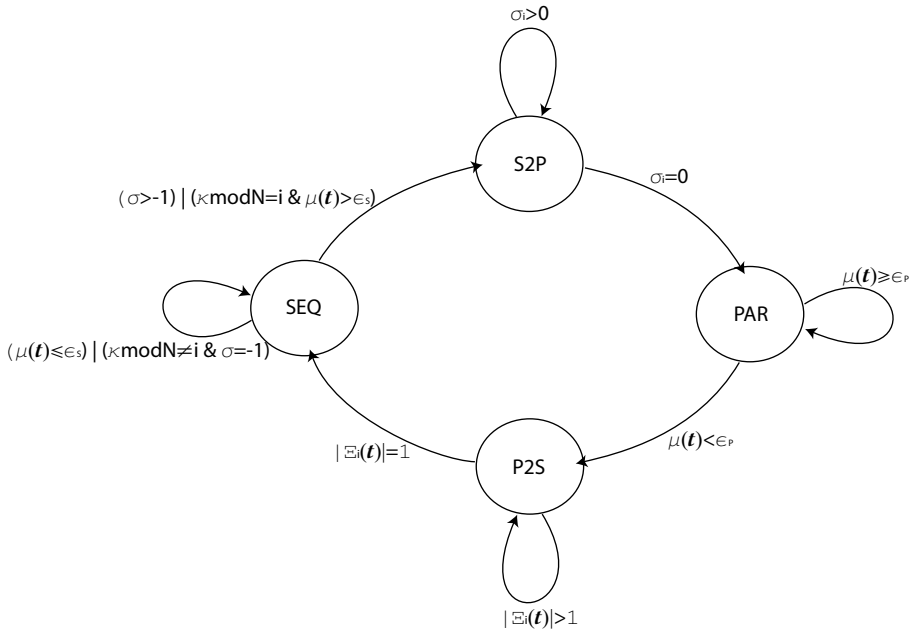

Fig. 5.9. Finite-state automata that describes transition between scenarios in Algorithm 5.4.1

1.2 GO TO $st_i(t)$

PAR:
1.1 IF$(t == \kappa\tau, \kappa = 0, 1, \ldots)$
  1.1.1 SOLVE (5.49)
  1.1.2 IF $\mu(t) \geq \epsilon_P$
    1.1.2.1 SOLVE (5.50)
    1.1.2.2 SOLVE (5.51)
  1.1.3 ELSE
    1.1.3.1 SET $st_i(t) =$ "$P2S$"
    1.1.3.2 GO TO $st_i(t)$
1.2 ELSE
  1.2.1 SET $g_i(t) = g_i(t-1)$
1.3 UPDA TE $\xi_i(t)$
1.4 TRANSMIT $\xi_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.5 APPLY $g_i(t)$
1.6 SET $st_i(t+1) = st_i(t)$

P2S:
1.1 COMPUTE $\Xi_i(t)$ AS IN (5.52)
1.2 IF $|\Xi_i(t)| > 1$

1.2.1 IF($t==\kappa\tau\&(\kappa \bmod N)=i, \kappa=0,1,\ldots$), SOLVE (5.53)
1.2.2 ELSE, SET $g_i(t)=g_i(t-1)$
1.2.3 UPDATE $\xi_i(t)$
1.2.4 TRANSMIT $\xi_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.2.5 APPLY $g_i(t)$
1.2.6 SET $\Delta\mathcal{G}_i(t)=0_{m_i}$
1.3 ELSE
1.3.1 SET $st_i(t)=$"SEQ"
1.3.2 GO TO $st_i(t+1)=st_i(t)$

SEQ:
1.1IF $\sigma_j(t-1)>-1, j\in\mathcal{N}_i$
1.1.1 SET $\sigma_i(t)=\sigma_j(t)-1$
1.1.2 TRANSMIT $\sigma_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.1.3 SET $st_i(t)=$"$S2P$"
1.1.4 GO TO $st_i(t)$
1.2ELSE
1.2.1 IF($t==\kappa\tau\&(\kappa \bmod N)=i, \kappa=0,1,\ldots$)
1.2.1.1 SOLVE (5.55)
1.2.1.2 IF $\mu(t)\leq\epsilon_S$,SOLVE(5.7)
1.2.1.3 ELSE
1.2.1.3.1 SET $\sigma_i(t)=d_{max}$
1.2.1.3.2 TRANSMIT $\sigma_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.2.1.3.3 SET $st_i(t)=$"$S2P$"
1.2.1.3.4 GO TO $st_i(t)$
1.2.2 ELSE, SET $g_i(t)=g_i(t-1)$
1.2.3 UPDATE $\xi_i(t)$
1.2.4 TRANSMIT $\xi_i(t)$ AND $\sigma_i(t)=-1$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.2.5 APPLY $g_i(t)$
1.2.6 SET $\Delta\mathcal{G}_i(t)=0_{m_i}$
1.2.7 SET $st_i(t+1)=st_(t)$

S2P:
1.1IF $\sigma_i(t)=0$
1.1.1 SET $st_i(t)=$"PAR"
1.1.2 GO TO $st_i(t)$
1.2ELSE
1.2.1 SET $\sigma_i(t+1)=\sigma(t)-1$
1.2.2 TRANSMIT $\sigma_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.2.3 UPDATE $\xi_i(t)$
1.2.4 TRANSMIT $\xi_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.2.5 APPLY $g_i(t)=g_i(t-1)$
1.2.6 SET $\Delta\mathcal{G}_i(t)=0_{m_i}$
1.2.7 SET $st_i(t+1)=st_i(t)$

Finally, we can present some properties enjoyed by the P-FFCG scheme.

Theorem 5.25. *Let assumptions* **A1-A2-A3** *be fulfilled for the system arising from the composition of N subsystems in form (5.1). Let consider the distributed* ***P-FFCG*** *Algorithm (5.4.1) and let an admissible aggregate command signal $g(0) = [g_1^T(0), \ldots, g_N^T(0)]^T \in \mathcal{W}_\delta$ be applied at $t = 0$ such that (3.8) holds true. Then*

1. *for each agent $i \in \mathcal{A}$, at each time $t = k\tau, k \in \mathbb{Z}_+$, the minimizer related to one of the problems (5.51), (5.53), (5.7), to be performed depending on the value of $st_i, i \in \mathcal{A}$, uniquely exists and can be obtained by solving a convex constrained optimization problem;*
2. *The overall system acted by agents implementing the P-FFCG supervisory policy never violates the constraints, i.e. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$.*
3. *Whenever $r_i(t) \equiv r_i$, $\forall i \in \mathcal{A}$ with r_i constant set-points the sequence of $g(t) = [g_1^T(t), \ldots, g_N^T(t)]^T$'s asymptotically converges either to $r = [r_1^T(t), \ldots, r_N^T(t)]^T$ if $r \in \mathcal{W}_\delta$ or to a point $\hat{r}$. that is Pareto-Optimal for the problem (5.9).*

Proof

1) The existence of an admissible solution for each agent at each time $k\tau$ can be proved by simply remarking that $g_i(t) = g_i(t-\tau), i = 1, \ldots, N$, is always an admissible, although not necessarily optimal, solution for the prescribed problems at time t, in any state st_i. In fact, the sets $\Delta\mathcal{V}(\Xi(t))$ and $\Delta\mathcal{V}(\Xi_i(t)), \forall i \in \mathcal{A}$ always contain the point 0_m (note that $\Xi(t)$ and $\Xi_i(t), \forall i \in \mathcal{A}$ are contained in $\mathcal{W}_\delta$).
2) At each time $t = k\tau$, with $k \in \mathbb{Z}_+$, from a centralized point of view, a command $g(k\tau)$ complying with (3.19) is applied to the overall plant. By construction, the latter implies that the set-valued virtual predictions along the virtual time i defined in (2.8) satisfy

$$\bar{c}(i, x(k\tau), g(k\tau)) \in \mathcal{C}, \forall i \in \mathbb{Z}_+,$$

Then, the statement is proved by simply noticing that the following inclusion

$$c(t) = c(i, x(k\tau), g(k\tau)) \in \mathcal{C},$$

holds true for all time instants $t = k\tau + i$, $i \in \{0, 1, ..., \tau - 1\}$ and by repeating the same argument for all $k \in \mathbb{Z}_+$.
3) The convergence follows simply because the sequences of solutions $g_i(t)$ are such that the local costs $||g_i(t) - r_i||^2_{\Psi_i}$ are non increasing for any $i = 1, ..., N$. In fact, there is not convenience for the agents to modify their actual optimal solutions if the costs could not be minimized further on. To this end, let $g_i(t)$ be the P-FFCG local action at time t, solution of any optimization problem among (5.51), (5.53), (5.7). As already seen in item 1) of this proof, at time $t + \tau$, $g_i(t)$ is still an admissible, though not

necessarily the optimal, solution and hence the sequences $||g_i(t) - r_i||^2_{\Psi_i}$ are non increasing, i.e. $\| g_i(t+\tau) - r_i \|^2_{\Psi_i} \leq \| g_i(t) - r_i \|^2_{\Psi_i}$.
Then, we want to show that any stationary optimal solution, viz. $g(t) = g(t+1)\ \forall t$, is Pareto Optimal by proving that a solution is not stationary if is not Pareto Optimal. To this end, let $g' = [g_1'^T, ..., g_N'^T]^T$ be the actual solution at time $t = k\tau, k \in \mathbb{Z}_+$ which is assumed to be not Pareto Optimal. As a consequence, other different solutions exist which improve the costs. Supposedly, vectors $v = [v_1^T, ..., v_N^T]^T \in \mathbb{R}^m$ would exist with $g' + v \in \mathcal{W}_\delta$, such that

$$||g_i' + v_i - r_i||^2_{\Psi_i} - ||g_i' - r_i||^2_{\Psi_i} \leq 0, \tag{5.56}$$

happens to hold for all $i \in \mathcal{A}' := \{i \in \mathcal{A} : v_i \neq 0\}$ with some of the above inequalities becoming strict for at least one index $i \in \mathcal{A}'$.
As already stated in item 3) of the proof of Theorem 5.7, the existence of a vector v satisfying (5.56) makes it convenient for all agents to move from g'. Hence we have to show that the policy underlying Algorithm 5.4.1 guarantees that at least one agent can modify its command g_i'. To this end, it is sufficient to prove that in scenario "PAR" and "SEQ" the cost will decrease for at least on agent. In fact "P2S" and "S2P" represent transient scenarios because the algorithm could dwell there at most d_{max} time steps. If at time t $st_i(t)$ ="SEQ",$\forall i \in \mathcal{A}$ and $\mu(t) \leq \epsilon_S$, where $\mu(t)$ is computed as in (5.55), then, the agents will go on sequentially and, because the discussion of point 3) of the proof of Theorem 5.7, at least one agent can change its command. Conversely, if $\mu(t) > \epsilon_S$, agents will not move in "SEQ" but they will switch on "S2P" and subsequently, at a time t', such that $t' - t \leq d_{max}$, in "PAR" with $\Xi(t') = \{g'\}$ reduced to a singleton. In such a situation $\mu(t')$ computed as in (5.49) will coincide with $\mu(t)$ in (5.55) and then it will be for sure bigger than ϵ_P because

$$\mu(t') = \mu(t) > \epsilon_S \geq \epsilon_P. \tag{5.57}$$

The latter and Proposition 3.5 imply that the set $\Delta\mathcal{V}(\Xi(t'))$, that will be in this case

$$\Delta\mathcal{V}(\Xi(t')) = \left\{ \Delta g \,\middle|\, \begin{matrix} \Delta g \in \Delta\mathcal{G}(g') \\ (g' + \Delta g) \in \mathcal{W}_\delta \end{matrix} \right\} \tag{5.58}$$

will contain a ball of finite radius centered at 0_m. As consequence the sets $\Delta\mathcal{G}_i(t'), \forall i \in \mathcal{A}$, derived by means of (5.50), will also contain a ball of finite radius centered at 0_{m_i} that allows each agent to move in any subdirection $v_i \in \mathbb{R}^{m_i}$.
Similar considerations are valid when, at time t, $st_i(t)$ ="PAR" and $\Xi(t)$ is not a singleton. In this case if $\mu(t)$, computed as in (5.49), is bigger than ϵ_P, it can be proved that $\Delta\mathcal{V}(\Xi(t))$ even contains a ball of finite radius (see Lemma 5.31 in the Appendix) and, by following the above discussion, we can conclude that all agents are capable to modify their command w.r.t g'. On the contrary if $\mu(t) < \epsilon_P$, agents will change their status st_i first in

"P2S" and then in "SEQ" and as already seen, in the worst case, they will be enabled to change their command after $2d_{max}$ steps, when their status st_i will again equal "PAR".
Thus, if we are not at a Pareto-optimal solution at time t, at least one of the agents will move from it by a finite amount of time steps.

Remark 5.26. Note that, because of **A1** and **A2** whenever g converges to a point $\hat{r}$ then

$$\lim_{t\to\infty} \hat{x}(t) = x_{\hat{r}}, \quad \lim_{t\to\infty} \hat{y}(t) = y_{\hat{r}} = \hat{r}, \quad \lim_{t\to\infty} \hat{c}(t) = c_{\hat{r}}. \tag{5.59}$$

Remark 5.27. It is worth noticing that this distributed strategy requires low data exchange rates to be implemented. Anyway, the communication loads do not affect the on-line execution times and computational burdens because data are exchanged only after the computation and application of local commands.

Remark 5.28. Note that, although the scalability of the P-FFCG is guaranteed (the dimension of the problem to be solved by each agent grows at most polynomially w.r.t. the numbers of constraints of the corresponding centralized problem), any single agent needs to know the sets $\mathcal{W}_\delta$ and $\Delta\mathcal{G}(g), \forall g \in \mathcal{W}_\delta$ associated to the entire aggregate system. Even if such a feature reduces the method flexibility, at the best of our acknowledge, it cannot be avoided in the general case unless a restricted (usually in a very conservative way) set of admissible steady-state set-points $\mathcal{W}_\delta$ is used.

5.4.1 Computational Details

In what follows a wide description of computation details related to above parallel strategy is presented. In particular, all technicalities required in order to implement the optimizations problems (5.51), (5.49), (5.55) and (5.53) will be considered.

The main difficulty in implementing Problem (5.51) relies in the computation of the set $\Delta\mathcal{V}(\Xi)$ in (5.46) and its Cartesian decomposition arising in (5.50). We show first how to compute $\Delta\mathcal{V}(\Xi)$ and, its Cartesian decomposition is then derived by means of the inner-boxing procedure described in [54]. The proposed scheme also represents the first step for computing the multi-box inner approximation presented in Section 5.3. Next, each agent, by solving problem (5.50), gets a box $\mathcal{B}(\underline{\Delta g}(t), \overline{\Delta g}(t)) = \Delta\mathcal{G}_1(t), \times, ..., \times\Delta\mathcal{G}_N(t)$ containing 0_m and can choose its local command in the following simple way

$$\begin{aligned} g_i(t) &= arg\min_{g_i} \|g_i - r_i(t)\|^2_{\Psi_i} \\ subject \quad to \ \underline{\Delta g}_i &\leq (g_i - g_i(t-\tau)) \leq \overline{\Delta g}_i \end{aligned} \tag{5.60}$$

that represents the rephrasing of problem (5.51).

Moreover the box structure for $\Delta\mathcal{G}_i(t)$ implies a box shape also for the set Ξ in (5.45) that can be uniquely identified by only two vectors according to Definition (5.16)

$$\Xi = \mathcal{B}(\underline{\xi}, \overline{\xi}). \tag{5.61}$$

The notation in (5.61) is useful to easily compute the set $\Delta\mathcal{V}(\Xi)$. The drawback, in this case, could arise from the fact that, according to Definition (5.46), it is needed to calculate sets $\Delta\mathcal{G}(g)$ for an infinite number of points $g \in \Xi$. A solution can be arranged in two steps by computing an inner-approximation of the set $\Delta\mathcal{V}(\Xi)$. The first step consists in exploiting the polyhedral structure of the set $\mathcal{C}$ (2.18)-(2.21) and separately considering the half-spaces

$$\mathcal{C}^j := \{c \in \mathbb{R}_c^n : T_j^T c \leq q_j\} \tag{5.62}$$

whose intersection constitute the constraints polyhedron $\mathcal{C} = \bigcap_{j=1}^{z} \mathcal{C}^j$. Then for each set $\mathcal{C}^j$ we compute

$$\Delta\mathcal{V}^j(\Xi) = \{\Delta g | \Delta g \in \Delta\mathcal{G}^j(g), \forall g \in \Xi\} \cap \{\Delta g | (g+\Delta g) \in \mathcal{W}_\delta^j, \forall g \in \Xi\} \tag{5.63}$$

where

$$\mathcal{W}_\delta^j := \{g \in \mathbb{R}^m : c_g \in \mathcal{C}_\delta^j\} \tag{5.64}$$

and

$$\Delta\mathcal{G}^j(g) := \{\Delta g : \|H_c \Phi^k (I - \Phi)^{-1} G \Delta g\| \leq \rho_{g+\Delta g}^j - \gamma \rho_g^j\} \tag{5.65}$$

with

$$\rho_{g+\Delta g}^j = \frac{q_j - T_j^T c_{g+\Delta g}}{\|T_j\|}, \tag{5.66}$$

$$c_g = H_c (I_n - \Phi)^{-1} G. \tag{5.67}$$

Hence if we did compute $\Delta\mathcal{V}^j(\Xi), j = 1, \ldots, z$ we would have an inner-approximation of $\Delta\mathcal{V}(\Xi)$ because of the following lemma

Lemma 5.29. *The intersection of sets $\Delta\mathcal{V}^j(\Xi)$ is such that* $\bigcap_{j=1}^{z} \Delta\mathcal{V}^j(\Xi) \subseteq \Delta\mathcal{V}(\Xi)$.

Proof
see the Appendix.

The second step consists of computing each set $\Delta\mathcal{V}^j(\Xi), j = 1, ..., z$. They can be determined in an easy way by finding for each hyperplane of $\mathcal{W}_\delta$ the point $\bar{g}^j \in \Xi$ that minimizes the distance between the region Ξ and the j-th hyperplane $T_j^T c_g = q_j$ of $\mathcal{W}_\delta$. Such a point is a solution of the following LP problem

$$\bar{g}^j = \arg\min_{g \in \Xi} \rho_g^j. \tag{5.68}$$

It is worth to remark that the constraints $g \in \Xi$ simply indicates that, by resorting to definition (5.61), the point g is between two vectors, i.e. $\underline{\xi} \leq g \leq \bar{\xi}$. In Figure 5.10 points $\bar{g}^j$ are depicted in the case where $\mathcal{W}_\delta$ is the intersection of two hyperplanes.

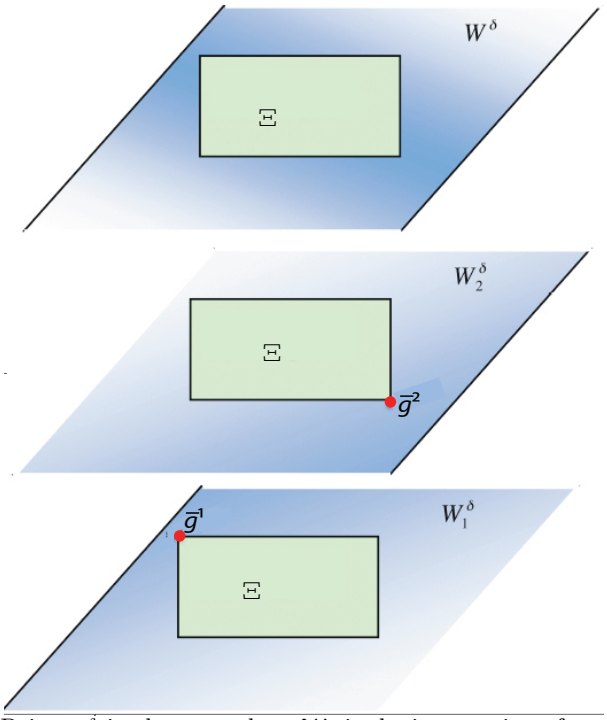

Fig. 5.10. Points $\bar{g}^j$ in the case where $\mathcal{W}_\delta$ is the intersection of two hyperplanes

Then the set

$$\Delta\mathcal{V}^j(\{\bar{g}^j\}) := \Delta\mathcal{G}^j(\bar{g}^j) \cap \{\Delta g : \bar{g}^j + \Delta g \in \mathcal{W}_\delta^j\}. \tag{5.69}$$

can be used in place of $\Delta\mathcal{V}^j(\Xi)$ because this choice is justified by the following lemma

Lemma 5.30. *The set $\Delta\mathcal{V}^j(\{\bar{g}^j\})$ is always contained or at the most coincides with $\Delta\mathcal{V}^j(\Xi)$, i.e. $\Delta\mathcal{V}^j(\{\bar{g}^j\}) \subseteq \Delta\mathcal{V}^j(\Xi)$.*

Proof
see the Appendix.

An immediate consequence of Lemmas 5.29 and 5.30 results is that

$$\bigcap_{j=1}^{z} \Delta\mathcal{V}^j(\{\bar{g}^j\}) \subseteq \Delta\mathcal{V}(\Xi) \tag{5.70}$$

Hence, each agent can use $\Delta\tilde{\mathcal{V}}(\Xi) = \bigcap_{j=1}^{z} \Delta\mathcal{V}^j(\{\bar{g}^j\})$ instead of $\Delta\mathcal{V}(\Xi)$ as the set of feasible command variations in solving problem (5.50).

In order to set up the problem (5.51) it remains to illustrate the box approximation underlying problem (5.50) for the determination of sets $\Delta\mathcal{G}_i, \forall i \in \mathcal{A}$ from $\Delta\tilde{\mathcal{V}}(\Xi)$. Hereinafter we provide a detailed procedure to solve it that exploits an algorithm borrowed from [54] and determine a box inscribed in a polyhedron as depicted in Figure 5.11. For this reason, we will focus on the case where set $\mathcal{C}$ is defined as a box that give rise to a polyhedral form for $\Delta\tilde{\mathcal{V}}(\Xi)$. Let $\mathcal{C}$ be defines as

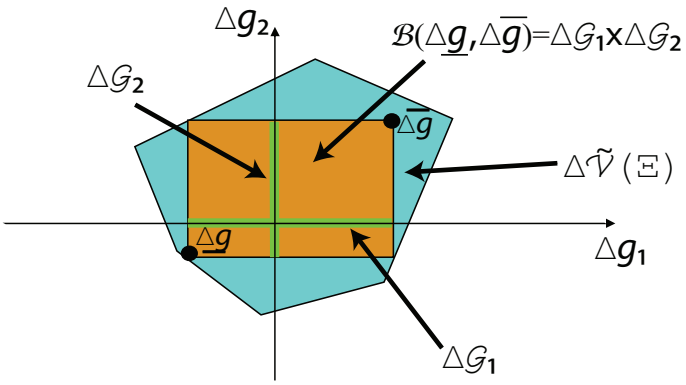

Fig. 5.11. Inner-box approximation of the set $\Delta\tilde{\mathcal{V}}(\Xi)$ in the case where 2 agents are operating in a bi-dimensional decision space

$$\mathcal{C} := \{c \in \mathbb{R}^{n_c} \ : \ \underline{q}_s \leq c_s \leq \bar{q}_s, s = 1, \ldots, n_c.\} \tag{5.71}$$

with $\underline{q}_s \in \mathbb{R}$ and $\bar{q}_s \in \mathbb{R}$ representing respectively the maximum and the minimum value that the i-th component of the constrained vector c may assume. Then we can easily decompose the constraint set $\mathcal{C}$ into the Cartesian product of n_c sets of scalar intervals:

$$\mathcal{C} = \mathcal{C}^{(1)} \times ... \times \mathcal{C}^{(s)} \times ... \times \mathcal{C}^{(n_c)} \tag{5.72}$$

where

$$\mathcal{C}^{(s)} := \left\{ c_s \in \mathbb{R} \,\middle|\, \underline{q}_s \leq c_s \leq \bar{q}_s \right\}. \tag{5.73}$$

By defining q as $q := [\bar{q}_1, \ldots, \bar{q}_s \ldots, \bar{q}_{n_c}, -\underline{q}_1, \ldots, -\underline{q}_s \ldots, -\underline{q}_{n_c}]^T$ the set $\mathcal{C}$ can be seen as

$$\mathcal{C} = \left\{ c \in \mathbb{R}^{n_c} \quad : \quad \begin{bmatrix} I_{n_c} \\ -I_{n_c} \end{bmatrix} c \leq q \right\}. \tag{5.74}$$

The latter definition of $\mathcal{C}$ specializes (2.18) when $T = [I_{n_c}, -I_{n_c}]^T$. By means of this particular structure one can rephrase the sets $\Delta\mathcal{G}^j(g)$ in (5.65) in the following very easy computable form

$$\Delta\mathcal{G}_g^{(j)} := \left\{ \Delta g \in \mathbb{R}^n : |H_c \Phi^k (I - \Phi)^{-1} G \Delta g| \leq \rho_{g+\Delta g}^{(j)} - \gamma \rho_g^{(j)}, k = 0, ..., k_0 \right\} \tag{5.75}$$

where $\rho_{g+\Delta g}^{(j)} = q_j - c_{g+\Delta g}$ and k_0 is the constraint horizon computed in the Algorithm (2.3.1). Hence, once computed $\bar{g}^j$ in (5.68), the set $\Delta\mathcal{V}^j(\bar{g}^j)$ in (5.69) takes the following matrix form

$$\Delta\mathcal{V}^j(\bar{g}^j) = \left\{ \Delta g \left| \begin{array}{c} S_{c,j}\Delta g \leq q_j - \delta[\sqrt{T_j^T T_j}] - S_{c,j}\bar{g}^j \\ (R_{c,j}^k + S_{c,j})\Delta g \leq q_j - S_{c,j}\bar{g}^j, \quad k = 0, ..., k_0 \\ (-R_{c,j}^k + S_{c,j})\Delta g \leq q_j - S_{c,j}\bar{g}\cdot j \quad k = 0, ..., k_0 \end{array} \right. \right\} \tag{5.76}$$

where $S_{c,j} = T_j^T H_c (I_n - \Phi)^{-1} G + L$ and $R_{c,j}^k = T_j^T H_c \Phi^k (I_n - \Phi)^{-1} G$. Eq. (5.76) allows one to express $\Delta\tilde{\mathcal{V}}(\Xi)$ as a convex polyhedron defined as

$$\tilde{A}\Delta g \leq \tilde{b} \tag{5.77}$$

with

$$\tilde{A} = \begin{bmatrix} S_{c,1} \\ R_{c,1}^k + S_{c,1} \\ -R_{c,1}^k + S_{c,1} \\ \vdots \\ S_{c,z} \\ R_{c,z}^k + S_{c,z} \\ -R_{c,z}^k + S_{c,z} \end{bmatrix} \in \mathbb{R}^{(z(1+2k_0))\times m}, \quad \tilde{b} = \begin{bmatrix} q_1 - \delta[\sqrt{T_1^T T_1}] - S_{c,1}\bar{g}^1 \\ q_1 - S_{c,1}\bar{g}^1 \\ q_1 - S_{c,1}\bar{g}^1 \\ \vdots \\ q_z - \delta[\sqrt{T_z^T T_z}] - S_{c,z}\bar{g}^z \\ q_z - S_{c,z}\bar{g}^z \\ q_z - S_{c,z}\bar{g}^z \end{bmatrix} \in \mathbb{R}^{(z(1+2k_0)} \tag{5.78}$$

where $z = 2n_c$. By exploiting the matrix description of $\Delta\tilde{\mathcal{V}}(\Xi)$ problem (5.50) becomes

$$\max_{\underline{\Delta g} \in \Delta\tilde{\mathcal{V}}(\Xi), \overline{\Delta g} \in \Delta\tilde{\mathcal{V}}(\Xi)} V(\mathcal{B}(\underline{\Delta g}, \overline{\Delta g})) \tag{5.79}$$

Several polynomial time algorithms are presented in [54] for computing the box with maximum volume contained in a convex polytope described in matrix form. One of these methods has been here extended into the more general case where inequalities (5.77) describe a polyhedron. Hence, an algorithm that solve problem (5.79) can be formulated as

$$\max_{\underline{\Delta g}, \Delta g^+} \quad \sum_{i=1}^{m} \ln \Delta g_i^+ \tag{5.80}$$

$$s.t.\ \tilde{A}\underline{\Delta g} + \tilde{A}^+ \Delta g^+ \leq b \tag{5.81}$$

$$\underline{\Delta g} \leq -\varepsilon \tag{5.82}$$

$$\underline{\Delta g} + \Delta g^+ \geq \varepsilon \tag{5.83}$$

where $\Delta g^+ := \overline{\Delta g} - \underline{\Delta g}$ and $\tilde{A}^+$ is the positive part of $\tilde{A}$ namely $\tilde{a}_{i,j}^+ := max(0, \tilde{a}_{i,j})$, please refer to [54] for details. Constraints (5.82) and (5.83) are added for two reasons, the first one is that the resulting sets $\Delta\mathcal{G}_i = \mathcal{B}(\underline{\Delta g}_i, \overline{\Delta g}_i), i \in \mathcal{A}$ will contain a ball of finite radius ε that contains 0_{m_i} (this guarantees that in the "PAR" scenario each agent has a margin of freedom in determining a variation for g_i), the second one is related with the fact that the set $\{(\underline{\Delta g}, \Delta g^+) \in \mathbb{R}^{2m} : \tilde{A}\underline{\Delta g} + \tilde{A}^+ \Delta g^+ \leq \tilde{b}\}$ could be not limited and then not compact. The Lemma 5.32 in Appendix proves that the set generated by (5.81)-(5.83) is limited.

Concerning Problems (5.49) and (5.55), from the above discussion we can directly conclude that $\mu(t)$ in (5.49) can be computed during the "PAR" scenario as

$$\mu(t) = \min_j \rho^j_{\bar{g}^j(t)}, j = 1, ..., z \tag{5.84}$$

where $\bar{g}^j(t)$ is the optimal cost of the j-th problem (5.68) achieved at time t while $\mu(t)$ in (5.55) is determined in the "SEQ" scenario as

$$\mu(t) = \min_j \rho^j_{g(t-\tau)}, j = 1, ..., z. \tag{5.85}$$

Finally Problem (5.53) can be implemented in two steps

1. Approximate $\Delta\mathcal{V}(\Xi_i)$ with $\Delta\tilde{\mathcal{V}}(\Xi_i) = \{\Delta g \in \mathbb{R}^m : \tilde{A}\Delta g \leq \tilde{b}\}$ by following procedure outlined in (5.61)-(5.77) by considering Ξ_i in the place of Ξ.
2. Compute $\Delta\mathcal{V}^0(\Xi_i)$ in (5.54) as

$$\Delta\mathcal{V}^0(\Xi_i) = \{\Delta g_i \in \mathbb{R}^{m_i} : \tilde{A}[0_1^T, ..., \Delta g_i^T, ..., 0_n^T]^T \leq \tilde{b}\} \tag{5.86}$$

Then, problem (5.53) at time t becomes

$$\begin{array}{l} g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel^2_{\Psi_i} \\ subject \quad to \ \ \tilde{A}^{(i)}(g_i - g_i(t-\tau)) \leq b \end{array} \tag{5.87}$$

where $\tilde{A}^{(i)}$ collects all the m_i columns of $\tilde{A}$ that multiply Δg_i in (5.86).

5.5 Conclusions

In this Chapter, two distributed FFCG schemes have been described for the supervision of dynamically coupled interconnected linear systems subject to

local and global constraints and used for solving constrained coordination problems in networked control system. Two distributed coordination algorithms have been outlined and the results on constraints fulfillment and stability deeply highlighted.

5.6 Appendix

Lemma 5.31. *Let μ be a finite scalar computed as in (5.49). Then, if $\mu \geq \epsilon_P > 0$, there exists a scalar $\eta^* > 0$ such that the set $\Delta\mathcal{V}(\Xi)$ contains a ball of radius η^*, i.e. $\exists \bar{\eta} > 0 : \Delta\mathcal{V}(\Xi) \supseteq \mathcal{B}_{\bar{\eta}},\ \forall g \in \mathcal{W}_\delta$;*

Proof - Notice that the set $\Delta\mathcal{V}(\Xi)$ in (5.46) can be rephrased as

$$\Delta\mathcal{V}(\Xi) = \{\Delta g \in \mathbb{R}^m : \Delta g \in \Delta\mathcal{G}(g), \forall g \in \Xi\} \cap \{\Delta g \in \mathbb{R}^m : g + \Delta g \in \mathcal{W}_\delta, \forall g \in \Xi\} \tag{5.88}$$

The statement is then is trivially proved because it isufficies to observe that:

1. because Ξ and the border of $\mathcal{W}_\delta$ are separated from a finite distance μ, each point $g \in \Xi$ could be the center of a ball with radius $\eta_1 \geq \mu$ entirely contained in $\mathcal{W}_\delta$, i.e. $g + \mathcal{B}_{\eta_1} \in \mathcal{W}_\delta, \eta \geq \mu, \forall g \in \Xi$. Then the set $\{\Delta g \in \mathbb{R}^m : g + \Delta g \in \mathcal{W}_\delta, \forall g \in \Xi\}$ in (5.88) always contains a ball of finite radius μ.
2. Item 3) of Proposition (3.5) implies that $\{\Delta g \in \mathbb{R}^m : \Delta g \in \Delta\mathcal{G}(g), \forall g \in \Xi\}$ contains a ball of finite radius η^*;

As consequence $\Delta\mathcal{V}(\Xi) \supseteq \mathcal{B}_{\bar{\eta}}$ where $\bar{\eta} = \min\{\mu, \eta^*\}$.

Proof of Lemma 5.29

This lemma can be easily proved by observing first that $\mathcal{W}_\delta = \bigcap_{j=1}^{z} \mathcal{W}_\delta^j$ implies

$$\bigcap_{j=1}^{z} \{\Delta g | (g + \Delta g) \in \mathcal{W}_\delta^j, \forall g \in \Xi\} \equiv \{\Delta g | (g + \Delta g) \in \bigcap_{j=1}^{z} \mathcal{W}_\delta^j, \forall g \in \Xi\} \equiv \{\Delta g | (g + \Delta g) \in \mathcal{W}_\delta, \forall g \in \Xi\} \tag{5.89}$$

Moreover $\Delta\mathcal{G}(g)$ can be rewritten as

$$\Delta\mathcal{G}(g) = \bigcap_{j=1}^{z} \Delta\mathcal{G}^j(g) \tag{5.90}$$

since ρ_g in (5.5) equals $\min_j \rho_g^j$ where ρ_g^j is defined in (5.66). Eq. (5.90) allows us to state that

$$\bigcap_{j=1}^{z}\{\Delta g|\Delta g\in\Delta\mathcal{G}^j(g),\forall g\in\Xi\}\equiv\{\Delta g|\Delta g\in\bigcap_{j=1}^{z}\Delta\mathcal{G}^j(g),\forall g\in\Xi\}\equiv \{\Delta g|\Delta g\in\Delta\mathcal{G}(g),\forall g\in\Xi\}, \tag{5.91}$$

Then, from (5.89) and (5.91) we obtain

$$\bigcap_{j=1}^{z}\{\Delta g|\Delta g\in\Delta\mathcal{G}^j(g),\forall g\in\Xi\}\cap\bigcap_{j=1}^{z}\{\Delta g|(g+\Delta g)\in\mathcal{W}_\delta^j,\forall g\in\Xi\}\equiv\Delta\mathcal{V}(\Xi), \tag{5.92}$$

and by exploiting basic properties of the $\cap$ operator we can state that

$$\Delta\mathcal{V}(\Xi)\equiv\bigcap_{j=1}^{z}\{\Delta g|\Delta g\in\Delta\mathcal{G}(g)^j,\forall g\in\Xi\}\cap\bigcap_{j=1}^{z}\{\Delta g|(g+\Delta g)\in\mathcal{W}_\delta^j,\forall g\in\Xi\}\equiv$$
$$\bigcap_{j=1}^{z}\left(\{\Delta g|\Delta g\in\Delta\mathcal{G}(g)^j,\forall g\in\Xi\}\cap\{\Delta g|(g+\Delta g)\in\mathcal{W}_\delta^j,\forall g\in\Xi\}\right)\equiv\bigcap_{j=1}^{z}\Delta\mathcal{V}^j(\Xi). \tag{5.93}$$

Proof of Lemma 5.30

In order to prove the claim, because of the definition of $\Delta\mathcal{V}^j(\Xi)$ in (5.63) and $\Delta\mathcal{V}^j(\bar{g}^j)$ in (5.69), it is sufficient to prove that

$$\Delta\mathcal{G}^j(\bar{g}^j)\subseteq\{\Delta g\in\mathbb{R}^m\,|\Delta g\in\Delta\mathcal{G}^j(g),\forall g\in\Xi\} \tag{5.94}$$
$$\{\Delta g\in\mathbb{R}^m\,|\bar{g}^j+\Delta g\in\mathcal{W}_\delta^j\}\subseteq\{\Delta g\in\mathbb{R}^m\,|g+\Delta g\in\mathcal{W}_\delta^j,\forall g\in\Xi\} \tag{5.95}$$

Let $g'\neq\bar{g}^j$ be a point in Ξ. Because (5.68) one has

$$\rho_{\bar{g}^j}^j\leq\rho_{g'}^j \tag{5.96}$$

where ρ_g^j is computed as in (5.66).

Now, let us focus first on (5.94) and let assume by contradiction that $\Delta\mathcal{G}^j(g')\subseteq\Delta\mathcal{G}^j(\bar{g}^j)$. This would imply, by exploiting (5.97), that

$$\|H_c\Phi^k(I-\Phi)^{-1}G\Delta g\|\leq\rho_{g'+\Delta g}^j-\gamma\rho_{g'}^j\leq\rho_{\bar{g}^j+\Delta g}^j-\gamma\rho_{\bar{g}^j}^j. \tag{5.97}$$

According to (5.97), one finds

$$\rho_{g'}^j+\rho_{\Delta g}^j-\gamma\rho_{g'}^j\leq\rho_{\bar{g}^j}+\rho_{\Delta g}^j-\gamma\rho_{\bar{g}^j}^j \tag{5.98}$$

that becomes

$$(1-\gamma)\rho_{g'}^j\leq(1-\gamma)\rho_{\bar{g}^j}^j. \tag{5.99}$$

The latter leads to the following expression

$$\rho_{g'}^j\leq\rho_{\bar{g}^j}^j \tag{5.100}$$

that is incompatible with (5.96). Then, because $\Delta\mathcal{G}^j(\bar{g}^j) \subseteq \Delta\mathcal{G}^j(g')$, for all $g' \in \Xi$, (5.94) is proved.

In order to prove (5.95), let consider g' as before and let investigate the structure of these two sets

$$\{\Delta g \in \mathbb{R}^m \,|\bar{g}^j + \Delta g \in \mathcal{W}_\delta^j\} = \{\Delta g \in \mathbb{R}^m \,|T^j H_c(I_n - \Phi)^{-1} G\Delta g \leq q_j - T^j H_c(I_n - \Phi)^{-1} G\bar{g}^j\} \tag{5.101}$$

and

$$\{\Delta g \in \mathbb{R}^m \,|g' + \Delta g \in \mathcal{W}_\delta^j\} = \{\Delta g \in \mathbb{R}^m \,|T^j H_c(I_n - \Phi)^{-1} G\Delta g \leq q_j - T^j H_c(I_n - \Phi)^{-1} Gg'\}. \tag{5.102}$$

Inequality (5.96) suggests that the right sides in (5.101) and (5.102) are related in the following way

$$q_j - T^j H_c(I_n - \Phi)^{-1} G\bar{g}^j \leq q_j - T^j H_c(I_n - \Phi)^{-1} Gg', \tag{5.103}$$

that means that all Δg belonging to $\{\Delta g \in \mathbb{R}^m \,|\bar{g}^j + \Delta g \in \mathcal{W}_\delta^j\}$ are in $\{\Delta g \in \mathbb{R}^m \,|g' + \Delta g \in \mathcal{W}_\delta^j\}$ or equivalently

$$\{\Delta g \in \mathbb{R}^m \,|\bar{g}^j + \Delta g \in \mathcal{W}_\delta^j\} \subseteq \{\Delta g \in \mathbb{R}^m \,|g' + \Delta g \in \mathcal{W}_\delta^j\} \tag{5.104}$$

that implies (5.95).

Lemma 5.32. *The set of constraints (5.81), (5.82) and (5.83) is limited, i.e.* $\forall [d^T, (d^+)^T]^T \neq 0 \; \exists \lambda > 0$ *and* $j \in \{1, \ldots, z\}$ *such that* $\tilde{a}_j^T(\underline{\Delta g} + \lambda d) + (\tilde{a}_j^+)^T(\Delta g^+ + \lambda d^+) > b_j$.

Proof

Let focus on constraints (5.82)-(5.83). Because each direction $\left[d^T, (d^+)^T\right]^T$ has to be chosen admissible, we have that the following conditions are true

$$\underline{\Delta g} + \lambda d > 0 \tag{5.105}$$

$$\underline{\Delta g} + \lambda d + \Delta g^+ + \lambda d^+ > 0. \tag{5.106}$$

Such inequalities suggest that $\Delta g^+ + \lambda d^+$ is always positive, i.e.

$$\lambda d + \Delta g^+ + \lambda d^+ \geq 0. \tag{5.107}$$

The rest of the proof is focused on verifying if for each direction $\left[d^T, (d^+)^T\right]^T$ there exists a $\lambda > 0$ that yields false one or more inequalities in (5.81). To this end, we have to consider two cases

1. d is a non parallel direction, i.e. $a_j^T d \neq 0, \forall j = 1, \ldots 3z$. Among equations (5.71)-(6.71) it is possible to observe that the box structure of $\mathcal{C}$ produces particular symmetries in the matrices $\tilde{A}$ and $\tilde{b}$ in (5.77). In particular, the first half of the lines of $\tilde{A}$ are opposite to the second half, i.e.

$$\tilde{a}_j^T = -\tilde{a}_{j+3z/2}^T, \quad j = 1, ..., 3z/2 \tag{5.108}$$

and the first half of components of $\tilde{b}$ equals the second half, i. e.

$$b_j^T = b_{j+3z/2}^T, \quad j = 1, ..., 3z/2. \tag{5.109}$$

Hence, we can consider the generic couple $(j, j + 3z/2)$ of lines of $\tilde{A}$. The related inequalities in (5.81) evaluated in the point $\left[\underline{\Delta g} + d, \Delta g^+ + \lambda d^+\right]$ are given by

$$\tilde{a}_j^T(\underline{\Delta g} + \lambda d) + (\tilde{a}_j^+)^T(\Delta g^+ + \lambda d^+) \leq b_j \tag{5.110}$$

$$-\tilde{a}_j^T(\underline{\Delta g} + \lambda d) + (\tilde{a}_{j+3z/2}^+)^T(\Delta g^+ + \lambda d^+) \leq b_{j+3z/2} = b_j, \tag{5.111}$$

Because (5.107), the terms $(\tilde{a}_j^+)^T(\Delta g^+ + d^+)$ and $\tilde{a}_{j+3z/2}^+(\Delta g^+ + d^+)$ are always positive so their substraction reinforces the above inequalities as follows

$$\tilde{a}_j^T(\underline{\Delta g} + \lambda d) \leq b_j \tag{5.112}$$

$$-\tilde{a}_j^T(\underline{\Delta g} + \lambda d) \leq b_j. \tag{5.113}$$

Then, for all $[d^T, (d^+)^T]^T \in \mathbb{R}^{2m}$, one can easily choose a sufficiently large λ such that at least one of the inequalities (5.112)-(5.113) fails.

2. d is a parallel direction, i.e $a_{j'}^T d = 0$ for some j'. Let consider the couple $(j', j' + 3z/2)$ of lines in $\tilde{A}$. The related inequalities in (5.81) evaluated at point $\left[(\underline{\Delta g} + \lambda d)^T, (\Delta g^+ + \lambda d^+)^T\right]^T$ are

$$\tilde{a}_{j'}^T(\underline{\Delta g} + \lambda d) + (\tilde{a}_{j'}^+)^T(\Delta g^+ + \lambda d^+) \leq b_{j'} \tag{5.114}$$

$$-\tilde{a}_{j'}^T(\underline{\Delta g} + \lambda d) + (\tilde{a}_{j'+3z/2}^+)^T(\Delta g^+ + \lambda d^+) \leq b_{j'}. \tag{5.115}$$

Because (5.107), the terms $(\tilde{a}_{j'}^+)^T(\Delta g^+ + \lambda d^+)$ and $(\tilde{a}_{j'+3z/2}^+)^T(\Delta g^+ + \lambda d^+)$ are always positive. Then (5.114) and (5.115) reduces to

$$\tilde{a}_{j'}^T(\underline{\Delta g} + \lambda d) \leq b_{j'} \tag{5.116}$$

$$-\tilde{a}_{j'}^T(\underline{\Delta g} + \lambda d) \leq b_{j'}. \tag{5.117}$$

that can be violated (at least one of them) by means of a sufficiently large value of λ for all $[d^T, (d^+)^T]^T \in \mathbb{R}^{2m}$.

6

Non-Iterative CG Distributed Schemes

The problem of interest in this Chapter is to extend the distributed supervision and coordination FF-CG strategies determined in the previous Chapter to the more general class of state-feedback Command Governor (CG) schemes recalled in Chapter 2 in the presence of bounded persistent disturbances.

The FF-CG solutions described in Chapter 3 are mainly characterized by the fact that their actions computation does not consist on the current measure or estimate of the state. The modified references are generated under the constraint of constantly applying a single FF-CG command for several sampling steps, so as to enforce the system evolutions to stay close to the set of feasible steady-state equilibria. Although this peculiarity of the FF-CG schemes make them an attractive solution for distributed frameworks their tracking and coordination performance are sub-optimal when fast-varying reference signals are of interest and especially when bounded persistent disturbances are present.

For this reasons the distributed approaches discussed here differ from those presented in Chapter 5 because here the state is assumed to be available (with some time-delay due to network latency) at the distributed master agents (see Figure 1.5) and is assumed to be available for computation.

Sequential (S-CG) and parallel (P-CG) distributed schemes inspired to S-FFCG and P-FFCG are presented and their stability, feasibility and viability (liveness) properties fully investigated. It is important to remark that the introduction of the state in the play introduces many new technical challenges for the development of distributed schemes which cannot overlooked and deserve carefully analysis. In this respect, this Chapter extends and makes clear several theoretical aspects of these sequential and parallel distributed schemes, not directly derivable from Chapter 5.

The Chapter is organized as follows: in Section 6.1 the system under consideration is described and the design problem formulated. In Section 6.2 the S-CG sequential strategy is fully described and analyzed. In Section 6.3 the parallel distributed version of the FF-CG is described. Section 6.4 concludes the Chapter.

6.1 System Description and Problem Formulation

Let us consider a set of N subsystems $\mathcal{A} = \{1, \ldots, N\}$, each one being a LTI closed-loop dynamical system regulated by a local controller which ensures stability and good closed-loop performance in linear regimes when the constraints are not active (small-signal regimes when the coordination is effective). Let the i-th closed-loop subsystem be described by the following discrete-time model

$$\begin{cases} x_i(t+1) = \Phi_{ii}x_i(t) + G_i g_i(t) + G_i^d d_i(t) + \sum_{j \in \mathcal{A}-\{i\}} \Phi_{ij} x_j(t) \\ y_i(t) = H_i^y x_i(t) \\ c_i(t) = H_i^c x(t) + L_i g(t) + L_i^d d(t) \end{cases} \tag{6.1}$$

where: $t \in \mathbb{Z}_+$, $x_i \in \mathbb{R}^{n_i}$ is the state vector (which includes the controller states under dynamic regulation), $g_i(t) \in \mathbb{R}^{m_i}$ the CG action, which, if no constraints were present, would essentially coincide with the reference $r_i(t) \in \mathbb{R}^{m_i}$. The vector $d_i(t) \in \mathbb{R}^{n_{d_i}}$ is an exogenous bounded disturbance satisfying $d_i(t) \in \mathcal{D}_i$, $\forall t \in \mathbb{Z}_+$ with $\mathcal{D}_i$ a specified convex and compact set such that $0_{n_{d_i}} \in \mathcal{D}_i$; $y_i(t) \in \mathbb{R}^m$ the output, viz. a performance related signal. Finally, $c_i \in \mathbb{R}^{n_i^c}$ represents the local constrained vector which has to fulfill the set-membership constraint

$$c_i(t) \in \mathcal{C}_i, \ \forall t \in \mathbb{Z}_+, \tag{6.2}$$

$\mathcal{C}^i$ being a convex and compact polytopic set. It is worth pointing out that, in order to possibly characterize global (coupling) constraints amongst states of different subsystems, the vector c_i in (5.1) is allowed to depend on the aggregate state and manipulable reference vectors $x = [x_1^T, \ldots, x_N^T]^T \in \mathbb{R}^n$, with $n = \sum_{i=1}^N n_i$, and $g = [g_1^T, \ldots, g_N^T]^T \in \mathbb{R}^m$, with $m = \sum_{i=1}^N m_i$, $d = [d_1^T, \ldots, d_N^T]^T \in \mathbb{R}^{n_d}$, with $n_d = \sum_{i=1}^N n_{d_i}$. Moreover, we denote by $r = [r_1^T, \ldots, r_N^T]^T \in \mathbb{R}^m$, $y = [y_1^T, \ldots, y_N^T]^T \in \mathbb{R}^m$ and $c = [c_1^T, \ldots, c_N^T]^T \in \mathbb{R}^{n^c}$, with $n^c = \sum_{i=1}^N n_i^c$, the other relevant aggregate vectors. The overall system arising by the composition of the above N subsystems is the system (2.1) already seen in Chapter 2

with

$$\Phi = \begin{pmatrix} \Phi_{11} & \ldots & \Phi_{1N} \\ \vdots & \ddots & \vdots \\ \Phi_{N1} & \ldots & \Phi_{NN} \end{pmatrix}, G = \begin{pmatrix} G_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & G_N \end{pmatrix}, G_d = \begin{pmatrix} G_1^d & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & G_N^d \end{pmatrix}$$

$$H^y = \begin{pmatrix} H_1^y & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & H_N^y \end{pmatrix}, H^c = \begin{pmatrix} H_1^c \\ \ldots \\ H_N^c \end{pmatrix}, L = \begin{pmatrix} L_1 \\ \ldots \\ L_N \end{pmatrix}, L_d = \begin{pmatrix} L_1^d \\ \ldots \\ L_N^d \end{pmatrix}.$$

As in the previous chapter the CG design problem is that of locally determining, at each time instant t and for each master agent $i \in \mathcal{A}$ associated

to each subsystem, a suitable reference signal $g_i(t)$ which is the best feasible approximation of $r_i(t)$ and such that its application never produces constraint violations, *i.e.* $c_i(t) \in \mathcal{C}^i, \forall t \in \mathbb{Z}_+, i \in \mathcal{A}$.

6.2 Distributed Sequential CG (S-CG)

Here we introduce a distributed CG scheme that is based on the centralized solution described in Chapter 2. According to this perspective the overall CG action $g(t)$ computed in a distributed way is admissible w.r.t. $c(t) \in \mathcal{C}$, where $\mathcal{C} \subseteq \{\mathcal{C}_1 \times ... \times \mathcal{C}_N\}$ is the global admissible region. In particular such a command is required to belong to the following convex and closed set of admissible virtual sequences

$$\mathcal{V}(x, \bar{k}) = \left\{ g \in \mathcal{W}_\delta : \bar{c}(k, x, g) \in \mathcal{C}_{k+\bar{k}}, \forall k \in \mathbb{Z}_+ \right\} \tag{6.3}$$

for a given $\bar{k} \geq 0$ $\mathcal{C}_{k+\bar{k}}$ is defined in (2.5). Note when $\bar{k} = 0$, one recovers to the standard definition (2.14). As it will be clarified hereinafter, the case $\bar{k} > 0$ needs to be considered to handle situations where time-delays and/or communication latency occur (see [58] for details).

We assume the agents (the Master nodes in Fig. 1.5) connected via a communication network. Such a network is modeled by the graph $\Gamma = (\mathcal{A}, \mathcal{B})$, defined in (5.1) with the related Hamiltonian cycle $\mathcal{H} = \{1, ...N\}$ Also in this case we will assume that each agent acts as a gateway in redistributing data amongst the other, no directly connected, agents. Then, at each time instant t, each i-th agent are aware of the following vectors:

$$\begin{array}{l} \xi_i(t) = [g_1^T(t - d_{i,1}), \ldots, g_i^T(t-1), \ldots, g_N^T(t - d_{i,N})]^T \\ \vartheta_i(t) = [x_1^T(t - d_{i,1}), \ldots, x_i^T(t), \ldots, x_N^T(t - d_{i,N})]^T \end{array} \tag{6.4}$$

As a consequence, the most recent common information regarding the measurement of the overall state available to each agent is $x(t - d_i)$ where $d_i = max_{j \in \mathcal{A}} d_{i,j}$.

The idea behind the approach is the same as in the S-FFCG approach and consists in allowing only one agent at time to manipulate its local command signal $g_i(t)$ while all others are instructed to keep applying their commands. After a new CG computation, the agent in charge transmits its local command and state to the next updating agent that is necessarily a neighbor. Such a polling policy implies that, eventually after a preliminary initialization cycle, at each time instant t, the following information are available to the generic i-th "agent in charge"

- the history of the aggregate vectors applied in the last N steps $g(t - N + j), j = \{1, \ldots, N-1\}$
- the measurement of the state at time $t - d_i$, i.e. $x(t - d_i)$

By exploiting this information, it is possible for it to compute the estimation $\hat{x}(t)$ of the current free-disturbance state at time t by means of the following recursions

$$\begin{aligned} &\hat{x}(t-d_i) = x(t-d_i) \\ &\hat{x}(k+1) = \Phi\hat{x}(k) + Gg(k), k = t-d_i, ..., t \end{aligned} \tag{6.5}$$

Then, by setting the parameter $\bar{k}$ in (6.3) equal to the time-delay d_i, we can formulate the following distributed S-CG algorithm:

Algorithm 6.2.1 ***Sequential-FFCG Algorithm (S-FFCG)*** - *Agent* i AT EACH TIME t

1.1 RECEIVE $\xi_j(t)$ AND $\vartheta_j(t)$ FROM EACH AGENT j IN $\mathcal{N}_i$
1.2 IF$(t \bmod N) == i$
  1.2.1 SOLVE

$$\begin{aligned} &g_i(t) = \arg\min_{(g_i)} \parallel g_i - r_i(t) \parallel^2_{\Psi_i} \\ &subject\ to: \\ &[g_1^T(t-1), ..., g_i^T, ..., g_N^T(t-1)]^T \in \mathcal{V}(\hat{x}(t), d_i) \end{aligned} \tag{6.6}$$

  1.2.2 APPLY $g_i(t)$
  1.2.3 UPDATE $g(t) = [g_1^T(t-1), ..., g_i^T(t), ..., g_N^T(t-1)]^T$
1.3 ELSE
  1.3.1 APPLY $g_i(t) = g_i(t-1)$
1.4 TRANSMIT $\xi_i(t)$ AND $\vartheta_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
where $\Psi_i > 0$ *are weighting matrices,* $t \bmod N$ *is the remainder of the integer division* t/N.

The properties of the above algorithm that we are going to enunciate are related to definitions, notations and assumptions given in the previous chapter and here not repeated for brevity. Furthermore, the following preliminary Lemma is needed to prove relevant properties of the scheme:

Lemma 6.1. *Let* **A3** *hold true. Then, for any perturbation* $\tilde{x}$ *satisfying* $\|\tilde{x}\| < \varrho, \varrho > 0$, *a point* $g' \in \mathcal{V}(x'_g + \tilde{x}, \bar{k})$ *is always viable,* $\forall \bar{k} > 0$

Proof - First it is useful to rephrase the set $\mathcal{V}(x'_g + \tilde{x}, \bar{k})$ into the following expression

$$\mathcal{V}(x_{g'} + \tilde{x}, \bar{k}) = \mathcal{W}_\delta \cap \mathcal{G}(x_{g'} + \tilde{x}, \bar{k}) \tag{6.7}$$

where

$$\mathcal{G}(x_{g'} + \tilde{x}, \bar{k}) := \{g \in \mathbb{R}^m : c_{g'} + H_c\Phi^k(x_{g'} + \tilde{x} - x_g) \in \mathcal{C}_{k+\bar{k}}, k \geq 0\} \tag{6.8}$$

and $\mathcal{C}_k$ is defined in (2.5). Because the set-membership of g to $\mathcal{V}(x_{g'} + \tilde{x}, \bar{k})$ requires that $g \in \mathcal{W}_\delta$, one is ensured that $c_g \in \mathcal{C}_\infty \sim \mathcal{B}^\delta$. Then

$$H_c\Phi^k(x_{g'} + \tilde{x} - x_g) \in \mathcal{C}_{k+\bar{k}} \sim (\mathcal{C} \sim \mathcal{B}^\delta) \Rightarrow g \in \mathcal{G}(x'_g + \tilde{x}, \bar{k}), \forall k \geq 0. \tag{6.9}$$

Moreover, because (2.5), one has

$$\mathcal{C}_{k+\bar{k}} \supseteq \mathcal{C}_\infty \supset (\mathcal{C}_\infty \sim \mathcal{B}^\delta) \tag{6.10}$$

that ensures that the following stronger sufficient condition for the set-membership of g to $\mathcal{G}(x'_g + \tilde{x}, \bar{k})$ holds true

$$H_c \Phi^k (x_{g'} + \tilde{x} - x_g) \in \mathcal{B}^\delta \Rightarrow g \in \mathcal{G}(x_{g'} + \tilde{x}, \bar{k}), \forall k \geq 0 \tag{6.11}$$

or equivalently

$$\|H_c \Phi^k (x_{g'} + \tilde{x} - x_g)\| \leq \delta \Rightarrow g \in \mathcal{G}(x_{g'} + \tilde{x}, \bar{k}), \forall k \geq 0. \tag{6.12}$$

The latter condition can be further enforced in the following way

$$\bar{\sigma}(H_c) M \|(x_{g'} + \tilde{x} - x_g)\| \leq \delta \Rightarrow g \in \mathcal{G}(x_{g'} + \tilde{x}, \bar{k}) \tag{6.13}$$

where $\|\Phi^k x\| \leq M \lambda^k \|x\|$ and $\bar{\sigma}(H_c)$ denotes the maximum singular value of H_c. By recalling the well known property of the norms $\|x + y\| \leq \|x\| + \|y\|$, simple algebraic manipulations change (6.13) into the following stronger condition

$$\|x_{g'} - x_g\| \leq \frac{\delta}{\bar{\sigma}(H_c) M} - \|\tilde{x}\| \Rightarrow g \in \mathcal{G}(x_{g'} + \tilde{x}, \bar{k}) \tag{6.14}$$

which, because of $x_g = H_c (I_n - \Phi)^{-1} G g$, becomes

$$\|H_c (I_n - \Phi)^{-1} (g' - g)\| \leq \frac{\delta}{\bar{\sigma}(H_c) M} - \|\tilde{x}\| \Rightarrow g \in \mathcal{G}(x'_g + \tilde{x}, \bar{k}). \tag{6.15}$$

Next, let $\varrho = \frac{\delta}{\bar{\sigma}(H_c) M}$. Then (6.15) implies that for all $\tilde{x}$ such that $\|\tilde{x}\| < \varrho$ there exists a ball centered at g' with radius $\eta > \frac{\varrho - \|\tilde{x}\|}{\bar{\sigma}((I_n - \Phi)^{-1} G)}$ entirely included in $\mathcal{G}(x_{g'} + \tilde{x}, \bar{k})$. As a consequence, the set $\mathcal{V}(x_{g'} + \tilde{x}, \bar{k})$ contains $\mathcal{W}_\delta \cap \mathcal{B}^\eta_{g'}$, i.e.

$$\mathcal{V}(x_{g'} + \tilde{x}, \bar{k}) \supseteq \mathcal{W}_\delta \cap \mathcal{B}^\eta_{g'}. \tag{6.16}$$

Hence, each decision set for g' related to the generic i-th agent is representable as $\mathcal{V}_i^{\mathcal{W}_\delta \cap \mathcal{B}^\eta_{g'}}(g')$. This reduces to $\mathcal{V}_i^{\mathcal{W}_\delta \cap \mathcal{B}^\eta_{g'}}(g') = \mathcal{V}_i^{\mathcal{W}_\delta}(g')$ because inside $\mathcal{B}^\eta_{g'}(g')$ all directions $d \in \mathbb{R}^m$ related to g' are admissible. Then, the enjoinment of the viability for g' depends only on its set-membership to $\mathcal{W}_\delta$ and it is ensured by **A3**.

Finally, the following properties can be shown to hold under **A3** for the above stated S-CG scheme

Theorem 6.2. *Let assumptions* **A1-A2-A3** *be fulfilled for systems arising from the composition of N subsystems in form (6.1). Let consider the distributed* ***S-CG*** *selection rule (6.6) and let $\mathcal{V}(x(t), d_{max})$ be non empty at time $t = 0$, where $d_{max} = \max_j d_j$. Then*

1) for each agent $i \in \mathcal{A}$, at each time t, the minimizer in (6.6) uniquely exists and can be obtained by locally solving a convex constrained optimization problem;
2) the overall system acted by the agents implementing the S-CG policy never violates the constraints, i.e. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$;
3) whenever $r(t) \equiv [\ r_1^T, \ldots, r_N^T\ ]^T, \forall t$, with r_i a constant set-point, the sequence of solutions $g(t) = [g_1^T(t), \ldots, g_N^T(t)]^T$ asymptotically converges to a Pareto-Optimal stationary (constant) solution of (5.9), which is given by r whenever $r \in \mathcal{W}_\delta$, or by any other Pareto-Optimal solution $\hat{r} \in \mathcal{W}_\delta$ otherwise.

Proof

1) The existence of an admissible solution for each agent at each time t can be proved by simply remarking that $g_i(t) = g_i(t-1)$ is always an admissible, although not necessarily the optimal, solution for the prescribed problem at time t.
2) At each time t, from a centralized point of view, a command $g(t)$ belonging to $\mathcal{V}(\hat{x}(t), d_i), i \in \mathcal{A}$ is applied to the overall plant. By construction, the latter set is equivalently characterizable as

$$\mathcal{K}(\hat{x}(t), d_i) := \Big\{ g \in \mathcal{W}_\delta : \bar{c}(k, \hat{x}(t), g) \oplus \bigoplus_{j=1}^{d_i} H_c \Phi^{j-1} G_d \mathcal{D} \oplus L_d \mathcal{D} \subset \mathcal{C}_k,\ \forall k \in \mathbb{Z}_+ \Big\}$$
$$= \big\{ g \in \mathcal{W}_\delta : \bar{c}(k, x, g) \in \mathcal{C}_k, \forall k \in \mathbb{Z}_+, \forall x \in \hat{x}(t) \oplus \bigoplus_{j=1}^{d_i} \Phi^{j-1} G_d \mathcal{D} \oplus L_d \mathcal{D} \big\}. \tag{6.17}$$

where $\oplus$ denotes the Pontryagin set sum according to Definition (2.2). Moreover, at each time t the state $\bar{x}(t)$, estimated by each acting i-th agent, satisfies

$$\bar{x}(t) = x(t) - \sum_{j=1}^{d_i} \Phi^{j-1} G_d d(i) + L_d d(t) \tag{6.18}$$

that ensures

$$x(t) \in \hat{x}(t) \oplus \bigoplus_{j=1}^{d_i} \Phi^{j-1} G_d \mathcal{D} + L_d \mathcal{D}. \tag{6.19}$$

The latter implies that at each time instant a command $g(t) \in \mathcal{V}(\bar{x}(t), d_i) \subseteq \mathcal{V}(x(t), 0)$ is selected, and hence, by construction, this is sufficient to guarantee that $c(t) \in \mathcal{C}, \forall t \in \mathbb{Z}_+$.
3) The stated convergence property follows simply because the sequences of solutions $g_i(t)$ makes the sequences of local costs $||g_i(t) - r_i||^2_{\Psi_i}$ non increasing for any $i = 1, ..., N$ under constant set-points. In fact, it is not convenient for the agents to modify their actual optimal solutions if the

costs cannot be decreased further on. To this end, let $g_i(t)$ be the S-CG action of the i-th agent at time t, solution of the optimization problem (5.7). As already discussed, $g_i(t)$ is still an admissible, though not necessarily the optimal, solution at time $t+1$. Hence, the sequences of costs $||g_i(t) - r_i||^2_{\Psi_i}$ are all non-increasing, i.e.

$$\| g_i(t+1) - r_i \|^2_{\Psi_i} \leq \| g_i(t) - r_i \|^2_{\Psi_i} \tag{6.20}$$

Then, we want to show that any stationary optimal solution, viz. $g(t) = g(t+1)\ \forall t$, is Pareto Optimal by proving that a solution is not stationary if not Pareto-Optimal. To this end, let $g'(t) = [g_1'^T(t), ..., g_N'^T(t)]^T$ be the actual solution at time $t \in \mathbb{Z}_+$ which is assumed to be not Pareto-Optimal. As a consequence, other different solutions exist which improve the costs. Supposedly, vectors $v = [v_1^T, ..., v_N^T]^T \in \mathbb{R}^m$ would exist with $g'(t) + v \in \mathcal{W}_\delta$, such that

$$||g_i'(t) + v_i - r_i||^2_{\Psi_i} - ||g_i'(t) - r_i||^2_{\Psi_i} \leq 0, \tag{6.21}$$

happens to hold for all $i \in \mathcal{A}' := \{i \in \mathcal{A} : v_i \neq 0\}$ with some of the above inequalities becoming strict for at least one index $i \in \mathcal{A}'$. Because of the strict convexity of the norm $||\cdot||^2_{\Psi_i}$, the following inequality happens to be true for all $\alpha \in (0,1)$

$$\begin{aligned} ||(1-\alpha)g_i' + \alpha(g_i'(t) + v_i) - r_i||^2_{\Psi_i} \\ < (1-\alpha)||g_i'(t) - r_i||^2_{\Psi_i} + \alpha||g_i'(t) + v_i - r_i||^2_{\Psi_i} \end{aligned} \tag{6.22}$$

Therefore, by means of straightforward algebraic manipulations, one arrives to

$$\begin{aligned} ||g_i'(t) + \alpha v_i - r_i||^2_{\Psi_i} - ||g_i'(t) - r_i||^2_{\Psi_i} \\ < \alpha(||g_i'(t) + v_i - r_i||^2_{\Psi_i} - ||g_i'(t) - r_i||^2_{\Psi_i}) \end{aligned} \tag{6.23}$$

for all $\alpha \in (0,1)$. Because (6.21), the right-hand term in (6.23) is always negative. Then, one can state

$$||g_i'(t) + \alpha v_i - r_i||^2_{\Psi_i} - ||g_i'(t) - r_i||^2_{\Psi_i} < 0, \forall \alpha \in (0,1) \tag{6.24}$$

The latter may be interpreted as the fact that if the above admissible direction v did exist at $g'(t)$, **for each** agent $i \in \mathcal{A}'$ it would be strictly convenient to move to $g_i'(t) + \alpha v_i$, for a suitable value of α, from its previous solution $g_i'(t)$.

Now we have to verify that at least one agent is allowed to move from $g_i'(t)$ along v_i because of constraints. To this end, we have to note that if $g(t) \equiv g\ \forall t$, since **A1** for each $\varrho > 0$ it exists a finite t_s such that

$$\tilde{x} = \|\hat{x}(t) - x_g\| \leq \varrho, \forall t \geq t_s, \forall g \in \mathcal{W}_\delta, \tag{6.25}$$

as consequence we can be sure that after a finite t_s,

$$\mathcal{V}(\hat{x}(t), \bar{k}) \supseteq \mathcal{V}(x_{g'(t)} + \tilde{x}, \bar{k}), \forall t \geq t_s, \forall \tilde{x} \leq \varrho \tag{6.26}$$

with $\varrho = \frac{\delta}{H_c M}$. Then, because of Lemma 6.1, $g'(t), \forall t \geq t_s$ will be viable or equivalently v_i will belong to $\mathcal{V}_i^{\mathcal{W}_\delta}(g'(t)) \forall t \geq t_s$ for all agents corresponding to any not empty subset $\mathcal{A}'_v$. Hence, according to the sequential S-CG updating policy, if at time t, the index $(t \bmod N) \in \mathcal{A}'_v$ then, because of (6.24), the agent $i' = t \bmod N$ will find convenient to move into $g'_{i'}(t) + \alpha v_{i'}, \alpha \in [0, \bar{\alpha}]$. In fact, because of the viability of $g'(t)$ (see Definition 5.4) $v_{i'} \in \mathcal{V}_{i'}^{\mathcal{W}_\delta}(g'(t))$ implies that a scalar $\bar{\alpha} \in (0, 1)$ exists ensuring $g' + [0_1^T, ..., \alpha v_i^T, ..., 0_N^T]^T \in \mathcal{W}_\delta$ for all $\alpha \in (0, \bar{\alpha})$.
When no agents in $\mathcal{A}'_v$ are allowed to update their actions at t, all of them constantly apply the most recently applied commands until one of them becomes the allowed agent. In fact, the condition $\mathcal{A}'_v \subseteq \mathcal{A}$ ensures that a future time $t' = (t + j')$, $j' \in [1, N]$, surely will exist for the agent $((t+j') \bmod N) \in \mathcal{A}'_v$. Please notice also that $\mathcal{A}'_v$ does not change because $g((t+j)) = g'(t)$, for all $j \in [1, j']$. Thus, if we are not at a Pareto-optimal solution at time t, at least one of the agents will move from it by N time steps.

6.2.1 Computational Details

Here we give a fully implementable formulation of the Problem (6.6) when $\mathcal{C}$ consists of polyhedral constraints (see (2.18)-(2.21)). To this end, it suffices to resort to the CG centralized problem (2.52) when the decision variable is not any longer the entire command vector $g \in \mathbb{R}^m$ but a subvector $g_i \in \mathbb{R}^{m_i}$ of it related to agent i. Then, Problem (6.6) is rephrased, by means of the procedure (2.34)-(2.51) given in Section 2.3.2, as

$$g_i(t) = \min_{g_i} (g_i - r_i(t))' \Psi_i (g_i - r_i(t))$$

$$\text{subject to} \tag{6.27}$$

$$T H_c \Phi^k \hat{x}(t) + T R_k^c [g_1^T(t-1), ..., g_i^T, ..., g_N^T(t-1)]^T \leq q^{(k+d_i)}, \quad k = 0, ..., k_0 \tag{6.28}$$

$$T(H_c(I - \Phi)^{-1} G + L)[g_1^T(t-1), ..., g_i^T, ..., g_N^T(t-1)]^T \leq q^{k_\varepsilon} - (\varepsilon + \delta)[\sqrt{T_j^T T_j}] \tag{6.29}$$

where $[\sqrt{T_j^T T_j}]$ is defined in (2.24), k_0 is determined according to Algorithm (2.3.2) and the constraints (6.28)-(6.29) characterize $\mathcal{V}(x(t), d_i)$.

6.2.2 Constraints Qualification and Viable Approximations

The S-CG strategy is subject to the same drawbacks of the S-FFCG strategy depicted in Figg. $5.2 - 5.3$. Fortunately, the same approximation described in

Section 5.3.1 can be used here in the case that a no viable polyhedron $\mathcal{W}_\delta$ arises. The resulting inner-approximating polyhedron $\mathcal{W}'_\delta$ is then used in the place of $\mathcal{W}_\delta$ and the S-CG problem (6.6) can be recast as follows

$$\begin{array}{l} g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel^2_{\Psi_i} \\ subject\ to: \\ \left\{g(t) = [g_1^T(t-1),..,g_i^T,...,g_N^T(t-1)]^T \in \mathcal{V}'(\bar{x}(t), d_i)\right. \end{array} \tag{6.30}$$

where

$$\mathcal{V}'(x, \bar{k}) := \left\{ g \in \mathcal{W}'_\delta : \bar{c}(k, x(t), g) \in \mathcal{C}_{k+\bar{k}},\ \forall k \in \mathbb{Z}_+. \right\} \tag{6.31}$$

Note that, because of Lemma 6.1 and item 3) of the proof of Theorem 6.2, the approximation procedure given in Section 5.3.1 involves the set $\mathcal{W}_\delta$ only.

6.3 Parallel Command Governor Scheme (P-CG)

We present here a more effective "parallel" distributed strategy where any agent is enabled to select its local command simultaneously at each time instant. To this end, we make use of ideas introduced in the P-FFCG approach presented in Section 5.4 by assuming, as done in Section 6.2, that the agents are connected via a communication network represented by a graph $\Gamma = (\mathcal{A}, \mathcal{B})$ defined in (5.1) with the corresponding Hamiltonian cycle $\mathcal{H} = \{1, ...N\}$. Such a communication structure allow the agents to act as gateways in redistributing at each time instant data amongst other, no directly connected, agents.

Then, at each time instant t, the most recent information on measured states and computed commands which the generic i-th agent has available locally about all other agents is represented by the already introduced **Local Information** vectors (6.4). Then, it results that the **Common Information** on the actual states and applied commands shared amongst all agents at each time t is given by the vector

$$\begin{array}{l} \xi(t) = \left[g_1^T(t-d_1), ...g_i(t-1), ..., g_N^T(t-d_N)\right]^T \\ \vartheta(t) = \left[x_1^T(t-d_1), ...x_i(t-1), ..., x_N^T(t-d_N)\right]^T \end{array} \tag{6.32}$$

where d_i is the maximum amongst all distances $d_{i,j}$ from the i-th agent to any other agent in the graph, i.e. $d_i = max_{j \in \mathcal{A}} d_{i,j}$. It results that the more recent measure of the overall state known by all agents at time t is given by

$$x(t - d_{max}) = \left[x_1^T(t-d_{max}), ..., x_N^T(t-d_{max})\right]^T \tag{6.33}$$

where $d_{max} = \max_j d_j$.

As in the P-FFCG scheme, the idea is that of generating, at each step and on the basis of the information shared by all agents of the network, a set of decoupled constraints (one for each agent) via the so-called *Set Cartesian*

Decomposition ([57]). These new constraints are generated in such a way that their local agent-wise fulfilment implies the fulfilment of the original global constraints (6.3). Unlike the P-FFCG approach, such constraints are determined on the basis of an estimation of the current state as well as the past CG actions computed by the agents.

As a result, the optimization problem decouples and each agent results simply to fulfil inclusions into local sets of the form

$$g_i(t) - g_i(t-1) \in \Delta\mathcal{V}_i(t), i = 1, ..., N \tag{6.34}$$

with $\Delta\mathcal{V}_i(t) \subseteq \mathbb{R}^{m_i}$, $i \in \mathcal{A}$, convex and compact sets containing 0_{m_i} for all $t \geq 0$ to be specified.

For the moment, we postpone the formal definition for such sets after the introduction of extra notations and we assume that each agent at time t is provided with a collection $\{\Delta\mathcal{V}_i(t-k)\}_{k=1}^{d_{max}} i \in \mathcal{A}$ of sets computed previously, where $d_{max} = \max_j d_j$. Such an information, together with the common information vector $\xi(t)$, can be exploited to define the set of all possible feasible values which $g(t)$ could have been assumed in the last d_{max} instants, computed as follows

$$\begin{array}{ll} \hat{g}_i(t-k|t) = g_i(t-k) & \text{if } k \geq d_i \\ \hat{g}_i(t-k|t) = g_i(t-d_i) \oplus \Delta\mathcal{V}_i(t-d_1+1) \oplus ... \oplus \Delta\mathcal{V}_i(t-k) & \text{if } k < d_i. \end{array} \tag{6.35}$$

Then, the set of all possible values for $g(t-k)$ is given by

$$\Xi(t-k|t) = \hat{g}_1(t-k|t) \times, ..., \times \hat{g}_N(t-k|t). \tag{6.36}$$

Moreover, the set of all the possible state predictions $\hat{x}(t)$ at time t, computed on the basis of the measured state $x(t-d_{max})$ available to all agents of the network is given by

$$\Omega(t) := \Phi^{d_{max}} x(t-d_{max}) + \bigoplus_{j=0}^{d_{max}-1} \Phi^{d_{max}-1-j} G \Xi(t-d_{max}+j|t) \tag{6.37}$$

Based on the above sets we may compute the set of admissible aggregate **command variations** as follows

$$\Delta\mathcal{V}(\Xi, \Omega, \bar{k}) := \left\{\Delta g : (g + \Delta g) \in \mathcal{V}(x, \bar{k}), \forall g \in \Xi, \forall x \in \Omega\right\} \tag{6.38}$$

Finally, the (approximated) Cartesian decomposition giving rise the agent-wise decoupled constraints (6.34) should satisfy the following set inclusion condition that can be determined independently and locally by all agents

$$\Delta\mathcal{V}_1(t) \times \ldots \times \Delta\mathcal{V}_N(t) \subseteq \Delta\mathcal{V}(\Xi(t-1|t), \Omega(t), d_{max}), \tag{6.39}$$

$$0_i \in In(\Delta\mathcal{V}_1(t)), \ldots, 0_N \in In(\Delta\mathcal{V}_N(t)) \tag{6.40}$$

Here we need also to consider four possible *operative scenarios* because, under certain conditions, the agents need to operate sequentially by adopting the S-CG strategy. The various scenarios can be characterized as follows:

- **PAR:** In this scenario agents go parallel and switch to the sequential scenario if the distance between $\Xi(t-1|t)$ and the border of $\mathcal{W}_\delta$ is smaller than a prefixed threshold ϵ_P. In particular:
  1.1 Each agent computes the minimum distance between $\Xi(t-1|t)$ and the border of $\mathcal{W}_\delta$ as

$$\mu(t) = \min_{g \in \Xi(t-1|t), g' \in \partial(\mathcal{W}_\delta)} \|g - g'\| \tag{6.41}$$

  2.1 If $\mu(t)$ is bigger than a predetermined threshold ϵ_P
    2.1.1 Each agent determines the collection of sets $\Delta\mathcal{V}_i(t), i = 1, ..., N$ as the solution of its instance of the following optimization problem

$$\begin{array}{c} \max\limits_{\Delta\mathcal{V}_i(t), i=1,...,N} V(\Delta\mathcal{V}_1(t) \times ... \times \Delta\mathcal{V}_N(t)) \\ subject \quad to \ (6.39), (6.40) \end{array} \tag{6.42}$$

    where $V(\cdot)$ denotes a possible measure of the volume of a set (to achieve good dynamical properties we want $(\Delta\mathcal{V}_1(t) \times ... \times \Delta\mathcal{V}_N(t))$ to be as large as possible)
    2.1.2 each agent chooses its own reference by solving the following convex optimization problem

$$\begin{array}{c} g_i(t) = arg \min\limits_{g_i} \|g_i - r_i(t)\|^2_{\Psi_i} \\ subject \quad to \ (g_i - g_i(t-1)) \in \Delta\mathcal{V}_i(t) \end{array} \tag{6.43}$$

  2.2 Otherwise, go to **P2S**
- **P2S:** This is a transition scenario between **PAR** and **SEQ**. In this case, only one agent at a time updates its action, according to its position in the Hamiltonian cycle $\mathcal{H}$, while all other non-active agents set their local set $\Delta\mathcal{V}_i = 0$. More formally:
  1.1 The agent in charge computes the following set on the basis of its most updated local information (6.4) and (6.36)

$$\Xi_i(t) = \{\xi_i(t)\} \oplus \Big(\bigoplus_{k=1}^{d_{i,1}} \Delta\mathcal{V}_1(t-k) \times \ldots \times \bigoplus_{k=1}^{d_{i,N}} \Delta\mathcal{V}_N(t-k) \Big) \tag{6.44}$$

  2.1 If Ξ_i is not a singleton, the agent computes first, on the basis of the current state $x(t-d_i)$, the set of all possible state predictions $\hat{x}(t)$ at time t

$$\Omega_i(t) := \Phi^{d_i} x(t-d_i) + \bigoplus_{j=0}^{d_i-1} \Phi^{d_i-1-j} G \Xi(t-d_i+j|t) \tag{6.45}$$

  Then, the value $g_i(t)$ is selected as

$$\begin{array}{l} g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel^2_{\Psi_i} \\ subject \quad to \ (g_i - g_i(t-1)) \in \Delta\mathcal{V}_i^0(\Xi(t-1|t), \Omega_i(t), d_i) \end{array} \tag{6.46}$$

where

$$\Delta\mathcal{V}_i^0(\Xi, \Omega_i) := \left\{ \Delta g_i \in \mathbb{R}^{m_i} \,|[0_1^T, ..., \Delta g_i^T, ..., 0_N^T]^T \in \Delta\mathcal{V}(\Xi, \Omega_i, d_i) \right\}. \tag{6.47}$$

2.2 Otherwise, go to scenario **SEQ**

- **SEQ:** In this case the agents go sequentially by using the S-CG strategy described in Section 6.2. They are instructed to come back to the **PAR** scenario when the distance between the global command $g(t-1)$ and the border of $\mathcal{W}_\delta$ is bigger than a prefixed threshold ϵ_S. When this event occurs, only the agent in charge is capable of catching it because it knows the actual applied global command $g(t-1)$. Then, before switching to **S2P**, it sends an acknowledgement to all other agents consisting of a number $\sigma_i(t) = d_{max}$ where $d_{max} = \max_j d_j$. More in details, the agent in charge:
  1.1 computes
  $$\mu(t) = \min_{g \in \partial(\mathcal{W}_\delta)} \|g - g(t-1)\| \tag{6.48}$$
  2.1 If $\mu(t)$ is smaller than ϵ_S computes $g_i(t)$ by means of (5.7).
  3.1 Otherwise it sets $\sigma_i(t) = d_{max}$, sends $\sigma_i(t)$ to neighbors and goes to **S2P**.

  On the contrary, each agent not in charge has to check if an acknowledgement $\sigma_j(t)$ has been received from one of their neighbors and, in that case, they set $\sigma_i(t) = \sigma_j(t) - 1$, communicate $\sigma_i(t)$ to their neighbors and goes to **S2P**.
- **S2P:** This is a transition scenario between **SEQ** and **PAR** that represents an initialization phase for the parallel procedure. In this case, all agents decrement $\sigma_i(t)$. When $\sigma_i(t) = 0$, all agents know what is the last applied command vector $g(t)$. Then, they can switch to **PAR** and the parallel procedure can restart with $\Xi(t-k|t) = \{g(t-k)\}, k > 0$. As consequence, in particular, the set $\Omega(t)$ also reduce to a singleton, i.e. $\Omega(t) = \{\hat{x}(t)\}$.

Remark 6.3. It is worth remarking that in the **P2S** scenario after d_{max} time steps the set $\Xi_i(t), \forall i \in \mathcal{A}$ reduces to a singleton. Hence, the transition towards the **SEQ** scenario is ensured.

The algorithm that we are going to present, depicted in Figure 5.4, has the same structure of the Algorithm 5.4.1. More formally, it consists of the following "parallel" version of distributed CG algorithm:

Algorithm 6.3.1 ***Parallel-CG Algorithm (P-CG)*** - *Agent i-th*
AT EACH TIME t
1.1 RECEIVE $\xi_j(t), \vartheta_j(t), \sigma_j(t)$ FROM ALL NEIGHBOURS $j \in \mathcal{N}_i$

1.2 GO TO $st_i(t)$

PAR:
1.1 SOLVE (6.41)
1.2 IF $\mu(t) \geq \epsilon_P$
  1.2.1 SOLVE (6.42)
  1.2.2 SOLVE (6.43)
1.3 ELSE
  1.3.1 SET $st_i(t) =$ "$P2S$"
  1.3.2 GO TO $st_i(t)$
1.4 UPDATE $\xi_i(t), \vartheta_i(t)$
1.5 TRANSMIT $\xi_i(t), \vartheta_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.6 APPLY $g_i(t)$
1.7 SET $st_i(t+1) = st_i(t)$

P2S:
1.1 COMPUTE $\Xi_i(t)$ AS IN (6.44)
1.2 IF $|\Xi_i(t)| > 1$
  1.2.1 IF$((t \bmod N) = i)$, SOLVE (6.46)
  1.2.2 ELSE, SET $g_i(t) = g_i(t-1)$
  1.2.3 UPDATE $\xi_i(t), \vartheta_i(t)$
  1.2.4 TRANSMIT $\xi_i(t), \vartheta_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
  1.2.5 APPLY $g_i(t)$
  1.2.6 SET $\Delta\mathcal{V}_i(t) = 0_{m_i}$
1.3 ELSE
  1.3.1 SET $st_i(t) =$ "SEQ"
  1.3.2 GO TO $st_i(t+1) = st_i(t)$

SEQ:
1.1 IF $\sigma_j(t-1) > -1, j \in \mathcal{N}_i$
  1.1.1 SET $\sigma_i(t) = \sigma_j(t) - 1$
  1.1.2 TRANSMIT $\sigma_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
  1.1.3 SET $st_i(t) =$ "$S2P$"
  1.1.4 GO TO $st_i(t)$
1.2 ELSE
  1.2.1 IF$((t \bmod N) = i)$
    1.2.1.1 SOLVE (6.48)
    1.2.1.2 IF $\mu(t) \leq \epsilon_S$,SOLVE(6.6)
    1.2.1.3 ELSE
      1.2.1.3.1 SET $\sigma_i(t) = d_{max}$
      1.2.1.3.2 TRANSMIT $\sigma_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
      1.2.1.3.3 SET $st_i(t) =$ "$S2P$"
      1.2.1.3.4 GO TO $st_i(t)$
  1.2.2 ELSE, SET $g_i(t) = g_i(t-1)$
  1.2.3 UPDATE $\xi_i(t), \vartheta_i(t)$

1.2.4 TRANSMIT $\xi_i(t), \vartheta_i(t)$ AND $\sigma_i(t) = -1$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.2.5 APPLY $g_i(t)$
1.2.6 SET $\Delta\mathcal{V}_i(t) = 0_{m_i}$
1.2.7 SET $st_i(t+1) = st_i(t)$

S2P:
1.1 IF $\sigma_i(t) = 0$
1.1.1 SET $st_i(t) =$ "PAR"
1.1.2 GO TO $st_i(t)$
1.2 ELSE
1.2.1 SET $\sigma_i(t+1) = \sigma(t) - 1$
1.2.2 TRANSMIT $\sigma_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.2.3 UPDATE $\xi_i(t), \vartheta_i(t)$
1.2.4 TRANSMIT $\xi_i(t), \vartheta_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$
1.2.5 APPLY $g_i(t) = g_i(t-1)$
1.2.6 SET $\Delta\mathcal{V}_i(t) = 0_{m_i}$
1.2.7 SET $st_i(t+1) = st_i(t)$

Finally, some properties enjoyed by the above P-CG scheme can be stated.

Theorem 6.4. *Let assumptions* **A1-A2-A3** *be fulfilled for the system arising from the composition of N subsystems (6.1). Consider the distributed* ***P-CG*** *Algorithm (6.3.1) and let $\mathcal{V}(x(t), d_{max})$ be non empty at time $t = 0$, where $d_{max} = \max_j d_j$. Then*

1. *for each agent $i \in \mathcal{A}$, at each time t, the minimizer related to one of the optimization problems (6.43), (6.46) and (6.6), to be solved depending on the value of $st_i, i \in \mathcal{A}$, uniquely exists and can be obtained by solving a convex constrained optimization problem;*
2. *The overall system acted by the agents implementing the P-CG supervisory policy never violates the constraints, i.e. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$.*
3. *Whenever $r_i(t) \equiv r_i$, $\forall i \in \mathcal{A}$, with r_i a constant set-point the sequence of $g(t) = [g_1^T(t), \ldots, g_N^T(t)]^T$ asymptotically converges either to $r = [r_1^T(t), \ldots, r_N^T(t)]^T$ if $r \in \mathcal{W}_\delta$ or to a point $\hat{r}$ that is Pareto-Optimal for the problem (5.9).*

Proof

1) The existence of an admissible solution for each agent at each time t can be proved by simply remarking that $g_i(t) = g_i(t-1), i = 1, \ldots, N$, is always an admissible, although not necessarily optimal, solution for the prescribed problems at time t, in any state st_i. In fact, the sets $\Delta\mathcal{V}(\Xi(t-1|t), \Omega(t), d_{max})$ and $\Delta\mathcal{V}(\Xi(t-1|t), \Omega_i(t), d_i), \forall i \in \mathcal{A}$ always contain the point 0_m (note that $\Xi(t)$ is contained in $\mathcal{W}_\delta$).
2) At each time t, from a centralized point of view, a command $g(t)$ belonging to $\mathcal{V}(\hat{x}(t), d_{max}), i \in \mathcal{A}$ is applied to the overall plant. The rest of the proof follows the same discussion made in item 2) of proof of Theorem 6.2.

3) The convergence follows simply because the sequences of solutions $g_i(t)$ are such that the local costs $||g_i(t) - r_i||^2_{\Psi_i}$ are non increasing for any $i = 1, ..., N$. In fact, there is not convenience for the agents to modify their actual optimal solutions if the costs could not be minimized further on. To this end, let $g_i(t)$ be the P-CG local action at time t, solution of any optimization problem among (6.43), (6.46), (6.6). As already seen in item 1) of this proof, at time $t+1$, $g_i(t)$ is still an admissible, though not necessarily the optimal, solution and hence the sequences $||g_i(t) - r_i||^2_{\Psi_i}$ are non increasing, i.e. $\| g_i(t+1) - r_i \|^2_{\Psi_i} \leq \| g_i(t) - r_i \|^2_{\Psi_i}$.
Then, we want to show that any stationary optimal solution, viz. $g(t) = g(t+1)\ \forall t$, is Pareto Optimal by proving that a solution is not stationary if is not Pareto Optimal. To this end, let $g' = [g_1'^T, ..., g_N'^T]^T$ be the actual solution at time $t \in \mathbb{Z}_+$ which is assumed to be not Pareto Optimal. As a consequence, other different solutions exist which improve the costs. Supposedly, vectors $v = [v_1^T, ..., v_N^T]^T \in \mathbb{R}^m$ would exist with $g' + v \in \mathcal{W}_\delta$, such that

$$||g_i' + v_i - r_i||^2_{\Psi_i} - ||g_i' - r_i||^2_{\Psi_i} \leq 0, \tag{6.49}$$

happens to hold for all $i \in \mathcal{A}' := \{i \in \mathcal{A} : v_i \neq 0\}$ with some of the above inequalities becoming strict for at least one index $i \in \mathcal{A}'$.
As already stated in item 3) of the proof of Theorem 5.7, the existence of a vector v satisfying (6.49) makes it convenient for all agents to move from g'. Hence we have to show that the policy underlying Algorithm 5.4.1 guarantees that at least one agent can modify its command g_i'. To this end, it is sufficient to prove that in scenario "PAR" and "SEQ" the cost will decrease for at least on agent. In fact "P2S" and "S2P" represent transient scenarios because the algorithm could dwell there at most d_{max} time steps. If at time t $st_i(t)$ ="SEQ",$\forall i \in \mathcal{A}$ and $\mu(t) \leq \epsilon_S$, where $\mu(t)$ is computed as in (6.48), then, the agents will go on sequentially and, since discussion of point 3) of the proof of Theorem 6.2 at least one agent can change its command. Conversely, if $\mu(t) > \epsilon_S$, agents will not move in "SEQ" but they will switch on "S2P" and subsequently, at a time t', such that $t' - t \leq d_{max}$, in "PAR" with $\Xi(t' - k|t') = \{g'\}, k > 0$ and $\Omega(t')$ reduced to a singleton. In such a situation $\mu(t')$ computed as in (6.41) will coincide with $\mu(t)$ in (6.48) and then it will be for sure bigger than ϵ_P because

$$\mu(t') = \mu(t) > \epsilon_S \geq \epsilon_P. \tag{6.50}$$

Moreover at t' the set of common admissible variations in (6.38) becomes

$$\begin{array}{l} \Delta\mathcal{V}(\Xi(t' - 1|t'), \Omega(t'), d_{max}) = \\ \Delta\mathcal{V}(g', \hat{x}(t'), d_{max}) = \{\Delta g : (g' + \Delta g) \in \mathcal{V}(x, d_{max})\}. \end{array} \tag{6.51}$$

If such a set does not contains a ball of finite radius centered at 0_m agents cannot perform decomposition (6.39) and the global command $g(t')$ will equals g'. Nevertheless, if $g(t) = g'$ is kept constant, as discussed in item 3) of proof of Theorem 6.2 after a finite time t_s the

set $\mathcal{V}(x, d_{max})$ will satisfies condition (6.26) w.r.t. to g' and, because of Lemma 6.1 it will contain a ball of finite radius centered at g' or equivalently $\Delta\mathcal{V}(\Xi(t'+t_s-1|t'+t_s), \Omega(t'+t_s), d_{max}) = \Delta\mathcal{V}(g(t'+t_s-1), \hat{x}(t'+t_s), d_{max})$ will contain a ball of finite radius centered at 0_m. As consequence the sets $\Delta\mathcal{V}_i(t'+t_s), \forall i \in \mathcal{A}$, derived by means of (6.42), will also contain a ball of finite radius centered at 0_{m_i} that allows each agent to move in any sub-direction $v_i \in \mathbb{R}^{m_i}$.
Similar considerations are valid when, at time t, $st_i(t) =$"PAR" and $\Xi(t)$ is not a singleton. In this case if $\mu(t)$, computed as in (6.41), is bigger than ϵ_P, if $\Delta\mathcal{V}(\Xi(t), \Omega(t), d_{max})$ does not contains a ball of finite radius, by following the above discussion, after a finite time it will and then we can conclude that all agents are capable to modify their command w.r.t g'. On the contrary if $\mu(t) < \epsilon_P$, agents will change their status st_i first in "P2S" and then in "SEQ" and as already seen, in the worst case, they will be enabled to change their command after $2d_{max}$ steps, when their status st_i will again equal "PAR".
Thus, if we are not at a Pareto-optimal solution at time t, at least one of the agents will move from it by a finite amount of time steps.

Remark 6.5. Note that, because of **A1** and **A2** whenever g converges to a point $\hat{r}$ then

$$\lim_{t\to\infty} \hat{x}(t) = x_{\hat{r}}, \quad \lim_{t\to\infty} \hat{y}(t) = y_{\hat{r}} = \hat{r}, \quad \lim_{t\to\infty} \hat{c}(t) = c_{\hat{r}}. \tag{6.52}$$

6.3.1 Computational Details

In this subsection a complete description of the computation details required to implement the above parallel strategy is presented. In particular, we describe all the needed technicalities required to implement the optimizations problems (6.43) and (6.46). Problems (6.41) and (6.48) can be computed by the same procedures described in Section 5.4.1 for Problems (5.49) and (5.55).

In order to build up Problem (6.43), we address the details for the computation of $\Delta\mathcal{V}(\Xi, \Omega, \bar{k})$ defined in (6.38) and its Cartesian decomposition arising in (6.42). Once problem (6.42) has been solved, the Cartesian decomposition produces a box $\mathcal{B}(\underline{\Delta g}(t), \overline{\Delta g}(t)) = \Delta\mathcal{V}_1(t), \times, ..., \times\Delta\mathcal{V}_N(t)$ containing 0_m and each agent can choose its local command by solving

$$\begin{aligned} g_i(t) &= arg\min_{g_i} \|g_i - r_i(t)\|^2_{\Psi_i} \\ subject \quad to \ \underline{\Delta g}_i &\leq (g_i - g_i(t-1)) \leq \overline{\Delta g}_i \end{aligned} \tag{6.53}$$

that represents an alternative way to represent problem (6.43).

The set Ξ in (6.36) and (6.44) keeps a box shape structure that can be uni-vocally identified by only two vectors according to Definition (5.16)

$$\Xi = \mathcal{B}(\underline{\xi}, \overline{\xi}). \tag{6.54}$$

As consequence, the set Ω in (6.37) has a polytopic structure

$$\Omega = \{x \in \mathbb{R}^n \,|A_\Omega x \leq b_\Omega\}. \tag{6.55}$$

The notation introduced in (6.54) and (6.55) is instrumental to easily compute the set $\Delta\mathcal{V}(\Xi, \Omega, \bar{k})$. The drawback, in this case, could arise from the fact that, according to Definition (6.38), it is needed to compute the set $\mathcal{V}(x, \bar{k})$ for an infinite number of points $(x, g) \in \Omega \times \Xi$. A solution can be arranged in two steps by computing an inner-approximation of the set $\Delta\mathcal{V}(\Xi, \Omega, \bar{k})$. The first step consists in exploiting the polyhedral structure of set $\mathcal{C}$ in (2.18)-(2.21) and (2.34)-(2.51) and separately considering their half-spaces

$$\begin{aligned} \mathcal{C}^j &:= \{c \in \mathbb{R}_c^n : T_j^T c \leq q_j\} \\ \mathcal{C}_\infty^j &:= \{c \in \mathbb{R}_c^n : T_j^T c \leq q_j^{k_\epsilon}\} \\ \mathcal{C}_\infty^{\delta,j} &:= \{c \in \mathbb{R}_c^n : T_j^T c \leq q_j^{k_\epsilon} - \delta\sqrt{T_j^T T_j}\} \end{aligned} \tag{6.56}$$

whose intersection constitutes the constrained polyhedrons $\mathcal{C} = \bigcap_{j=1}^z \mathcal{C}^j$, $\mathcal{C}_\infty = \bigcap_{j=1}^z \mathcal{C}_\infty^j$ and $\mathcal{C}_\infty^\delta = \bigcap_{j=1}^z \mathcal{C}_\infty^{\delta,j}$. Then, we compute for each set $\mathcal{C}^j$

$$\begin{aligned} \Delta\mathcal{V}^j(\Xi, \Omega, \bar{k}) = \{\Delta g | (g + \Delta g) \in \mathcal{G}^j(x, \bar{k}), \forall g \in \Xi, x \in \Omega\} \cap \\ \{\Delta g | (g + \Delta g) \in \mathcal{W}_\delta^j, \forall g \in \Xi\} \end{aligned} \tag{6.57}$$

where

$$\mathcal{W}_\delta^j := \{g \in \mathbb{R}^m : c_g \in \mathcal{C}_\infty^{\delta,j}\} \tag{6.58}$$

and

$$\mathcal{G}^j(x, \bar{k}) := \bigcap_{k=0}^{k_0} \mathcal{G}_k^j(x, \bar{k}) \tag{6.59}$$

being

$$\mathcal{G}_k^j(x, \bar{k}) := \{g : T_j^T H_c \Phi^k x + T_j^T R_k^c g \leq q_j^{k+\bar{k}}\} \tag{6.60}$$

with R_k^c borrowed from the notation used in (2.26)-(2.28). Hence, if $\Delta\mathcal{V}^j(\Xi, \Omega, \bar{k})$, $j = 1, \ldots, z$ were computed we would have an inner-approximation of $\Delta\mathcal{V}(\Xi, \Omega, \bar{k})$ because of the following lemma.

Lemma 6.6. *The intersection between sets $\Delta\mathcal{V}^i(\Xi, \Omega, \bar{k}) \bigcap_{i \neq j} \Delta\mathcal{V}^j(\Xi, \Omega, \bar{k})$ is always contained or at the most coincides with $\Delta\mathcal{V}(\Xi, \Omega, \bar{k})$, i.e.*

$$\bigcap_{j=1}^z \Delta\mathcal{V}^j(\Xi, \Omega, \bar{k}) \subseteq \Delta\mathcal{V}(\Xi, \Omega, \bar{k}). \tag{6.61}$$

Proof reported in the Appendix.

The second step is to compute all sets $\mathcal{V}^j(\Xi, \Omega), j = 1, ..., z$. They can be determined in an easy way by separately computing the sets $\{\Delta g | (g + \Delta g) \in$

$\mathcal{W}_\delta^j, \forall g \in \Xi\}$ and $\{\Delta g | (g + \Delta g) \in \mathcal{G}^j(x, d_{max}), \forall g \in \Xi, x \in \Omega\}$. The first set is determined by finding for each hyperplane $\mathcal{W}_\delta^j$ of $\mathcal{W}_\delta$ the point $\bar{g}^j \in \Xi$ that minimizes the distance between Ξ and the j-th hyperplane $T_j^T c_g = q_j$ of $\mathcal{W}_\delta$. Such a point can be determined as the solution of the following LP optimization problem

$$\bar{g}^j = \arg\min_{g \in \Xi} \rho_g^j. \tag{6.62}$$

with

$$\rho_g^j = \frac{q_j - T_j^T c_g}{\|T_j\|}, \tag{6.63}$$

$$c_g = H_c(I_n - \Phi)^{-1} G. \tag{6.64}$$

It is worth remarking that the constraints $g \in \Xi$ simply indicate that, by resorting to definition (6.54), the point g is between two vectors, i.e. $\underline{\xi} \leq g \leq \overline{\xi}$. For the computation of the second set it is needed to compute each set $\mathcal{G}_k^j(x, d_{max}), k = 0, ..., k_0$ composing $\mathcal{G}^j(x, d_{max})$. Actually, for each $k \in [0, k_0]$ we can determine pairs $(\bar{x}_k^j, \bar{g}_k^j) \in \Omega \times \Xi$ as solution of the following LP optimization problem

$$(\bar{x}_k^j, \bar{g}_k^j) = \arg \min_{x \in \Omega, g \in \Xi} \left(q_j^{k+\bar{k}} - T_j^T H_c \Phi^k x + T_j^T R_k^c g \right). \tag{6.65}$$

Then, the set

$$\begin{array}{l} \Delta\mathcal{V}^j(\bar{g}^j \cup \{\bar{g}_k^j\}_{k=0}^{k_0}, \{\bar{x}_k^j\}_{k=0}^{k_0}, \bar{k}) = \\ \bigcap_{k=0}^{k_0} \{\Delta g | (\bar{g}_k^j + \Delta g) \in \mathcal{G}_k^j(\bar{x}_k^j, \bar{k})\} \cap \{\Delta g : (\bar{g}^j + \Delta g) \in \mathcal{W}_\delta^j\}. \end{array} \tag{6.66}$$

can be used in the place of $\Delta\mathcal{V}^j(\Xi, \Omega, \bar{k})$, being this choice justified by the following lemma

Lemma 6.7. *The set $\Delta\mathcal{V}^j(\bar{g}^j \cup \{\bar{g}_k^j\}_{k=0}^{k_0}, \{\bar{x}_k^j\}_{k=0}^{k_0}, \bar{k})$ is always contained or at the most coincides with $\Delta\mathcal{V}^j(\Xi, \Omega, \bar{k})$, i.e.*

$$\Delta\mathcal{V}^j(\bar{g}^j \cup \{\bar{g}_k^j\}_{k=0}^{k_0}, \{\bar{x}_k^j\}_{k=0}^{k_0}, \bar{k}) \subseteq \Delta\mathcal{V}^j(\Xi, \Omega, \bar{k}). \tag{6.67}$$

Proof reported in the Appendix.

An immediate consequence of Lemmas 6.6 and 6.7 is that

$$\bigcap_{j=1}^{z} \Delta\mathcal{V}^j(\bar{g}^j \cup \{\bar{g}_k^j\}_{k=0}^{k_0}, \{\bar{x}_k^j\}_{k=0}^{k_0}, \bar{k}) \subseteq \Delta\mathcal{V}(\Xi, \Omega, \bar{k}), \tag{6.68}$$

Hence, each agent can use $\Delta\tilde{\mathcal{V}}(\Xi, \Omega, \bar{k}) = \bigcap_{j=1}^{z} \Delta\mathcal{V}^j(\bar{g}^j \cup \{\bar{g}_k^j\}_{k=0}^{k_0}, \{\bar{x}_k^j\}_{k=0}^{k_0}, \bar{k})$ instead of $\Delta\mathcal{V}(\Xi, \Omega, \bar{k})$ for constraining its command variations in solving problem (6.42).

Next step, in order to build problem (6.43) is to illustrate how to undertake the box approximation underlying problem (6.42) for the determination of sets $\Delta\mathcal{V}_i, \forall i \in \mathcal{A}$ from $\Delta\tilde{\mathcal{V}}(\Xi, \Omega, \bar{k})$. For this purpose, it is convenient to put $\Delta\tilde{\mathcal{V}}(\Xi, \Omega, \bar{k})$ in a matrix form and apply the Algorithm (5.80) presented in Section 5.4.1. Once computed $\bar{g}^j$, $\{\bar{g}_k^j\}_{k=0}^{k_0}$ and $\{\bar{x}_k^j\}_{k=0}^{k_0}$ in (6.62) and (6.65), the set $\Delta\mathcal{V}^j(\bar{g}^j \cup \{\bar{g}_k^j\}_{k=0}^{k_0}, \{\bar{x}_k^j\}_{k=0}^{k_0}, \bar{k})$ in (6.66) takes the following matrix form

$$\Delta\mathcal{V}^j(\bar{g}^j \cup \{\bar{g}_k^j\}_{k=0}^{k_0}, \{\bar{x}_k^j\}_{k=0}^{k_0}, \bar{k}) = \left\{ \Delta g \;\middle|\; \begin{array}{l} S_{c,j}\Delta g \leq q_j^{k_\epsilon} - \delta[\sqrt{T_j^T T_j}] - S_{c,j}\bar{g}^j \\ R_k^{c,j}\Delta g \leq q_j^{k+\bar{k}} - T_j^T H_c \Phi^k \bar{x}_k^j \\ \qquad\qquad - R_k^{c,j}\bar{g}_k^j, k = 0, ..., k_0 \end{array} \right\} \tag{6.69}$$

where $S_{c,j} = T_j^T H_c (I_n - \Phi)^{-1} G + L$, $R_k^{c,j} = T_j^T R_k^c$ and k_0 is computed according to Algorithm (2.3.2). Eq. (6.69) allows one to express $\Delta\tilde{\mathcal{V}}(\Xi, \Omega)$ as a convex polyhedron of the form

$$\tilde{A}\Delta g \leq \tilde{b} \tag{6.70}$$

with

$$\tilde{A} = \begin{bmatrix} S_{c,1} \\ R_k^{c,1} \\ \vdots \\ S_{c,z} \\ R_k^{c,z} \end{bmatrix} \in \mathbb{R}^{(z(1+k_0))\times m}, \; \tilde{b} = \begin{bmatrix} q_1^{k_\epsilon} - \delta[\sqrt{T_1^T T_1}] - S_{c,1}\bar{g}^1 \\ q_1^{k+d_{max}} - T_j^T H_c \Phi^k \bar{x}_k^1 - R_k^{c,1}\bar{g}_k^1, \\ \vdots \\ q_z^{k_\epsilon} - \delta[\sqrt{T_z^T T_z}] - S_{c,z}\bar{g}^z \\ q_z - T_j^T H_c \Phi^k \bar{x}_k^z - R_k^{c,z}\bar{g}_k^z \end{bmatrix} \in \mathbb{R}^{z(1+k_0)} \tag{6.71}$$

By exploiting the matrix description of $\Delta\tilde{\mathcal{V}}(\Xi, \Omega)$, problem (6.42) becomes

$$\max_{\tilde{A}\underline{\Delta g} \leq \tilde{b}, \tilde{A}\overline{\Delta g} \leq \tilde{b}} V(\mathcal{B}(\underline{\Delta g}, \overline{\Delta g})) \tag{6.72}$$

As regards Problem (6.46), we give a procedure for its implementation in two steps

1. Approximate $\Delta\mathcal{V}(\Xi_i, \Omega_i, \bar{k})$ with $\Delta\tilde{\mathcal{V}}(\Xi_i, \Omega_i, \bar{k}) = \{\Delta g \in \mathbb{R}^m : \tilde{A}\Delta g \leq \tilde{b}\}$ by following the procedure (6.54)-(6.70) outlined above by considering Ξ_i in the place of Ξ, Ω_i in the place of Ω and d_i in the place of d_{max} .
2. Compute $\Delta\mathcal{V}^0(\Xi_i, \Omega_i, \bar{k})$ in (6.47) as

$$\Delta\mathcal{V}^0(\Xi_i, \Omega_i, \bar{k}) = \{\Delta g_i \in \mathbb{R}^{m_i} : \tilde{A}[0_1^T, ..., \Delta g_i^T, ..., 0_n^T]^T \leq \tilde{b}\} \tag{6.73}$$

Then, at time t problem (6.46) becomes

$$\begin{array}{l} g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel_{\Psi_i}^2 \\ \textit{subject to } \tilde{A}^{(i)}(g_i - g_i(t-1)) \leq b \end{array} \tag{6.74}$$

where $\tilde{A}^{(i)}$ collects all the m_i columns of $\tilde{A}$ that multiply Δg_i in (6.73).

6.4 Conclusions

In this Chapter, the distributed FF-CG schemes presented in Chapter 5 have been extended to the case where agents can deliver and share their local states over a communication network. The obtained distributed coordination algorithms have been fully investigated and the results on constraints fulfilment and stability rigorously presented.

6.5 Appendix

Proof of Lemma 6.6

This lemma can be easily proved by observing first that $\Delta\mathcal{V}(\Xi, \Omega, \bar{k})$ can be written as

$$\begin{aligned}\Delta\mathcal{V}(\Xi, \Omega, \bar{k}) = \{\Delta g | (g + \Delta g) \in \mathcal{G}(x, \bar{k}), \forall g \in \Xi, x \in \Omega\} \cap \\ \{\Delta g | (g + \Delta g) \in \mathcal{W}_\delta, \forall g \in \Xi\}\end{aligned} \tag{6.75}$$

where

$$\mathcal{G}(x, \bar{k}) := \bigcap_{j=1}^{z} \mathcal{G}^j(x, \bar{k}). \tag{6.76}$$

Because $\mathcal{W}_\delta = \bigcap_{j=1}^{z} \mathcal{W}_\delta^j$ and (6.76), one has

$$\begin{aligned}\Delta\mathcal{V}(\Xi, \Omega, \bar{k}) \equiv \bigcap_{j=1}^{z} \{\Delta g | (g + \Delta g) \in \mathcal{G}^j(x, \bar{k}), \forall g \in \Xi, \forall x \in \Omega\} \cap \\ \bigcap_{j=1}^{z} \{\Delta g | (g + \Delta g) \in \mathcal{W}_\delta^j, \forall g \in \Xi\},\end{aligned} \tag{6.77}$$

and, by exploiting basic properties of the $\cap$ operator, we can state that

$$\begin{aligned}\Delta\mathcal{V}(\Xi, \Omega, \bar{k}) \equiv \bigcap_{j=1}^{z} \big(\{\Delta g | (g + \Delta g) \in \mathcal{G}^j(x, \bar{k}), \forall g \in \Xi, \forall x \in \Omega\} \cap \\ \{\Delta g | (g + \Delta g) \in \mathcal{W}_\delta^j, \forall g \in \Xi\}\big) \equiv \bigcap_{j=1}^{z} \Delta\mathcal{V}^j(\Xi, \Omega, \bar{k}).\end{aligned} \tag{6.78}$$

Proof of Lemma 6.7

In order to prove eq. (6.67), it is sufficient to prove that

$$\{\Delta g \in \mathbb{R}^m | \bar{g}^j + \Delta g \in \mathcal{W}_\delta^j\} \subseteq \{\Delta g \in \mathbb{R}^m | g + \Delta g \in \mathcal{W}_\delta^j, \forall g \in \Xi\} \tag{6.79}$$

and

$$\bigcap_{k=0}^{k_0} \{\Delta g | (\bar{g}_k^j + \Delta g) \in \mathcal{G}_k^j(\bar{x}_k^j, \bar{k})\} \subseteq \{\Delta g | (g + \Delta g) \in \mathcal{G}^j(x, \bar{k}), \forall g \in \Xi, \forall x \in \Omega\} \tag{6.80}$$

In fact (6.79)-(6.80) imply that

$$\{\Delta g \in \mathbb{R}^m \,|\bar{g}^j + \Delta g \in \mathcal{W}_\delta^j\} \cap \bigcap_{k=0}^{k_0} \{\Delta g | (\bar{g}_k^j + \Delta g) \in \mathcal{G}_k^j(\bar{x}_k^j, \bar{k})\} \subseteq$$
$$\{\Delta g \in \mathbb{R}^m \,|g + \Delta g \in \mathcal{W}_\delta^j, \forall g \in \Xi\} \cap \{\Delta g | (g + \Delta g) \in \mathcal{G}^j(x, \bar{k}), \forall g \in \Xi, \forall x \in \Omega\} \tag{6.81}$$

or equivalently (6.67) because of the definition for $\Delta\mathcal{V}^j(\Xi, \Omega, \bar{k})$ given in (6.57) and $\mathcal{V}^j(\bar{g}^j \cup \{\bar{g}_k^j\}_{k=0}^{k_0}, \{\bar{x}_k^j\}_{k=0}^{k_0}, \bar{k})$ in (6.66).

Condition (6.79) has been already proved in the proof of Lemma 5.30. Then we focus our attention on eq. (6.80) only. In order to prove it, (6.59) and (6.60), suggest to establish for each $k = 0, ..., k_0$ that

$$\{\Delta g | (\bar{g}_k^j + \Delta g) \in \mathcal{G}_k^j(\bar{x}_k^j, \bar{k})\} \subseteq \{\Delta g | (g + \Delta g) \in \mathcal{G}_k^j(x, \bar{k}), \forall g \in \Xi, \forall x \in \Omega\} \tag{6.82}$$

Let us make explicit the structure of the sets appearing in (6.82)

$$\{\Delta g | (\bar{g}_k^j + \Delta g) \in \mathcal{G}_k^j(\bar{x}_k^j, \bar{k})\} = \{\Delta g : T_j^T R_k^c \Delta g \leq q_j^{k+\bar{k}} - T_j^T H_c \Phi^k \bar{x}_k^j - T_j^T R_k^c \bar{g}_k^j\} \tag{6.83}$$
$$\{\Delta g | (g' + \Delta g) \in \mathcal{G}_k^j(x', \bar{k})\} = \{\Delta g : T_j^T R_k^c \Delta g \leq q_j^{k+\bar{k}} - T_j^T H_c \Phi^k \bar{x}_k^j - T_j^T R_k^c g'\} \tag{6.84}$$

Moreover let us compute the set appearing in (6.84) in the generic points $g' \neq \bar{g}_k^j$ and $x' \neq \bar{x}_k^j$ belonging to Ξ and Ω respectively for any $k \in [0, k_0]$. Because of (6.65), the following inequality holds true

$$q_j^{k+\bar{k}} - T_j^T H_c \Phi^k \bar{x}_k^j - T_j^T R_k^c \bar{g}_k^j \leq q_j^{k+\bar{k}} - T_j^T H_c \Phi^k x' - T_j^T R_k^c g', k \in [0, k_0] \tag{6.85}$$

Inequality (6.85) and structures (6.83)-(6.84) suggest that all Δg belonging to $\{\Delta g | (\bar{g}_k^j + \Delta g) \in \mathcal{G}_k^j(\bar{x}_k^j, \bar{k})\}$ are in $\{\Delta g | (g' + \Delta g) \in \mathcal{G}_k^j(x', \bar{k})\}$ which can be formalized as

$$\{\Delta g \in \mathbb{R}^m \,|\bar{g}^j + \Delta g \in \mathcal{G}_k^j(\bar{x}_k^j, \bar{k})\} \subseteq \{\Delta g \in \mathbb{R}^m \,|g' + \Delta g \in \mathcal{G}_k^j(x', \bar{k})\} \tag{6.86}$$

Because genericity of $g' \in \Xi$ and $x' \in \Omega$, (6.86) is satisfied for all $g \in \Xi$ and $x \in \Omega$. This means that (6.82) is true and, in turn, (6.80) is proved.

7

Iterative Distributed Command Governor Schemes

In this Chapter we present a class of iterative methods suitable to build up distributed CG and FF-CG strategies. The goal here is to recast the CG and FF-CG distributed control design problems stated in Sections 6.1 and 5.1 into multi-objective optimization schemes and solve them in a distributed way. In the literature several methods for solving such a kind of problems have been reported (see [59]). Amongst many, we resort here to the iterative distributed optimization procedure presented in [60, 61][1] which is based on the use of penalty functions.

The Chapter is organized as follows: in Section 7.1 the distributed optimization method proposed in [60, 61] is recalled and some notations and results taken from [61] are reported. Then, in Sections 7.2 and 7.3, customizations of such a method to address the specific features of the CG and, respectively, FF-CG approaches are presented.

7.1 A distributed optimization algorithm

In this Section, we introduce the penalty functions method proposed in [59] to solve in a distributed way a multi-objective optimization problem. This penalty functions method provides a connection between constrained and unconstrained optimization and it is a useful design structure for proving global properties of the solution, such as convergence, based on local solutions and actions of the single agents. Later, an iterative algorithm that uses this penalty-functions method is presented in order to solve the distributed optimization problems of interest.

[1] In that work the Author refers to a *decentralized* procedure. Here, according to definitions given in the Introduction we prefer to consider such an approach as a distributed method.

7.1.1 Problem Formulation

By means of the notions of decomposition variables and over-lapping constraints [63], a multi-objective optimization problem composed by N local functions $f_i : \mathbb{R}^{m_i} \to \mathbb{R}_+$ can be described, from a standard centralized point of view, as follows:

Definition 7.1. ***(Centralized Optimization problem)***

$$\begin{array}{ll} \min_{g\in\mathbb{R}^m} & [f_1(g_1), ..., f_i(g_i), ..., f_N(g_N)] \\ s.t. & h(g) \leq 0 \end{array} \tag{7.1}$$

Here, the overall vector $g = [g_1^T, g_2^T, ..., g_N^T]^T \in \mathbb{R}^m$ collects all the optimization variables of each agent $g_i \in \mathbb{R}^{m_i}, \forall i \in \mathcal{A} = \{1, ..., N\}$, and $h : \mathbb{R}^m \to \mathbb{R}^e$ represents the constraints to be fulfilled. The following optimality notion will be used for the above optimization problem:

Definition 7.2. ***Pareto Optimal solution:*** *The vector $g^{*_P} \in \mathcal{S} = \{g \in \mathbb{R}^m : h(g) \leq 0\}$ is a Pareto optimal solution to the centralized optimization problem (7.1) if there exist no other solutions $g \in \mathcal{S}, j \in \mathcal{A}$, such that: $f_i(g_i) \leq f_i(g_i^{*_P})$, $\forall i \in \mathcal{A}$ and $f_j(g_j) < f_j(g_j^{*_P})$.*

A centralized implementation and solution of the optimization problem 7.1 require a central computational facility with access to all system information. On the contrary, here we are interested in implementations on N computational nodes each with restricted information about the whole system. To encode the limited information horizons of all agents, we require the notion of neighborhood of a given agent i:

Definition 7.3. ***(Neighborhood of the i-th agent:)*** *The neighborhood of the i-th agent is defined as the set of all other agents j whose decision variables g_j are jointly involved with g_i in some constraints and have a direct communication link with node i, that is $\mathcal{N}_i = \{j :$ the i-th agent has a constraint involving the j-th agent and directly communicates with it $\}$*

Remark 7.4. Notice that in this Chapter the notation of neighborhood is quite more stronger than the same definition used in the previous Chapters.

As an immediate consequence of Definition 7.3, we introduce the set

$$[g]_i = \{\text{All subvectors } g_j \text{ of } g \text{ such that } j \in \mathcal{N}_i\} \in \mathbb{R}^{|\mathcal{N}_i|} \tag{7.2}$$

as the set of all neighborhood j associated to the i-th agent. Under this set, the problem of interest can be defined

Definition 7.5. ***(Decentralized Optimization Problem)*** *The decentralized optimization problem for each agent i, is defined as:*

$$\begin{array}{ll} \min_{g_i\in\mathbb{R}^{n:i}} & f_i(g_i) \\ s.t. & h_i(g_i|[g]_i) \leq 0 \end{array} \tag{7.3}$$

where the notation $h_i(\cdot|[g]_i)$ is used to represent functions of g_i given that the neighborhood decision variables $[g]_i$ are fixed.

Moreover we define

$$\mathcal{S}_i = \{g_i : h_i(g_i|[g]_i) \leq 0\} \tag{7.4}$$

as the feasible region for the i-th agent given a particular neighborhood value, $[g]_i$. We define the optimality for the decentralized optimization problem by using the concept of Nash equilibria

Definition 7.6. ***(Nash Equilibrium for Decentralized Coordination)*** *$g^{*de} = [g_1^{*de^T}, ..., g_i^{*de^T}, ..., g_m^{*de^T}]^T \in \mathcal{S}$ is a Nash equilibrium for the decentralized optimization problem if for any $g_i \in \mathcal{S}_i$,*

$$f_i(g_i^{*de}) \leq f_i(g_i) \forall i \in \mathcal{A} \tag{7.5}$$

Pareto Optimal solutions differ from Nash equilibrium solutions $g^{*n} \in \mathcal{S}$ to the centralized problem, which have the property that $f_i(g_i^{*n}) \leq f_i(g_i), \forall i \in \mathcal{A}$, where $[g_1^{*n^T}, g_2^{*n^T}, ..., g_i^{*n^T}, ..., g_m^{*n^T}]^T \in \mathcal{S}$. The following result allows us to map optimal results in the decentralized optimization problem directly back to the original centralized problem with complete information model.

Proposition 7.7. ***(Equivalence of Centralized and Decentralized Nash Equilibria)****: g^* is a Nash equilibrium of the centralized optimization problem (7.1) if and only if it is a Nash equilibrium of the decentralized optimization problem (7.3).*

Proof See [60, 61].

7.1.2 Penalty Methods and Block Iterations for Decentralized Problems

Penalty methods (e.g. [62]) provide a convenient way for solving decentralized optimization problems. In fact, as it will shown, this class of methods will allow us to address the possible presence of infeasible constraints imposed by the different subsystems is an effective way. However, unlike centralized or distributed methods, the fundamental property of the intermediate solutions of being bounded above by the optimal solution value does't hold any longer in a decentralized context. Thus, the direct implementation of such penalty function methods can result in numerically ill-conditioned problems as the local penalty parameters are driven to infinity. For the decentralized optimization problem described in (7.3), we consider a modified form for the penalty augmented cost function of the i-th agent,

$$\begin{aligned} F_i(g_i, \beta_i|[g]_i) &= \beta_i f_i(g_i) + P_i(g_i|[g]_i) \\ &= \beta_i[f_i(g_i) + \tfrac{1}{\beta_i} P_i(g_i|[g]_i)], \beta_i \neq 0 \end{aligned} \tag{7.6}$$

where $\beta_i \geq 0$ is our new local penalty parameter. Here the local penalty function, P_i, penalizes violations of the constraints given in (7.3) (i.e. $P_i(g_i|[g]_i) =$

$0 \iff g_i \in \mathcal{S}_i$). By using this modified method, local optimization for each agent can then be defined as follows:

Definition 7.8. ***(Local Optimization)*** *Local optimization for the i-th agent in the decentralized optimization problem for fixed value of* $[g]_i$ *(neighborhood variables) and* β_i *is defined as*

$$\min_{g_i \in \mathbb{R}^{n_i}} F_i(g_i, \beta_i | [g]_i) \tag{7.7}$$

with the optimal solution denoted as

$$[g_i^* | \beta_i, [g]_i] = \arg \min_{g_i \in \mathbb{R}^{n_i}} F_i(g_i, \beta_i | [g]_i) \tag{7.8}$$

A good choice for $P_i(g_i|[g]_i)$ could be $P_i(g_i|[g]_i) = \sum_{k=1}^{e_i} \max(0, h_{i,k}(g_i|[g]_i))^\gamma$, $\gamma \geq 2$. For simplicity of notation, we denote the solution of the local optimization problem for a particular β_i^l and $[g]_i$ as g_i^l, and the solution to the problem with β_i^{l+1} and $[g]_i$ as g_i^{l+1}. It is simple to show that, for a local optimization, F_i e P_i decrease as β_i decreases

Lemma 7.9. *For a fixed value of* $[g]_i$*, if* $\beta_i^l > \beta_i^{l+1} \geq 0$ *and* $f_i(g_i^l) > 0$ *then*

$$F_i(g_i^{l+1}, \beta_i^{l+1} | [g]_i) < F_i(g_i^l, \beta_i^l | [g]_i) \tag{7.9}$$

$$P_i(g_i^{l+1} | [g]_i) \leq P_i(g_i^l | [g]_i) \tag{7.10}$$

Proof See [60, 61].

This property allows each agent to use the local β_i selection as a tool to achieve possibly less constraints violation (and indirectly, more cooperation) while not resulting in an increase in $F_i(g_i, \beta_i|[g]_i)$.

At the end of this Section we will present an algorithm, which combines sequential subsystem optimizations with a bargaining scheme between subsystems. Before doing this, we present two key assumptions and a metric which will be used to analyze the convergence properties of the algorithm.

A4. $F_i(g_i^{l+1}, \beta_i^{l+1}|[g]_i)$ embeds all the constraints $h_i(g_i, [g]_i) \leq 0$ that g_i is associated with.

In addition,we state the second assumption:

A5. Common global constraints (i.e. interconnection of agent constraints) and their penalty functions enter each associated agent optimization problem identically: $h_{i,j}(g_i|g_k, ...) = h_{k,l}(g_k|g_i, ...)$ e $P_{h_{i,j}}(g_i|g_k, ...) = P_{h_{k,l}}(g_k|g_i, ...)$ where $h_{i,j}$ represents a constraint denoted as the j-th constraint of the i-th agent.

Finally, we introduce a metric under which we will be able to prove contraction through sequential local optimizations. By using this metric and thanks to Assumptions **A4** and **A5**, the global cost function can be defined as follows:

Definition 7.10. *(**Global cost function**) The global cost function described as*

$$F(g,\beta)=\sum_{i=1}^{m}[\beta_i f_i(g_i)+P_i(g_i,[g]_i)] \tag{7.11}$$

is the cost associated to the distributed optimization problem; $F:\mathbb{R}^n\times\mathbb{R}^N\to\mathbb{R}$ *and* $\beta=[\beta_1,...,\beta_m]^T\in\Gamma\subset\mathbb{R}_+^m$*, where* Γ *is compact.*

Assumption **A5** dictates that for each global constraint, the penalty function related to that constraint is "viewed identically" by each of the decision makers involved in this constraint fulfilment. Thus, a single penalty function dealing with such a global constraint enters into the summation.

From the i-th agent perspective, the global cost function can be broken into two parts, representing respectively the local optimization function for the i-th agent $F_i(g_i,\beta_i|[g]_i)$ and the remainder defined as its complement, $\bar{F}_i(g\setminus g_i,\beta\setminus\beta_i)$:

$$\begin{aligned}F(g,\beta)&=F(g_i,\beta_i|g\setminus g_i,\beta\setminus\beta_i)\\&=\beta_i f_i(g_i)+P_i(g_i|[g]_i)+\bar{F}_i(g\setminus g_i,\beta\setminus\beta_i)\\&=F_i(g_i,\beta_i|[g]_i)+costant\end{aligned}$$

Here the $w\setminus v$ notation refers to all entries of w excluding the entries of v. Notice that F is a function of g_i and β_i for given $g\setminus g_i$ (and $\beta\setminus\beta_i$). Thus, $F_i(g_i,\beta_i|[g]_i)$ and $F(g_i,\beta_i|g\setminus g_i,\beta\setminus\beta_i)$ are minimized by the same g_i. This implies that optimizing $F_i(g_i,\beta_i|[g]_i)$ actually corresponds to optimize $F(g,\beta)$ while $g\setminus g_i$ and $\beta\setminus\beta_i$ are kept fixed. If these optimizations are recursively carried by all agents, the overall effect emulates a nonlinear block optimization iteration on $F(g,\beta)$.

In the following, we want to design a scheme which uses sequential optimizations locally, where each agent i solves a sequence of local optimization problems involving its neighborhood $\mathcal{N}_i$ only. In particular, at each iterate it optimizes the cost (7.11) by a local selection of β_i which is then made aware to all other agents $j\in\mathcal{N}_i$. As this process is being done simultaneously by each agent, exponential growth in the number of solutions is avoided by introducing local elimination criteria. In fact solutions branch in time when they are transmitted from one agent to its neighborhood, then some of the solutions get "eliminated" from the many solutions received during a particular time interval T_m when a subsystem selects one by a local selection criteria (such as minimum constraint violation or minimum local cost). Before the local selection taking place, each of the received solutions is optimized via (7.11) in totally independent parallel running processes which are *spawned*. In order to understand this property, we define a *thread* to be a solution that

exists within the coordination algorithm after k steps. Multiple threads (indexed below using w superscript) can exist at anytime within the coordination algorithm.

In order to aid each individual agent in its local selection, one can provide information about the global performance of the thread within the decentralized structure. In our case, we introduce the metrics ΔF_{total} and ΔP_{total} which keep track of the total cost and total constraints violation decrements in time. These parameters are updated in a decentralized fashion via the local cost decrements $\Delta F_i(k)$ and $\Delta P_i(k)$ at any k-th step.

In order to keep track of the convergence of a thread, we introduce a set of flag vectors $C = \{C_i\}$ and $TC = \{TC_i\}$ which signal respectively (a) if the agent has converged to a solution for that particular thread and (b) if all of the agents in its neighborhood have converged. The collection of solutions, the metrics and the flags are passed in the information vector $I = I_i, i \in \mathcal{A}$. The update is done locally through the *subconcat* operator, which replaces only the portion corresponding to the i-th subsystem I_i. After initialization, as the solutions propagate through the network of agents, the complete operator allows each subsystem to reconstruct the missing portions of I: any missing variable of $[g]_i$ is inserted by the subsystem, by using the original values that this subsystem had used to generate the solution in the received I.

During the evolution of the sequential optimization process, the subsystems are effectively bargaining: they propose a solution $(g_i^+|[g]_i)$ and receive a counter offer $([g_j^{++}]_i|g_i^+)$ when the other agents in the neighborhood change their individual moves[2].

Below, a pseudo-code implementing the algorithm for the generic agent i with neighborhood $\mathcal{N}_i$ is reported.

Algorithm 7.1.1 ***(Algorithm for Distributed Optimization)***

initialization

[1.1] TRANSMIT $g_i = g_i(0), I_i = I_i(0)$
[1.2] RECEIVE $[g(0)]_i = \{g_j(0)\}_{j\in\mathcal{N}_i}, I = \{I_i(0), I_j(0)\}\forall j \in \mathcal{N}_i$
[1.3] SET $k = 0$ RECEIVED DATA=1
[1.4] WRITE $g_i(0), [g(0)]_i, I(0)$ IN BATCH
[1.5] SET $t_i = 0, w = 0$

main:

[1.1]WHILE $t_i < T_m$ DO:
  [1.1.1]IF DATA RECEIVED=0
    [1.1.1.2]WAIT,
  [1.1.2]ELSE
    [1.1.2.1]SET $w = w + 1$

[2] The selection of β_i gives each subsystem a tool to "bargain": for large values of β_i the resulting solution provides minimal constraints satisfaction; as β_i decreases, the constraints satisfaction (and indirectly the cooperation) increases, as shown in Lemma 7.9

[1.1.2.2]RECEIVE $(I^w(k))$
[1.1.2.3]COMPLETE $(\{g_j^w(k)\}_i)$
[1.1.2.4]SELECT $\beta_i^w(k+1)$
[1.1.2.5]SPAWN (**optimize**$(\{g_j^w(k)\}_i, \beta_i^w(k+1), \Delta F_{total}^w(k), \Delta P_{total}^w(k)))$
[1.1.2.6]SET DATA RECEIVED=0
END
[1.1.3]EXECUTE (**localselect**(BATCH))
[1.1.4]SET $t_i = 0, w = 0$
[1.1.5]GO TO **main**

optimize(g_j, I_j):
[1.1] $g_i^w(k+1) = \arg\min F_i(g_i, \beta_i^w(k+1)|\{g_j^w(k)\}_i)$
[1.2] $\Delta P_i^w(k+1) = P_i^w(g_i^w(k)|\{g_j^w\}_i(k)) - P_i^w(g_i^w(k+1)|\{g_j^w\}_i(k))$
[1.3] $\Delta F_i^w(k+1) = F_i^w(g_i^w(k), \beta_i^w(k)|\{g_j^w\}_i(k)) - F_i^w(g_i^w(k+1), \beta_i^w(k+1)|\{g_j^w\}_i(k))$
[1.4] $\Delta F_{total}^w(k+1) = \Delta F_{total}^w(k) + \Delta F_i^w(k+1)$
[1.5] $\Delta P_{total}^w(k+1) = \Delta P_{total}^w(k) + \Delta P_i^w(k+1)$
[1.6]IF $\Delta F_{total}^w(k) < \epsilon | k \geq N_{it}$
[1.6.1] SET $C_i = 1$
[1.6.2] IF $C_j = 1 \forall j \in N_i$ SET $TC_i = 1$
[1.7] $I^w(k+1) =$ **subconcat**$(g_i^w(k+1), C, TC, I^w(k))$
[1.8]WRITE $g_i^{k+1}, \beta_i^w(k+1), I^w(k+1), f_i(g_i^w(k+1))$ TO BATCH

localselect (Batch):
[1.1] $g_i(k+1), I(k+1) =$ **select**(LOCAL CRITERIA,BATCH)
[1.2]TRANSMIT $I(k+1)$ A $j \in \mathcal{N}_i$
[1.3]WRITE $I(k+1)$ TO MEMORY
[1.4]SET $k = k+1$

For the remainder of our discussion, we denote $g^d(k) = [g_1^d(k)^T, ..., g_i^d(k)^T, ..., g_m^d(k)^T]^T$ as the optimization variables of a thread still valid within the coordination scheme after k steps. In addition, for compactness of representation, we denote $\beta^d(k) = [\beta_1^d(k), ..., \beta_i^d(k), ..., \beta_m^d(k)]$ as the collection of the penalty parameters used for generating $g^d(k)$.

Proposition 7.11. ***(Global Convergence of Algorithm 7.1.1)*** *Let $\{g^d(k)\}$ correspond to a thread generated by the sequential optimization process (7.1.1) by using a non-increasing sequence of penalty parameters $\{\beta^d(k)\} \to \beta^*$. Given Assumptions **A4** and **A5**, the global cost function (Definition 7.11) $\{F(g^d(k)), \beta^d(k)\}$ converges to a constant limit F^*, for any $g(0) \in X$. In addition, the algorithm terminates in a finite number of iterations for any termination criterion such as $F(g^d(k), \beta^d(k)) - F(g^d(k+1), \beta^d(k+1)) \geq \epsilon, \forall i = 1, ..., N$, with $\epsilon > 0$. This means that none of the N agents can improve local costs associated to $(g_i^d | [g]_i^d)$ by more than ϵ.*

Proof See [60, 61].

7.1.3 Optimality Analysis

Although Proposition 7.11 proves the convergence of the sequential distributed optimization of Algorithm 7.1.1, it neither provides any indications of the convergence to a particular solution nor insight to the solution types (i.e. feasibility and the centralized optimality of the result). In this section, we analyze these two key points. To this end, further assumptions need to be introduced.

A6. Each optimization vector g_i takes values over a compact subset $\Omega_i \subset \mathbb{R}^{m_i}$ with $0 \leq \beta_i \leq K < \infty$. Functions $F_i(g_i, \beta|[g]_i) : \mathbb{R}^{m_i} \times \mathbb{R} \times \mathbb{R}^{N_i} \to \mathbb{R}$ are assumed to be with in the class of $\mathcal{C}^2(\mathbb{R}^{m_i+1+N_i}, \mathbb{R})$.

By this assumption $F(g, \beta) \in \mathcal{C}^2(\mathbb{R}^{N+m}, \mathbb{R})$ through the construction of $F(g, \beta)$ given in Definition 7.11.

Proposition 7.12. ***(Solution is a Nash Equilibrium)*** *Let $\{g^d(k)\}$ correspond to a sequence generated by sequential optimization process of Algorithm (7.1.1) by using a non-increasing sequence of penalty parameters $\{\beta^d(k)\} \to \beta^*$. If Assumption **A6** is satisfied, then there exists at least one cluster point g^{*de} (not necessarily feasible) for which $F(g^{*de}, \beta^{*de}) = F^*$ where $\beta^{*de} = \beta^*$ corresponds to the associated penalty parameters. In addition, all such g^{*de} are Nash equilibria for $F_i(g_i, \beta_i|[g]_i), \forall i = 1, ..., N$.*

Proof See [60, 61].

Moreover, in [60] the well-known constraint qualification condition is exploited in order to prove the feasibility of the computed solution

Definition 7.13. ***(Linear independence Constraint Qualification (LICQ))*** *Let $I(g) = \{s|h_s(\bar{g}) \geq 0, s = \{1, ..., e\}\}$ represent the active constraint set at $g = \bar{g}$. Then the linear independence constraint qualification condition is satisfied at $\bar{g}$, if $\frac{\partial h_j}{\partial g}|_{g=\bar{g}} \in I(\bar{g})$ are linearly independent.*

The LICQ condition provides the sufficient assumption to eliminate cases where two or more agents converge to an infeasible solution and cannot move based on the fact that there are linearly dependent counter-opposing gradient (descent) directions.

Proposition 7.14. ***(Global Convergence to Feasible Solutions)****:*
*Let $F_i(g_i, \beta i|\{g_j\}_i) : \mathbb{R}^{m_i} \times \mathbb{R} \times \mathbb{R}^{N_i} \to \mathbb{R}$ satisfy Assumption **A6**. Then, the decentralized optimization algorithm will converge globally (for any $g(0) \in \Omega$) to a feasible solution, g^{*de}, as the non-increasing $\{\beta_i\} \to 0 \forall i = 1, ..., N$, if g^{*de} satisfies the linear independence constraint qualification condition (LICQ 7.13).*

Proof See [60, 61].

By using proposition (7.14), a stronger result, which makes a connection between the decentralized optimal solution and Pareto-optimality for the centralized problem (7.1) can be presented.

Proposition 7.15. *(ϵ - **Optimality**) Let $\{\beta\} \to \beta^{*de} > 0$ be a non-increasing sequence where β^{*de} can be brought arbitrarily close to zero. By assuming that the decentralized optimal solution g^{*de} satisfies the LICQ condition, then for any given $\epsilon > 0$, g^{*de} can be brought ϵ-close to satisfying each of the constraints $h_j(g) \leq 0, \forall j = 1, ..., e$. In addition, g^{*de} will be a Pareto Optimal solution for a relaxed version of the centralized problem (7.1) which is no more than ϵ-far from a feasible solution.*

Proof See [60, 61].

7.2 FFCG based distributed iterative Schemes (DI-FFCG)

In this section we discuss how to customize the above presented procedure to the case of FF-CG problem described in Chapter 3 applied to a distributed system. To this end, let us consider a set of N subsystems $\mathcal{A} = \{1, \ldots, N\}$ as defined in (6.1) of Section 6.1 and their composition (2.1). The problem that we want to consider has been already faced in Chapters 5 and 6 and consists in locally determining, at each time instant t and for each agent $i \in \mathcal{A}$, a suitable reference signal $g_i(t)$ which is the best feasible approximation of $r_i(t)$ and such that its application never produces constraint violations, *i.e.* $c_i(t) \in \mathcal{C}^i, \forall t \in \mathbb{Z}_+, i \in \mathcal{A}$, with $\mathcal{C}_i$ being convex and compact polytopic sets defined as boxes (see 3.53).

The main goal is to show that this problem can be recast as an instance of Problem (7.1). With this aim, we first notice that, according to the previously stated FF-CG methodology, an aggregate vector $g(t)$, solution of the above stated problem, could be computed every τ steps by solving the following multi-objective optimization problem

$$\begin{aligned} &\min_{g=[g_1^T,\ldots,g_N^T]^T} [\| g_1 - r_1 \|^2_{\Psi_1}, \ldots, \| g_i - r_i \|^2_{\Psi_i}, \ldots, \| g_N - r_N \|^2_{\Psi_N}] \\ &subject\ to \begin{cases} g \in \mathcal{W}_\delta \\ (g - g(t-\tau)) \in \Delta\mathcal{G}(g(t-\tau)) \end{cases} \end{aligned} \tag{7.12}$$

where $\Delta\mathcal{G}(g)$ has been already defined in (5.5). By following the same procedure used to derive Problem (3.63) we can recast Problem (7.12) into the following explicit formulation

$$\min_g[\| g_1 - r_1 \|^2_{\Psi_1}, \ldots, \| g_i - r_i \|^2_{\Psi_i}, \ldots, \| g_N - r_N \|^2_{\Psi_N}]$$
$$s.t. \begin{cases} (S_{c,j})\, g \le \bar{q}_j^{k_\varepsilon} - (\varepsilon + \delta) \\ (-S_{c,j})\, g \le -\underline{q}_j^{k_\varepsilon} - (\varepsilon + \delta) \\ (R^k_{c,j} + S_{c,j})g - R^k_{c,j} g(t-\tau) \le \bar{q}_j^{k_\varepsilon} - (\varepsilon + \delta) - \gamma \rho^{(j)}_{g(t-\tau)} \\ (-R^k_{c,j} + S_{c,j})g + R^k_{c,j} g(t-\tau) \le \bar{q}_j^{k_\varepsilon} - (\varepsilon + \delta) - \gamma \rho^{(j)}_{g(t-\tau)} \\ (R^k_{c,j} - S_{c,j})g - R^k_{c,j} g(t-\tau) \le -\underline{q}_j^{k_\varepsilon} - (\varepsilon + \delta) - \gamma \rho^{(j)}_{g(t-\tau)} \\ (-R^k_{c,j} - S_{c,j})g + R^k_{c,j} g(t-\tau) \le -\underline{q}_j^{k_\varepsilon} - (\varepsilon + \delta) - \gamma \rho^{(j)}_{g(t-\tau)}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j = 1, ..., n_c,\ k = 0, ..., k_0 \end{cases} \tag{7.13}$$

or equivalently into a compact form

$$\begin{array}{c} \min_g[\| g_1 - r_1 \|^2_{\Psi_1}, \ldots, \| g_i - r_i \|^2_{\Psi_i}, \ldots, \| g_N - r_N \|^2_{\Psi_N}] \\ s.t. Ag - b(g(t-\tau)) \le 0 \end{array} \tag{7.14}$$

where

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_j \\ \vdots \\ A_{n_c} \end{bmatrix} \in \mathbb{R}^{(2+4k_0)n_c \times m}, b(g(t-\tau)) = \begin{bmatrix} b_1(g(t-\tau)) \\ \vdots \\ b_j(g(t-\tau)) \\ \vdots \\ b_{n_c}(g(t-\tau)) \end{bmatrix} \in \mathbb{R}^{(2+4k_0)n_c} \tag{7.15}$$

with

$$A_j = \begin{bmatrix} S_{c,j} \\ -S_{c,j} \\ R^1_{c,j} + S_{c,j} \\ -R^1_{c,j} + S_{c,j} \\ R^1_{c,j} - S_{c,j} \\ -R^1_{c,j} - S_{c,j} \\ \vdots \\ R^{k_0}_{c,j} + S_{c,j} \\ -R^{k_0}_{c,j} + S_{c,j} \\ R^{k_0}_{c,j} - S_{c,j} \\ -R^{k_0}_{c,j} - S_{c,j} \end{bmatrix} \in \mathbb{R}^{(2+4k_0)\times m},$$

$$b_j(g(t-\tau)) = \begin{bmatrix} \bar{q}^{k_\varepsilon}_j - (\varepsilon+\delta) \\ -\underline{q}^{k_\varepsilon}_j - (\varepsilon+\delta) \\ \bar{q}^{k_\varepsilon}_j - (\varepsilon+\delta) - \gamma\rho^{(j)}_{g(t-\tau)} + R^1_{c,j}g(t-\tau) \\ \bar{q}^{k_\varepsilon}_j - (\varepsilon+\delta) - \gamma\rho^{(j)}_{g(t-\tau)} - R^1_{c,j}g(t-\tau) \\ -\underline{q}^{k_\varepsilon}_j - (\varepsilon+\delta) - \gamma\rho^{(j)}_{g(t-\tau)} + R^1_{c,j}g(t-\tau) \\ -\underline{q}^{k_\varepsilon}_j - (\varepsilon+\delta) - \gamma\rho^{(j)}_{g(t-\tau)} - R^1_{c,j}g(t-\tau) \\ \vdots \\ \bar{q}^{k_{0\varepsilon}}_j - (\varepsilon+\delta) - \gamma\rho^{(j)}_{g(t-\tau)} + R^{k_0}_{c,j}g(t-\tau) \\ \bar{q}^{k_{0\varepsilon}}_j - (\varepsilon+\delta) - \gamma\rho^{(j)}_{g(t-\tau)} - R^{k_0}_{c,j}g(t-\tau) \\ -\underline{q}^{k_{0\varepsilon}}_j - (\varepsilon+\delta) - \gamma\rho^{(j)}_{g(t-\tau)} + R^{k_0}_{c,j}g(t-\tau) \\ -\underline{q}^{k_{0\varepsilon}}_j - (\varepsilon+\delta) - \gamma\rho^{(j)}_{g(t-\tau)} - R^{k_0}_{c,j}g(t-\tau) \end{bmatrix} \in \mathbb{R}^{(2+4k_0)} \tag{7.16}$$

where, according to (3.62),

$$\rho^{(j)}_{g(t-\tau)} = min\left\{S_{c,j}g - \underline{q}^{k_\varepsilon}_i, \bar{q}^{k_\varepsilon}_i - S_{c,j}g\right\} \tag{7.17}$$

Notice that (7.14) represents a specialized version of Problem (7.1) with $f_i(g_i) = \|g_i - r_i\|_{\Psi_i}, \forall i \in \mathcal{A}$ and $h(g) = Ag - b(g(t-\tau))$. In particular, in this case, the Neighborhood of an agent could be redefined as

Definition 7.16. *(**Neighborhood of i-th agent**) For the i-th agent the neighborhood is defined as* $\mathcal{N}_i = \{j:$ it exists a line $a = [a_1, ..., a_N]$ in A such that $a_i \neq 0_{m_i}$ and $a_j \neq 0_{m_j}\}$.

It remains to discuss, starting from (7.14), how to derive for each agent a local problem in the form of (7.3). To this end, it is sufficient to associate to each agent the functional involving its local command g_i from the vector of objective functions in (7.14) and decompose the matrices A and b in (7.14) so as to collect, in the local formulation, only the constraints involving the agent i and its neighbours. Then, the following local problem for the i-th agent results

$$\begin{array}{c} \min_{g_i} \parallel g_i - r_i \parallel^2_{\Psi_i} \\ s.t. \\ A^i[g_1^i, ..., g_i ..., g_{|\mathcal{N}_i|}^i] - b^i([g_1^i(t-\tau), ..., g_i(t-\tau) ..., g_{|\mathcal{N}_i|}^i(t-\tau)]) \leq 0 \end{array} \tag{7.18}$$

where A^i is a matrix that collects all lines $a_h \in \mathbb{R}^m$ of A whose elements related to g_i are no zero. Each g_j^i in the vector $[g_1^i, ..., g_i ..., g_{|\mathcal{N}_i|}^i]$ is set to a fixed value and represents the decision variable of agent j-th belonging to $\mathcal{N}_i$. The vector $b^i(g(t-\tau))$ collects all components $b_h([g_1^i(t-\tau), ..., g_i(t-\tau) ..., g_{|\mathcal{N}_i|}^i(t-\tau)])$ of $b(g(t-\tau))$ related to each line a_h in A^i. Notice that, although $b(g(t-\tau))$ requires the entire aggregate vector $g(t-\tau)$ to be computed, its structure allows one to compute $b^i([g_1^i(t-\tau), ..., g_i(t-\tau) ..., g_{|\mathcal{N}_i|}^i(t-\tau)])$ by means of, at most, $g_j(t-\tau)_i$ and $g_i(t-\tau)$ only. In fact the h-th line of A related with $\{g_j(t-\tau)\}_i$ and $g_i(t-\tau)$ is such that the i-th and j-th components of $S_{c,h}$ or $R_{c,h}^k, k > 0$ are not zero. Then, in this case, for the computation of the groups of lines of $b(g(t-\tau))$, related to $g_j(t-\tau)_i$ and $g_i(t-\tau)$ the entire vector $g(t-\tau)$ is not required.

Notice that (7.18) represents a specialized version of Problem (7.3) with $f_i(g_i) = \|g_i - r_i\|_{\Psi_i}$, $h_i(g) = A^i[g_1^i, ..., g_i ..., g_{|\mathcal{N}_i|}^i] - b^i([g_1^i(t-\tau), ..., g_i(t-\tau) ..., g_{|\mathcal{N}_i|}^i(t-\tau)])$ and $[g]_i = \{g_1^i, ..., g_{|\mathcal{N}_i|}^i\}$. Finally, the following DI-FFCG algorithm to be executed at each time instant by each agent can be presented

Algorithm 7.2.1 ***Distributed Iterative FFCG Algorithm (DI-FFCG)*** *- Agent i*

AT EACH TIME t

1.1 IF($t == \kappa\tau, \kappa = 0, 1, \ldots$)

1.1.1 RECEIVE $g_j(t-\tau)$ FOR EACH $j \in \mathcal{N}_i$

1.1.2 COMPUTE $g_i(t)$ BY MEANS OF ALGORITHM 7.1.1 CUSTOMIZED FOR PROBLEMS (7.18)

1.1.3 APPLY $g_i(t)$

1.2 ELSE

1.2.1 APPLY $g_i(t) = g_i(t-1)$

1.3 TRANSMIT $g_i(t)$ TO $\mathcal{N}_i$

It is possible to prove that, under the assumption that a feasible solutions $g(0)$ complying with (7.14) exists at time $t = 0$ (which also implies constraints specification and disturbance set $\mathcal{D}$ which make $\mathcal{W}_\delta$ and $\mathcal{G}(g(0))$ non-empty), the above DI-FFCG scheme enjoys the following properties:

Theorem 7.17. *Let assumptions* **A1**,**A2**,**A4**,**A5**,**A6** *be fulfilled for the system arising from the composition of N subsystems in form (6.1). Let consider the distributed* ***DI-FFCG*** *Algorithm 7.2.1 and let an admissible aggregate command signal $g(0) = [g_1^T(0), \ldots, g_N^T(0)]^T \in \mathcal{W}_\delta$ be applied at $t = 0$ such that (3.8) holds true. Then*

1) for each agent $i \in \mathcal{A}$, at each time $t = \kappa\tau, \kappa \in \mathbb{Z}_+$, the minimizer in (7.18) uniquely exists and can be obtained by locally solving the Algorithm 7.1.1.

*2) the overall system acted by the agents implementing the **DI-FFCG** policy never violates the constraints, i.e. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$;*

3) whenever $r(t) \equiv [\ r_1^T, \ldots, r_N^T\ ]^T, \forall t$, with r_i a constant set-point, the sequence of solutions $g(t) = [g_1^T(t), \ldots, g_N^T(t)]^T$ converges in finite time to a Pareto-Optimal stationary (constant) solution of (5.9) , which is given by r whenever $r \in \mathcal{W}_\delta$, or by any other Pareto-Optimal solution $\hat{r} \in \mathcal{W}_\delta$ otherwise.

Proof

1) The existence of an admissible solution for each agent at each time $\kappa\tau$ can be proved by simply remarking that $g_i(t) = g_i(t-\tau)$, is always an admissible, although not necessarily the optimal, solution for the prescribed problem at time t.
2) At each time $t = \kappa\tau$, with $\kappa \in \mathbb{Z}_+$, from a centralized point of view, the command $g(\kappa\tau)$ computed by means of Algorithm 7.1.1 satisfies the constraints of problem (7.2) because of Proposition (7.14). As consequence, a command $g(\kappa\tau)$ fulfilling (3.19) is applied to the overall plant. Then, the statement can be proved by following the arguments addressed in item 2) of the proof of Theorem 5.7.
3) The stated convergence property follows simply because the sequences of solutions $g_i(t)$ makes the sequences of local costs $||g_i(t) - r_i||^2_{\Psi_i}$ non increasing as already explained for others described distributed techniques. Also in this case, we show that any stationary optimal solution, viz. $g(t) = g(t+1)\ \forall t$, is Pareto Optimal for problem 5.9 by proving that a solution to Problem 5.9 is stationary only if it is Pareto-Optimal.
To this end, let $g(t') = [g_1^T(t'), ..., g_N'^T(t')]^T$ be the actual solution at time $t' = \kappa'\tau, \kappa' \in \mathbb{Z}_+$ which is assumed to be not Pareto-Optimal in $\mathcal{W}_\delta$. As a second step, it useful to consider the set of all admissible solutions at time $t' + \tau$ for problem 7.2 as those contained in the intersection $\mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\}$. Then, two possible situations have to be considered at time $t' + \tau$
   1. if $g(t')$ is not Pareto-Optimal in $\mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\}$, the agents, by performing Algorithm 7.1.1, will converge to a new solutions that will be Pareto Optimal for problem 7.2 and $g(t' + \tau) \neq g(t')$.
   2. if $g(t')$ is Pareto-Optimal in $\mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\}$ there is not reasons for the agents to change their commands, then $g(t' + \tau) = g(t')$. But we show that the Pareto optimality of $g(t') \in \mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\}$ is in contradiction with the fact that $g(t')$ is not Pareto Optimal in $\mathcal{W}_\delta$. In fact, if $g(t')$ is not Pareto Optimal in $\mathcal{W}_\delta$ at least one vector $v = [v_1^T, ..., v_N^T]^T \in \mathbb{R}^m$ would exist with $g(t') + v \in \mathcal{W}_\delta$, such that

$$||g_i(t') + v_i - r_i||^2_{\Psi_i} - ||g_i(t') - r_i||^2_{\Psi_i} \leq 0, \tag{7.19}$$

   happens to hold for all $i \in \mathcal{A}' := \{i \in \mathcal{A} : v_i \neq 0\}$ with some of the above inequalities becoming strict for at least one index $i \in \mathcal{A}'$.

As already discussed in item 3) of the proof of Theorem 5.7 and in others similar previous proofs, by means of straightforward algebraic manipulations, (7.19) leads to

$$||g_i(t') + \alpha v_i - r_i||^2_{\Psi_i} - ||g_i(t') - r_i||^2_{\Psi_i} < 0, \forall \alpha \in (0,1), \forall i \in \mathcal{A}' \quad (7.20)$$

that may be interpreted as the fact that if the above admissible direction v did exist at $g(t')$, **for each** agent $i \in \mathcal{A}'$ it would be strictly convenient to move to $g_i(t') + \alpha v_i$, for a suitable value of α, from its previous solution $g_i(t')$. Since $g(t')$ is Pareto Optimal in $\mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\}$ all the solutions $g(t') + \alpha v$ satisfying (7.20) would not belong to $\mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\}$, i.e.

$$g(t') + \alpha v \notin \mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\}, \forall \alpha \in (0,1) \quad (7.21)$$

The latter represents a contradiction because, since Proposition 3.5, $\Delta\mathcal{G}(g(t'))$ contains a ball of finite radius centered at 0_m. Then

$$\mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\} \supseteq \mathcal{W}_\delta \cap \mathcal{B}^\eta_{g(t')} \quad (7.22)$$

holds where $\mathcal{B}^\eta_{g(t')}$ is a ball of finite radius centered at $g(t')$. As a consequence, for each $v \in \mathbb{R}^m$ such that $g(t') + v \in \mathcal{W}_\delta$ it necessarily exists an $\alpha' \in (0,1)$ such that

$$g(t') + \alpha' v \in \mathcal{W}_\delta \cap \mathcal{B}^\eta_{g(t')}, \forall \alpha \in (0,1) \quad (7.23)$$

that is not compatible with the Pareto Optimality of $g(t')$ in $\mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\}$. Then, the no Pareto Optimality of $g(t')$ in $\mathcal{W}_\delta$ implies the no Pareto Optimality of $g(t')$ in $\mathcal{W}_\delta \cap \{g : (g - g(t')) \in \Delta\mathcal{G}(g(t'))\}$. Hence $g(t')$ cannot be a stationary point.

The previous analysis then suggests us that

$$g(t) \to \hat{r} \quad (7.24)$$

with $\hat{r}$ a generic Pareto Optimum in $\mathcal{W}_\delta$. In order to prove that $\hat{r}$ is reached in a finite time, it is useful to remind that for a finite positive scalar η, because of Proposition (3.5), $\Delta\mathcal{G}(g) \supseteq \mathcal{B}^\eta, \forall g \in \mathcal{W}_\delta$. Moreover, (7.24) ensures that, for a positive η, it exists a finite time t_η such that

$$g(t + t_\eta) - \hat{r} \in \mathcal{B}^\eta, \forall t \geq t_\eta \quad (7.25)$$

Then, at time $t + t_\eta + \tau$

$$\hat{r} \in \mathcal{W}_\delta \cap \Delta\mathcal{G}(g(t + t_\eta)) \quad (7.26)$$

and the procedure 7.1.1 can produce a solution arbitrarily close to $\hat{r}$ (see Proposition 7.14).

7.3 CG based distributed iterative Schemes (DI-CG)

In this section we discuss how to customize the procedure presented in Section 7.1 to address distributed CG design problems described in Chapter 2. The problem that we want to consider has already been introduced at the beginning of previous section and consisting in locally determining, at each time instant t and for each master agent $i \in \mathcal{A}$ associated to each subsystem, a suitable reference signal $g_i(t)$ for system (6.1) which is the best feasible approximation of $r_i(t)$ and such that its application never produces constraints violation, *i.e.* $c_i(t) \in \mathcal{C}^i, \forall t \in \mathbb{Z}_+, i \in \mathcal{A}$, with $\mathcal{C}_i$ being convex and compact polytopic sets.

By repeating arguments used for the FF-CG approach the aggregate vector $g(t)$ can be computed at each time t by solving the following multi-objective optimization problem

$$\begin{array}{c} \min\limits_{g=[g_1^T,\ldots,g_N^T]^T} [\| g_1 - r_1 \|_{\Psi_1}^2, \ldots, \| g_i - r_i \|_{\Psi_i}^2, \ldots, \| g_N - r_N \|_{\Psi_N}^2] \\ subject\ to\ \ g \in \mathcal{V}(x(t)) \end{array} \tag{7.27}$$

where $\mathcal{V}(x) := \mathcal{V}(x,0)$, with $\mathcal{V}(x,0)$ already defined in (6.3) and $\mathcal{C} = \mathcal{C}_1 \times, ..., \times \mathcal{C}_{n_c}$ consisting of polyhedral constraints. Again, the main goal here is to show how this problem could be recast in an instance of Problem (7.1).

To this end, it is sufficient to specialize the CG centralized problem (7.27) to the case where the decision variable is no longer the entire command vector $g \in \mathbb{R}^m$ but a subvector $g_i \in \mathbb{R}^{m_i}$ of it, related to agent i. Then, Problem (7.27) can be rephrased, by means of the procedure (2.34)-(2.51) given in Section 2.3.2, as

$$\begin{array}{l} \min\limits_{g=[g_1^T,\ldots,g_N^T]^T} [\| g_1 - r_1 \|_{\Psi_1}^2, \ldots, \| g_i - r_i \|_{\Psi_i}^2, \ldots, \| g_N - r_N \|_{\Psi_N}^2] \\ \text{subject to } \begin{cases} TH_c\Phi^k x(t) + TR_k^c g \leq q^k, \ k = 0, ..., k_0 \\ T\left(H_c(I-\Phi)^{-1}G + L\right) g \leq q^{k_\varepsilon} - (\varepsilon + \delta)[\sqrt{T_j^T T_j}] \end{cases} \end{array} \tag{7.28}$$

where $[\sqrt{T_j^T T_j}]$ is defined in (2.24), k_0 is determined according to Algorithm (2.3.2), or equivalently into a compact form

$$\begin{array}{c} \min_g [\| g_1 - r_1 \|_{\Psi_1}^2, \ldots, \| g_i - r_i \|_{\Psi_i}^2, \ldots, \| g_N - r_N \|_{\Psi_N}^2] \\ s.t. \ \ Ag + \tilde{A}x(t) - b \leq 0 \end{array} \tag{7.29}$$

where

$$\begin{array}{c} A = \begin{bmatrix} TR_k^c \\ T\left(H_c(I-\Phi)^{-1}G + L\right) \end{bmatrix} \in \mathbb{R}^{(1+k_0)z \times m}, \ \tilde{A} = \begin{bmatrix} TH_c\Phi^k \\ 0_{z \times n} \end{bmatrix} \in \mathbb{R}^{(1+k_0)z \times n} \\ b = \begin{bmatrix} q^k \\ q^{k_\varepsilon} \end{bmatrix} \in \mathbb{R}^{(1+k_0)z}. \end{array} \tag{7.30}$$

Problem (7.29) represents a specialized version of Problem (7.1). In particular, in this case, the Neighborhood of an agent could be redefined as

Definition 7.18. *(**Neighborhood of i-th agent**) For the i-th agent the neighborhood is defined as* $\mathcal{N}_i = \{j:$ it exists a line $a = [a_1, ..., a_i..., a_N]$, in A *or a line* $\tilde{a} = [\tilde{a}_1, ..., \tilde{a}_i..., \tilde{a}_N]$, *in* $\tilde{A}$ *such that* $\big(a^i \neq 0_{m_i}$ and $a^j \neq 0_{m_j}$ *or* $\tilde{a}^j \neq 0_{m_j}\big)\}$.

Unlike the DI-FFCG approach of the previous Section, in this case, the neighboring agents are required to share the local state $x_i(t)$. Such an additional duty does not involve any further modification to Algorithm (7.1.1) because, as we show hereafter, the state $x_i(t)$ represents a simple constant parameter for the local optimization. In order to achieve, for each agent, local problems of the form (7.3) starting from (7.29), it is sufficient to associate to each agent the function involving its local command g_i from the vector objective functions in (7.29) and decompose the matrices A $\tilde{A}$ and b in (7.29) so as to collect in the local formulations only the constraints involving agent i and its neighbours, with the related local commands and states. Then, the local problem for the generic i-th agent is given by

$$\begin{array}{c} \min_{g_i} \parallel g_i - r_i \parallel^2_{\Psi_i} \\ s.t. \;\; A^i[g^i_1, ..., g_i..., g^i_{|\mathcal{N}_i|}] + \tilde{A}^i[x^i_1(t), ..., x_i(t)..., x^i_{|\mathcal{N}_i|}(t)] - b^i \leq 0 \end{array} \tag{7.31}$$

where A^i $\tilde{A}^i$ and b^i are built by means of the simple following rule: the h-th lines a_h of A, $\tilde{a}_h$ of $\tilde{A}$ and b_h of b appear in A^i, $\tilde{A}^i$ and b^i respectively if the i-th component of a_h is not zero. g^i_j is the decision variable of the agent j-th belonging to $\mathcal{N}_i$, which, in the local problem of the i-th agent, is a constant term. Notice that also Problem 7.31 is in the class of Problems (7.3). The above described formulation allows us to present the DI-CG algorithm to be performed at each time instant by each agent.

Algorithm 7.3.1 ***Distributed Iterative CG Algorithm (DI-CG)*** *- Agent i*

AT EACH TIME t

1.1 RECEIVE $g_j(t-1)$, $x_j(t-1)$ FROM EACH $j \in \mathcal{N}_i$

1.2 COMPUTE $g_i(t)$ BY MEANS OF ALGORITHM 7.1.1 CUSTOMIZED FOR PROBLEMS (7.31)

1.3 APPLY $g_i(t)$

1.4 TRANSMIT $g_i(t)$ AND $x_i(t)$ TO $\mathcal{N}_i$

Finally, the following properties can be shown to hold for the above stated DI-CG scheme

Theorem 7.19. *Let assumptions* **A1**,**A2**,**A4**,**A5**,**A6** *be fulfilled for the system arising from the composition of N subsystems in form (6.1). Let consider the distributed* ***DI-CG*** *Algorithm 7.3.1 and let* $\mathcal{V}(x(0), d_{max})$ *be non empty at time* $t = 0$. *Then*

1) for each agent $i \in \mathcal{A}$, at each time $t, t \in \mathbb{Z}_+$, the minimizer in (7.31) uniquely exists and can be obtained by locally solving the Algorithm 7.1.1.
*2) the overall system acted by the agents implementing the **DI-CG** policy never violates the constraints, i.e. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$;*
3) whenever $r(t) \equiv [\ r_1^T, \ldots, r_N^T\ ]^T, \forall t$, with r_i a constant set-point, the sequence of solutions $g(t) = [g_1^T(t), \ldots, g_N^T(t)]^T$ in a finite time converges to a Pareto-Optimal stationary (constant) solution of (5.9), which is given by r whenever $r \in \mathcal{W}_\delta$, or by any other Pareto-Optimal solution $\hat{r} \in \mathcal{W}_\delta$ otherwise.

Proof

1) The existence of an admissible solution for each agent at each time t can be proved by simply remarking that $g_i(t) = g_i(t-1)$, is always an admissible, although not necessarily the optimal, solution for the prescribed problem at time t.
2) At each time t, with $t \in \mathbb{Z}_+$, from a centralized point of view, the command $g(t)$ computed by means of Algorithm 7.1.1 satisfies the constraints of problem (7.3) because of Proposition (7.14). As a consequence, a command $g(t)$ belonging to $\mathcal{V}(x(t))$ is applied to the overall plant. Then, the statement can be proved by following the arguments used in the proof of item 2) of Theorem 6.2.
3) The stated convergence property follows simply because the sequence of solutions $g_i(t)$ makes the sequence of local costs $||g_i(t) - r_i||^2_{\Psi_i}$ non increasing as already explained for other described distributed techniques. Also in this case, we show that any stationary optimal solution, viz. $g(t) = g(t+1)\ \forall t$, is Pareto Optimal for problem 5.9 by proving that a solution to Problem 5.9 is not stationary if it is not Pareto-Optimal.
   To this end, let $g(t') = [g_1^T(t'), ..., g_N'^T(t')]^T$ be the actual solution at time $t' \in \mathbb{Z}_+$ which is assumed to be not Pareto-Optimal in $\mathcal{W}_\delta$. Then, two possible situations have to be considered at time $t' + 1$
   1. if $g(t')$ is not Pareto-Optimal in $\mathcal{V}(x(t'))$, the agents, by performing Algorithm 7.1.1, will converge on a new solution that is Pareto Optimal for the problem 7.3 and $g(t'+1) \neq g(t')$.
   2. if $g(t')$ is Pareto-Optimal in $\mathcal{V}(x(t'))$ there is not reasons for the agents to change their commands. Then $g(t'+1) = g(t'), \forall t \geq t'$. But we show that if $g(t')$ is not Pareto Optimal in $\mathcal{W}_\delta$ such a situation can reoccur a finite number of times. In fact, the proof of Lemma 6.1 and item 3) of the proof of Theorem 6.2 ensure that after a finite t_s

$$\mathcal{V}(x(t'+t_s)) \supseteq \mathcal{W}_\delta \cap \mathcal{B}^\varrho_{g(t'+t_s)}, \forall t \geq t'+t_s, \varrho > 0 \tag{7.32}$$

   where $\mathcal{B}^\varrho_{g(t'+t_s)}$ is a ball centered at $g(t'+t_s)$ with a finite radius $\varrho > 0$. Then, if at time $t'+t_s$, $g(t'+t_s) = g(t')$ is not Pareto Optimal in $\mathcal{W}_\delta$, at least one vector $v = [v_1^T, ..., v_N^T]^T \in \mathbb{R}^m$ would exist with $g(t')+v \in \mathcal{W}_\delta$, such that

$$||g_i(t') + v_i - r_i||^2_{\Psi_i} - ||g_i(t') - r_i||^2_{\Psi_i} \leq 0, \tag{7.33}$$

happens to hold for all $i \in \mathcal{A}' := \{i \in \mathcal{A} : v_i \neq 0\}$ with some of the above inequalities becoming strict for at least one index $i \in \mathcal{A}'$. As already discussed in item 3) of the proof of Theorem 7.19, by means of straightforward algebraic manipulations, the (7.19) leads to

$$||g_i(t') + \alpha v_i - r_i||^2_{\Psi_i} - ||g_i(t') - r_i||^2_{\Psi_i} < 0, \forall \alpha \in (0,1), \forall i \in \mathcal{A}' \tag{7.34}$$

that may be interpreted as the fact that if the above admissible direction v did exist at $g(t')$, **for each** agent $i \in \mathcal{A}'$ it would be strictly convenient to move to $g_i(t') + \alpha v_i$, for a suitable value of α, from its previous solution $g_i(t')$. Since $g(t')$ is Pareto Optimal in $\mathcal{V}(x(t' + ts))$ all the solutions $g(t') + \alpha v$ satisfying (7.34) would not belong to $\mathcal{V}(x(t' + ts))$, i.e.

$$g(t') + \alpha v \notin \mathcal{V}(x(t' + ts)), \forall \alpha \in (0,1) \tag{7.35}$$

The latter represents a contradiction because, since (7.32), for each $v \in \mathbb{R}^m$ such that $g(t') + v \in \mathcal{W}_\delta$ it necessarily exists an $\alpha' \in (0,1)$ such that

$$g(t') + \alpha' v \in \mathcal{W}_\delta \cap \mathcal{B}^o_{g(t')}, \forall \alpha \in (0,1) \tag{7.36}$$

that is not compatible with the Pareto Optimality of $g(t')$ in $\mathcal{V}(x(t' + ts))$. Then the no Pareto Optimality of $g(t')$ in $\mathcal{W}_\delta$ implies the no Pareto Optimality of $g(t')$ in $\mathcal{V}(x(t' + ts))$. Hence $g(t')$ cannot be a stationary point.

The previous analysis then suggests us that

$$g(t) \to \hat{r} \tag{7.37}$$

with $\hat{r}$ a generic Pareto Optimum in $\mathcal{W}_\delta$. In order to prove that $\hat{r}$ is reached in a finite time, it is useful to remind that because of properties (4.32) and (4.33) for system (2.1), (7.37) ensures that, it exists a finite time t_δ such that

$$c(t + t_\delta) - c_{\hat{r}} \in \mathcal{B}^\delta, \forall t \geq t_\delta. \tag{7.38}$$

The latter indicates that predictions for $\bar{c}$ along virtual time k, starting from $x(t_\delta)$, will satisfy

$$\bar{c}(k, x(t_\delta), \hat{r}) - c_{\hat{r}} \in \mathcal{B}^\delta, \forall k \geq 0. \tag{7.39}$$

Since $c_{\hat{r}} \in \mathcal{C}_\infty \sim \mathcal{B}^\delta$, if we add $c_{\hat{r}}$ to the right side of the latter we obtain

$$\bar{c}(k, x(t_\delta), \hat{r}) \in \mathcal{C}_\infty \sim \mathcal{B}^\delta \oplus \mathcal{B}^\delta, \forall k \geq 0 \tag{7.40}$$

that becomes

$$\bar{c}(k, x(t_\delta), \hat{r}) \in \mathcal{C}_\infty \sim \mathcal{B}^\delta, \forall k \geq 0. \tag{7.41}$$

Then, (7.41) implies $\hat{r} \in \mathcal{V}(x(t_\delta))$ or equivalently that at time t_δ it will be an admissible solution for Problem (7.27). Then, the procedure 7.1.1 can produce a solution arbitrarily close to $\hat{r}$.

Remark 7.20. It is worth pointing out that, unlike the non-iterative methods of Chapters 5 and 6, the DI-CG and DI-FFCG strategies guarantee at each iteration possibly unfeasible solutions which are Pareto Optimal besides the convergence to a feasible Pareto Optimum in finite time. However, according to Proposition 7.14, if the iterative procedure 7.1.1 is stopped before the convergence is reached, there is not any guarantee that a feasible solution is provided. Such a limitation represents the main drawback of the described iterative methods.

7.4 Conclusions

In this Chapter a distributed iterative optimization method borrowed from [60] has been discussed and customized to develop iterative CG and FF-CG distributed methods. All technicalities required to adapt the discussed optimization method to the CG and FF-CG design problems have been provided. Furthermore, we have proved that the resulting DI-CG and DI-FFCG strategies hold similar properties w.r.t centralized CG and FF-CG approaches.

8

Case Study: An eight-tank water distribution system application

Water distribution systems consist essentially of pumping units, water tanks and distribution pipes connecting them. Many operational and coordination constraints may also be present and such large-scale dynamically coupled interconnected systems are a benchmark to check the applicability of distributed supervision strategies aiming at actively coordinating the action of the pumping units in achieving global goals in response to changed conditions.

In literature, several results have been published on the control of water distribution systems with solutions based both on centralized ([64, 65, 66, 67]) and distributed ([68, 69]) MPC approaches. These distributed schemes are based on optimization procedures which require several iteration steps which, under mild assumptions (e.g. linearity of the subsystem models and convexity of the optimization problems to be solved), allow the achievement of control performance similar to those achievable by centralized solutions. The main drawback of such iterative schemes lies in the huge amount of iterations required, leading to decision times remarkably larger than the typically sampling periods used in such control applications. In order to overcome this problem, in ([70]) a non-iterative distributed MPC scheme is proposed for irrigation canals control.

In this Chapter, the distributed supervising strategies described in the previous Chapters are applied to the management of the water level set-points of the downstream tanks of a water network which is assumed to be subject to several pointwise-in-time coordination and operational constraints. Because of the high dimensionality and spatial distribution of the dynamical model, centralized water level management strategies can be infeasible due to the large communication/computational requirements. On the contrary, a distributed management is to be preferred because its action is computed via many distributed and communicating computing devices. The problem is complicated because the coordination constraints can be global and involve variables at far locations in the network. Then, the actions of local devices, typically one for each subsystem, have to be coordinated in order to rigorously fulfil all constraints and attain, in such a distributed implementation, behaviors and

properties which are similar to those usually pertaining to centralized supervisors.

The system here presented seems to be suitable for comparing all the distributed strategies presented in the second part of this thesis. Then, after a brief description of such a system several simulation results are reported which are contrasted with the results of the centralized supervising strategies described in Chapters 2 and 3.

8.1 Model Description

Consider the water tank network depicted in Figure 8.1. The system consists of the interconnection of four cascaded two-tank models. Each cascaded subsystem is described by the following non-linear equations

$$\begin{cases} \rho S_i^1 \dot{h}_i^1 = -\rho A_i^1 \sqrt{2gh_i^1} + u_i \\ \rho S_i^2 \dot{h}_i^2 = -\rho A_i^2 \sqrt{2gh_i^2} + \rho A_i^1 \sqrt{2gh_i^1} + \sum_{j \in \mathcal{S}^i} \rho A_j^1 \sqrt{2gh_i^1} \end{cases} \tag{8.1}$$

where u_i is the water flow supplied by the pump whose command is the

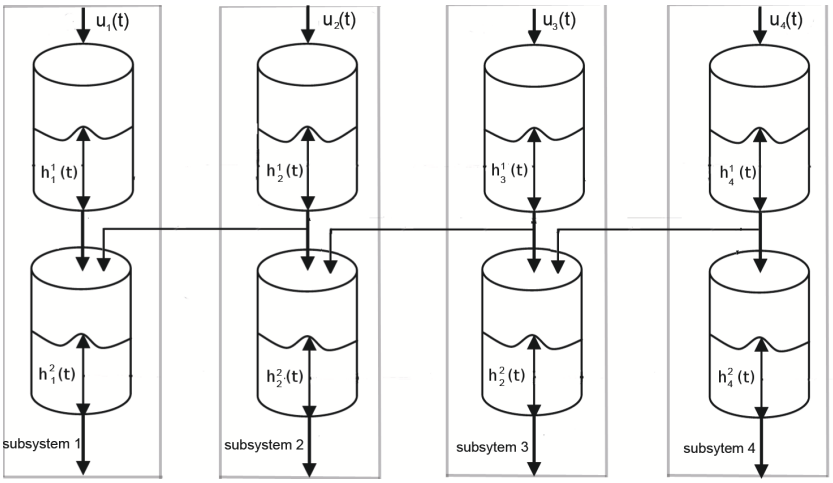

Fig. 8.1. A four cascaded two-tank water system

voltage V_i, $i \in \mathcal{A} := \{1, .., 4\}$. Moreover, for each $q = 1, 2$, S_i^q are the tank sections, h_i^q, the water level in the tanks, A_q^i the section of pipes connecting the tanks, and g and ρ the gravity constant and the water density respectively. Their values are specified in Tables 1-2.

With $\mathcal{S}^i$ we denote the set of subsystems which provide water to the downstream tank of the i-th subsystem; in our case $\mathcal{S}^1 := \{2\}$, $\mathcal{S}^2 = \{3\}$, $\mathcal{S}^3 = \{4\}$

and $\mathcal{S}^4 = \emptyset$. Each cascaded two-tank subsystem has a related *decision maker* or *agent* in charge of regulating the levels $h_i^2(t)$, $i \in \mathcal{A}$ by modifying properly their set-points and by exchanging relevant data with the other agents by means of the communication graph depicted in Figure 8.2. A simple static equation is used to model the relationship between the input voltage $V_i(t)$ and the incoming mass of water

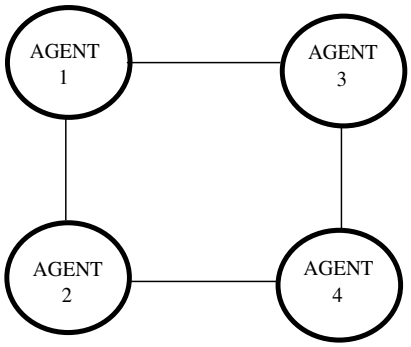

Fig. 8.2. Communication graph

Subsystem 1			
Tank 1	Value	Tank 2	Value
S_1^1	2500 cm^2	S_1^2	2500 cm^2
A_1^1	4 cm^2	A_1^2	8 cm^2
$\overline{h}_i^1$	80 cm	$\overline{h}_i^2$	70 cm
$\underline{h}_i^1$	1 cm	$\underline{h}_i^2$	1 cm
Subsystem $i = \{2, 3, 4\}$			
Tank 1	Value	Tank 2	Value
S_i^1	2500 cm^2	S_i^2	2500 cm^2
A_i^1	8 cm^2	S_i^2	8 cm^2
$\overline{h}_i^1$	80 cm	$\overline{h}_i^2$	70 cm
$\underline{h}_i^1$	1 cm	$\underline{h}_i^2$	1 cm

Table 8.1. Tanks and constraints values

$$u_i(t) = \begin{cases} V_i(t) \; if \; V_i(t) \geq 0 \\ 0 \qquad if \; V(t) < 0 \end{cases} \tag{8.2}$$

The following local and global constraints are to be enforced at each time instant

Parameters	Value
g	980 $cm/(sec^2)$
ρ	$10^{(-3)}$ $Kg/(cm^3)$
V_{max}	4
T_c	$0.8sec$

Table 8.2. Parameter values

$$\begin{gathered} \underline{h}_i^1 \leq h_i^1 \leq \overline{h}_i^1, \ \forall \ i \in \mathcal{A}, \\ \underline{h}_i^2 \leq h_i^2 \leq \overline{h}_i^2, \ \forall \ i \in \mathcal{A}, \\ 0 \leq V_i \leq V_{max}, \ \forall \ i \in \mathcal{A}, \\ |h_1^1 - h_2^1| <= 5 \ cm, \ |h_2^1 - h_3^1| <= 5 \ cm, \ |h_3^1 - h_4^2| <= 5 \ cm \end{gathered} \tag{8.3}$$

The system is linearized around the equilibrium $\bar{V}_i = \bar{u}_i^{eq} = 2, \ i \in \mathcal{A} \ \bar{h}_i^j = 32cm$ and discretized with sampling time $T_c = 0.8 \ sec$. Local decentralized tracking LQ output feedback controllers ([71]) are implemented, which act properly on the incoming water flows $u_i(t)$, in such a way that the offset property **A2** is satisfied. The resulting equation of the overall pre-compensated system has the same structure as in (2.1) where, in this case, the commands $g(t) \in \mathbb{R}^4$ represents the applied references for the downstream tanks.

$$\Phi = \begin{bmatrix} 0.7237 & -0.1053 & -0.0016 & 0.0044 & 0.0001 & -0.0004 & 0 & 0 \\ 0.0086 & 0.9894 & 0.0043 & -0.0001 & 0 & 0 & 0 & 0 \\ -0.0016 & -0.0504 & 0.7253 & -0.0573 & -0.0009 & 0.0026 & 0 & -0.0003 \\ 0 & -0.0001 & 0.0043 & 0.9897 & 0.0043 & -0.0001 & 0 & 0 \\ 0.0001 & 0.0020 & -0.0009 & -0.0567 & 0.7252 & -0.0568 & -0.0009 & 0.0028 \\ 0 & 0 & 0 & -0.0001 & 0.0043 & 0.9897 & 0.0043 & -0.0001 \\ 0 & -0.0002 & 0 & 0.0025 & -0.0009 & -0.0565 & 0.7252 & -0.0593 \\ 0 & 0 & 0 & 0 & -0.0001 & 0.0043 & 0.9899 & \end{bmatrix}$$

$$G = \begin{bmatrix} 0.3816 & -0.2776 & 0.2734 & -0.2730 \\ 0.0020 & 0.0001 & 0 & 0 \\ 0.0520 & 0.6051 & -0.5487 & 0.5462 \\ 0.0001 & 0.0017 & 0.0001 & 0 \\ -0.0021 & 0.0585 & 0.6045 & -0.5488 \\ 0 & 0.0001 & 0.0017 & 0.0001 \\ 0.0002 & -0.0026 & 0.0583 & 0.6072 \\ 0 & 0 & 0.0001 & 0.0015 \end{bmatrix}$$

$$H_y = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -83.6479 & -33.0636 & -0.5010 & 1.3954 & 0.0209 & -0.1186 & -0.0017 & 0.0141 \\ -0.5010 & -15.8341 & -83.1365 & -17.9865 & -0.2722 & 0.8234 & 0.0121 & -0.0805 \\ 0.0209 & 0.6384 & -0.2722 & -17.8015 & -83.1660 & -17.8298 & -0.2688 & 0.8756 \\ -0.0017 & -0.0522 & 0.0121 & 0.7951 & -0.2688 & -17.7492 & -83.1782 & -18.6248 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 119.8475 & -87.1772 & 85.8588 & -85.7509 \\ 16.3351 & 190.0304 & -172.3228 & 171.5557 \\ -0.6592 & 18.3668 & 189.8685 & -172.3768 \\ 0.0539 & -0.8211 & 18.3128 & 190.6896 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

8.2 Simulation Scenario

The reported simulations investigate the behavior of the overall system when the desired set-points r_i to the water levels of the downstream tanks have the profiles depicted in Figures 8.10-8.13 (red dashed line). At the beginning, the desired references $r_i = 32cm,\ i \in \mathcal{A}$ correspond to an equilibrium. At time $t = 30$ *sec*, the reference r_1 related to the downstream tank of subsystem 1 is changed from 32 cm to 42 cm. At the same time, also the reference r_2 is modified from 32 cm to 34 cm. These values are kept constant until time instant $t = 400$ *sec* when they are changed back to their initial values. Simultaneously, the desired references r_3 and r_4 change their values at time $t = 300$ *sec* from 32 cm to 27.85 cm and, respectively, 28.5 cm. After that, these new values are kept constant up to time $t = 800$ *sec*, when they are brought back to the previous values.

8.3 Agents design and simulation parameters

The supervision of the system in Figure 8.1 is achieved by considering several strategies described in previous Chapters. In particular, we have analyzed the behavior of the system under the action of

- the centralized Standard CG described in Chapter 2,
- the centralized FF-CG of Chapter 3,

- the distributed non-iterative schemes S-FFCG (Chapter 5) and S-CG (Chapter 6),
- the distributed non-iterative schemes P-FFCG (Chapter 5) and P-CG (Chapter 6),
- the distributed iterative schemes DI-FFCG and DI-CG of Chapter 7.

Hereafter, the design parameters used for each considered strategy are reported.

Centralized CG design parameters

The problem to be solved in this case is (2.29) of Chapter 2 where

- $\Psi = I_m$, which implies that the cost to be minimized is $J_c = \|g_1 - r_1\| + \|g_2 - r_2\| + \|g_3 - r_3\| + \|g_4 - r_4\|$.
- $T = [I_n, -I_n]^T$, and

$$q = [0.3, 0.3, 0.3, 0.3, 2, 2, 2, 2, 0.05, 0.05, 0.05, 0.3, 0.3, 0.3, 0.3, 2, 2, 2, 2, 0.05, 0.05, 0.05]^T,$$

- $k_0 = 3$, computed according to algorithm (2.3.1),
- $\delta = 10^{-8}$ and the corresponding set $\mathcal{W}_\delta$ given by

$$\mathcal{W}_\delta : \left\{ g \in \mathbb{R}^4 : \begin{bmatrix} 1 & -1 & 1 & -1 \\ 0 & 2 & -2 & 2 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 2 \\ 3.1360 & -3.1360 & 3.1360 & -3.1360 \\ 0 & 6.2720 & -6.2720 & 6.2720 \\ 0 & 0 & 6.2720 & -6.2720 \\ 0 & 0 & 0 & 6.2720 \\ 1 & -3 & 3 & -3 \\ 0 & 2 & -4 & 4 \\ 0 & 0 & 2 & -4 \\ -1 & 1 & -1 & 1 \\ 0 & -2 & 2 & -2 \\ 0 & 0 & -2 & 2 \\ 0 & 0 & 0 & -2 \\ -3.1360 & 3.1360 & -3.1360 & 3.1360 \\ 0 & -6.2720 & 6.2720 & -6.2720 \\ 0 & 0 & -6.2720 & 6.2720 \\ 0 & 0 & 0 & -6.2720 \\ -1 & 3 & -3 & 3 \\ 0 & -2 & 4 & -4 \\ 0 & 0 & -2 & 4 \end{bmatrix} g \leq q - \delta\sqrt{[T_i^T T_i]} \right\} \tag{8.4}$$

Centralized FFCG design parameters

Algorithm (3.2.2) presented in Chapter 3 is implemented in this case with

- Ψ, k_0, δ and $\mathcal{W}_\delta$ the same as in the previous CG design,
- $\gamma = 0.3$, which implies $\tau = 5$ computed according to Algorithm (3.2.1).

Distributed S-FFCG design parameters

Algorithm (5.2.1) presented in Chapter 5 is implemented here with

- the neighbourhoods for each agent, according to Figure 8.2, are: $\mathcal{A}_1 = \{2,3\}$, $\mathcal{A}_2 = \{1,4\}$, $\mathcal{A}_3 = \{1,4\}$, $\mathcal{A}_4 = \{2,3\}$, $d_{max} = 2$,
- T, q,k_0 δ, γ,τ are the same as in the FFCG design,
- $\mathcal{W}_\delta$ is modified by means of procedure described in section 5.3.1 into $\mathcal{W}'_\delta$ that represents the convex hull of a multi-box inner approximation carried out on $\mathcal{W}_\delta$ (Details are omitted for brevity).

Distributed S-CG design parameters

Algorithm (6.2.1) presented in Chapter 6 is implemented here with the same terms $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, d_{max}, T, q,k_0 δ, and $\mathcal{W}'_\delta$ used in the S-FFCG design.

Distributed P-FFCG design parameters

Algorithm (5.4.1) presented in Chapter 5 is implemented here with

- $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, d_{max}, T, q,k_0 δ, and $\mathcal{W}'_\delta$ used in the S-FFCG design.
- $\epsilon_P = 0.0038$ $\epsilon_S = 0.0042$,

Distributed P-CG design parameters

Algorithm (6.3.1) presented in Chapter 6 is implemented here with the same $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, d_{max}, T, q,k_0 δ, ϵ_P, ϵ_S and $\mathcal{W}'_\delta$ used in the P-FFCG design.

Distributed DI-FFCG design parameters

At each time instant, problem (7.12) is solved by means of Algorithm (7.1.1) with

- $\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{A}_3 = \mathcal{A}_4 = \mathcal{A}$. In fact, in this case, because of the constraints structure (see (8.4)) all agents belong to the same neighborhood,
- T, q,k_0 δ, γ, τ, $\mathcal{W}_\delta$ are the same used in the FFCG design,
- the number of iterations considered for the optimization is $N_{it} = 10$.

Distributed DI-CG design parameters

At each time instant, problem (7.27) is solved by means of Algorithm (7.1.1) with the same $\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{A}_3 = \mathcal{A}_4 = \mathcal{A}$, T, q,k_0 δ, $\mathcal{W}_\delta$ and N_{it} used in the DI-FFCG design.

8.4 Simulation Results

Observe first the constrained vector responses reported in Figures 8.3-8.5 and 8.6-8.9 for all strategies. It is important to note how such a vector violates the constraints at several time instants when no CG unit is operating. It is worth noticing that all centralized solutions (CG and FFCG) and all distributed iterative strategies (DI-FFCG and DI-CG) are more active and the constrained signals are often near to the constraints' boundaries. On the contrary, the distributed non iterative methods present a more conservative behavior.

In order to evaluate and compare the performance related to each considered strategy, we provide the evolutions of the downstream water levels in Figures 8.10-8.13, while in Figures 8.14-8.17 the various actions are reported. Moreover, a further way to compare the relative performance for all methods has been provided in Figure 8.18, where the value of the cost $J = \frac{1}{T}\sum_{k=0}^{T} |r(k) - \theta_L(k)|$ is reported for each method.

From the observation of the above indicated figures the following conclusions arise:

1. **Centralized schemes perform better than distributed schemes**: Among the distributed methods, only DI-CG compares similarly to the centralized methods (see Figure 8.18). On the contrary, all other distributed schemes exhibits, as expected, slower responses to step changing set-points. The standard CG and the FFCG centralized schemes are faster to reach a new optimal equilibrium after a step set-point change than the distributed methods because the latter don't cooperate in reaching the optimal solution. Such a situation can be noticed in Figures 8.14-8.17 after time $t = 200$ when all methods converged. At time $t = 300$, the references $r_3(t)$ and $r_4(t)$ change while $r_1(t)$ and $r_2(t)$ remain constant. This means that $\|g_3(t) - r_3(t)\|$ and $\|g_4(t) - r_4(t)\|$ would increase if g_3 and g_4 were kept constant while $\|g_1(t) - r_1(t)\|$ and $\|g_2(t) - r_2(t)\|$ would remain unchanged. Since the distributed methods are not cooperative, agents 1 and 2 do not modify their local costs because they would worsen it, while agents 3 and 4 start to find new optimal solutions. On the contrary, in the centralized case, at time $t = 300$ the entire vector $g(t)$ is modified because $g(300)$ is no more an optimum solution for J_c. Then, a new optimal equilibrium is reached faster than the distributed competitors.
2. **Iterative methods are better than non-iterative ones**: In fact, the iterative procedure (7.1.1) guarantees that a Pareto Optimal solution is reached at each iteration step while there is not such a guarantee for the non-iterative methods.
3. **CG-based methods are better than FFCG methods**: As expected, in both centralized and distributed schemes, state-based CG methods can compute their action by means of a more accurate system future state predictions because computed on the basis of a direct measure of the actual state. On the contrary, FFCG based methods, besides making use of less

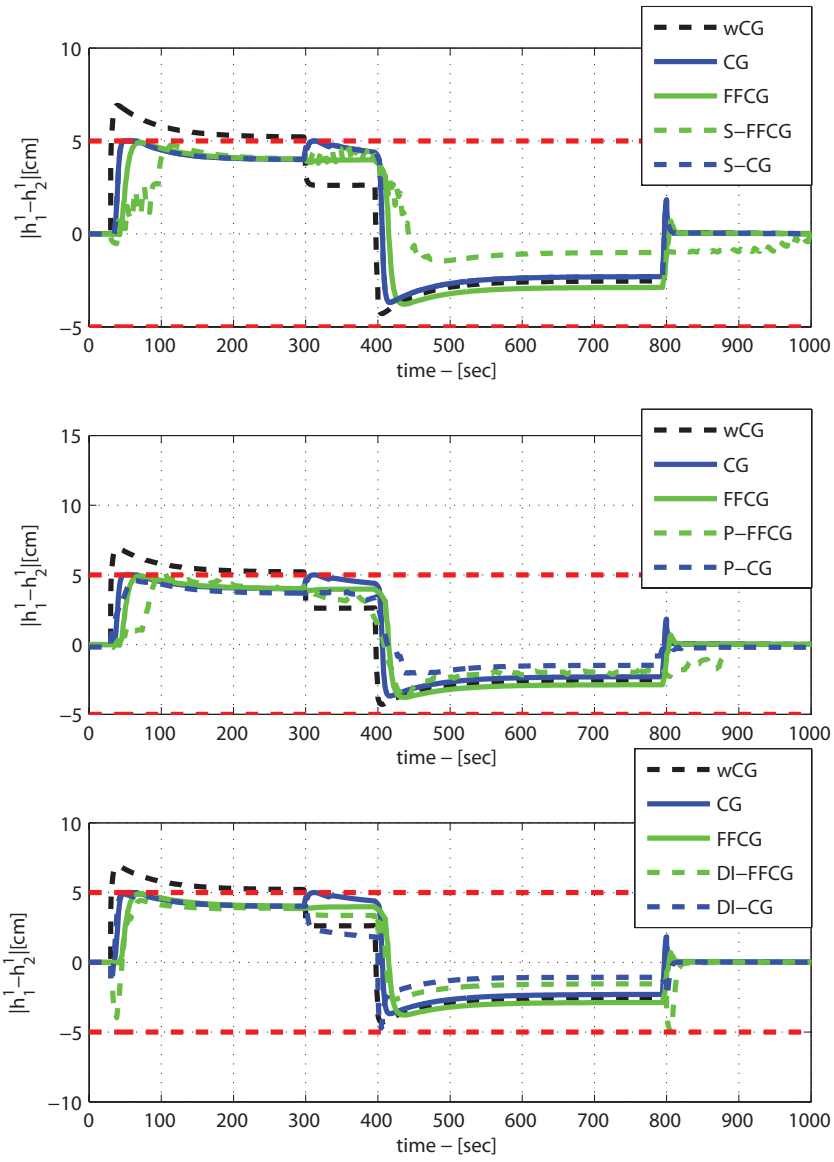

Fig. 8.3. Coordination constraint $|h_1^1(t) - h_2^1(t)|$

accurate future state predictions, update their commands at multiples of the sampling period.

4. **Sequential approaches are worse than parallel approaches**: The increased efficiency of parallel approaches w.r.t. sequential ones, in both

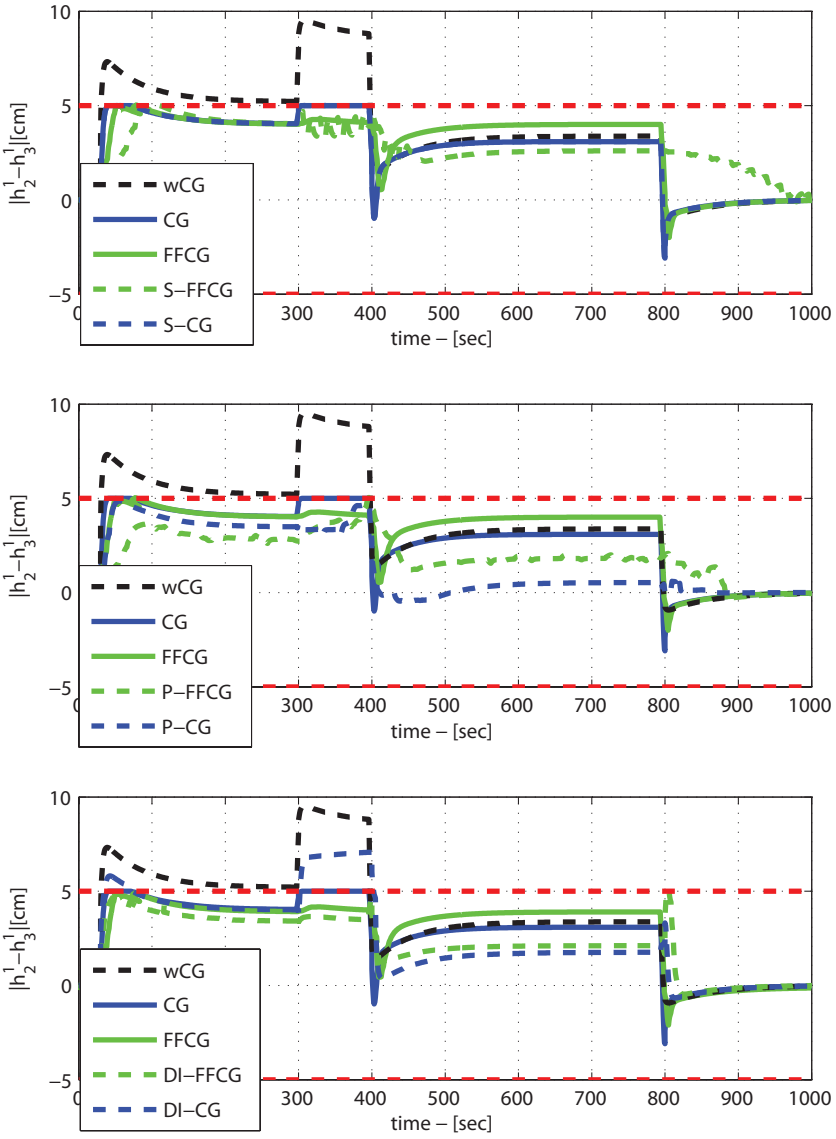

Fig. 8.4. Coordination constraint $|h_2^1(t) - h_3^1(t)|$

CG and FFCG based methods, is remarkably noticeable. Such a gap become larger if the number of agents increases.

Nevertheless, although the performance of a centralized solutions, especially those based on the direct measure of the state, outperform any distributed solution, the difference is modest as it can be observed in Figure 8.18.

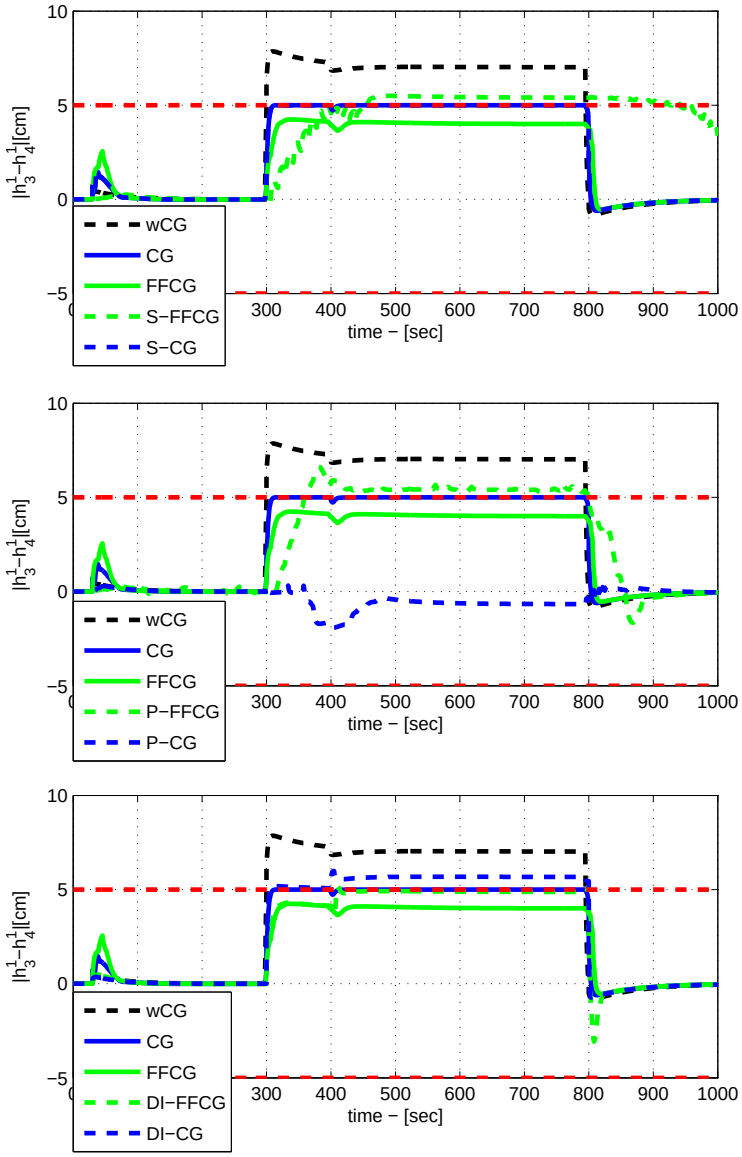

Fig. 8.5. Coordination constraint $|h_3^1(t) - h_4^1(t)|$

As far as the numerical complexity and the data exchange of the methods is concerned, an investigation of the required CPU execution time and the rate of data exchanged (see Figures 8.19 and 8.20) was undertaken for contrasting a single supervisory agent of each distributed schemes with the overall computational burden and data exchanged for the centralized solutions. It has been resulted that a single agent in all non-iterative schemes has a computational burden which is an order of magnitude lower than any centralized CG solution. Furthermore, in this case, it is worth remarking that the iterative methods require a computational time that is larger than the ones pertaining to the centralized methods. Although this result could be surprising, its explanation is quite trivial. In fact, in the iterative methods the optimization problem is iterated until the agents converge to a solution or, as in our case, by a finite number of times ($N_{it} = 10$). Then, the DI-FFCG and DI-CG computational burdens are the summation of the CPU times needed for the resolution of 10 problems and the time required for the corresponding data exchanges amongst the agents at each iteration. Then, the results depicted in Figure 8.19 for the iterative methods are strongly related to the amount of data exchanged depicted in Figure 8.20. Such a figure points out the main drawback of the iterative methods that consists in the huge amount of exchanged data required for the convergence to a solution that is remarkably higher if compared with the centralized and non-iterative methods. Actually, distributed iterative methods could become more competitive in very large-scale applications with weak interconnections between the subsystems (in this case the subnetworks are much smaller than the overall system).

For the sake of completeness concerning the P-FFCG and P-CG strategies, we have included in Figures 8.21 and 8.22 the switching signals amongst the operative scenarios. It can be observed that in both cases, for almost half of the simulation the strategies operate in the sequential mode due to the fact that the command $g(t)$ is kept close to the boundaries of the admissible region $\mathcal{W}'_\delta$.

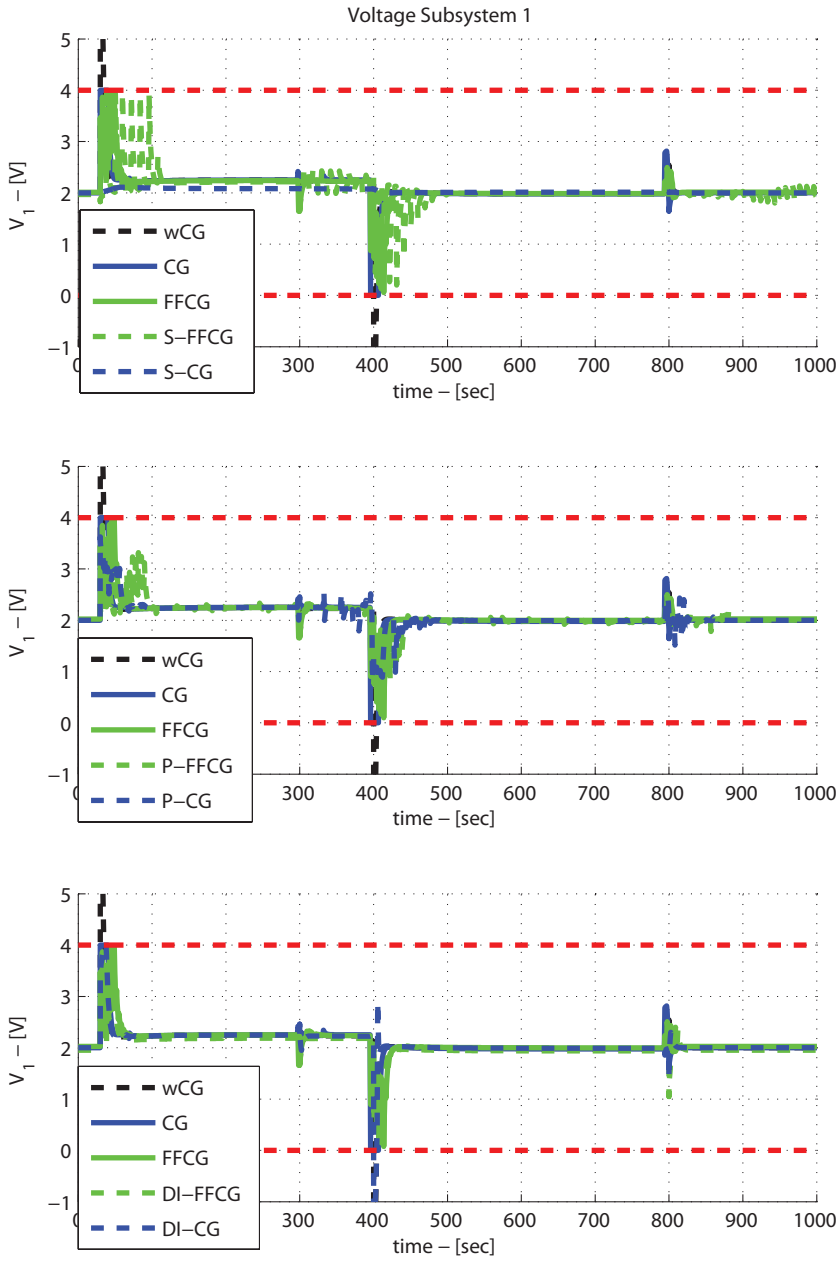

Fig. 8.6. Voltage - Input Subsystem 1

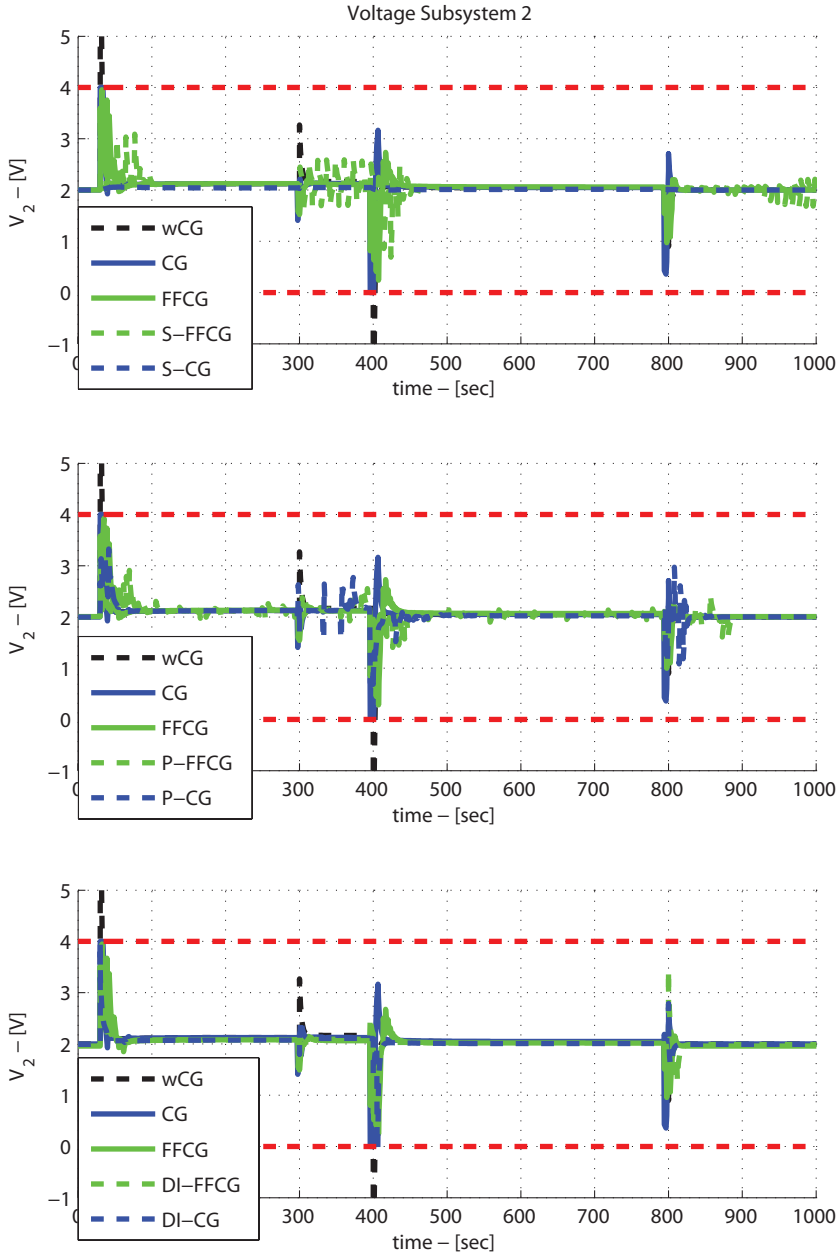

Fig. 8.7. Voltage - Input Subsystem 2

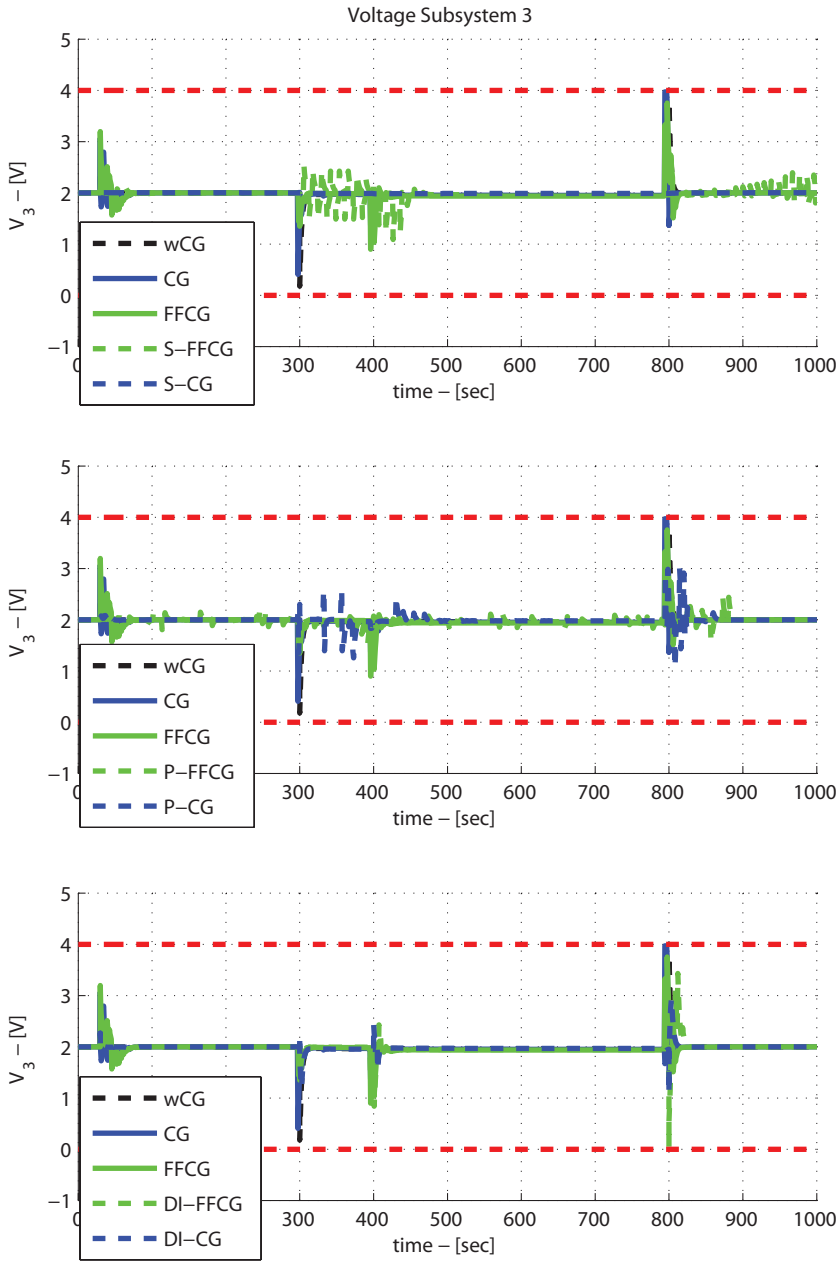

Fig. 8.8. Voltage - Input Subsystem 3

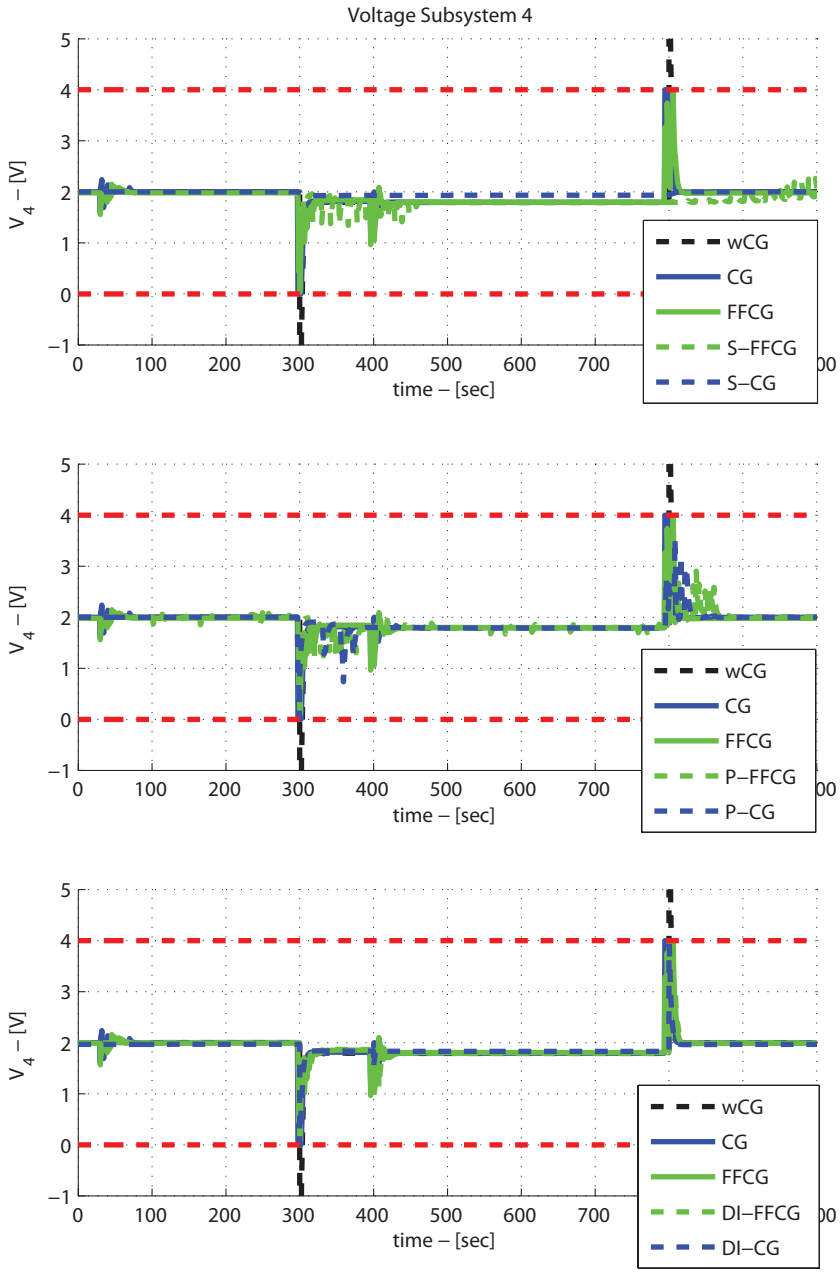

Fig. 8.9. Voltage - Input Subsystem 4

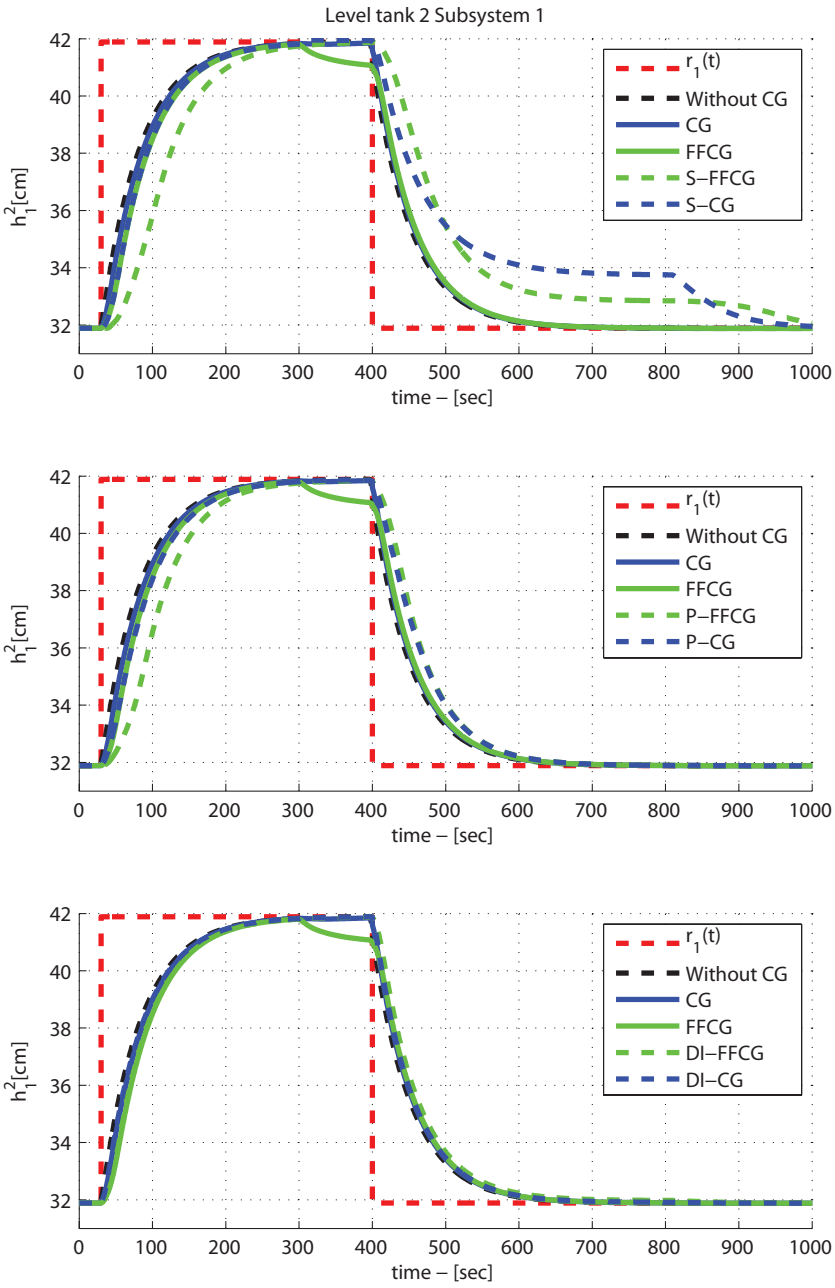

Fig. 8.10. Water level in the downstream tank of Subsystem 1

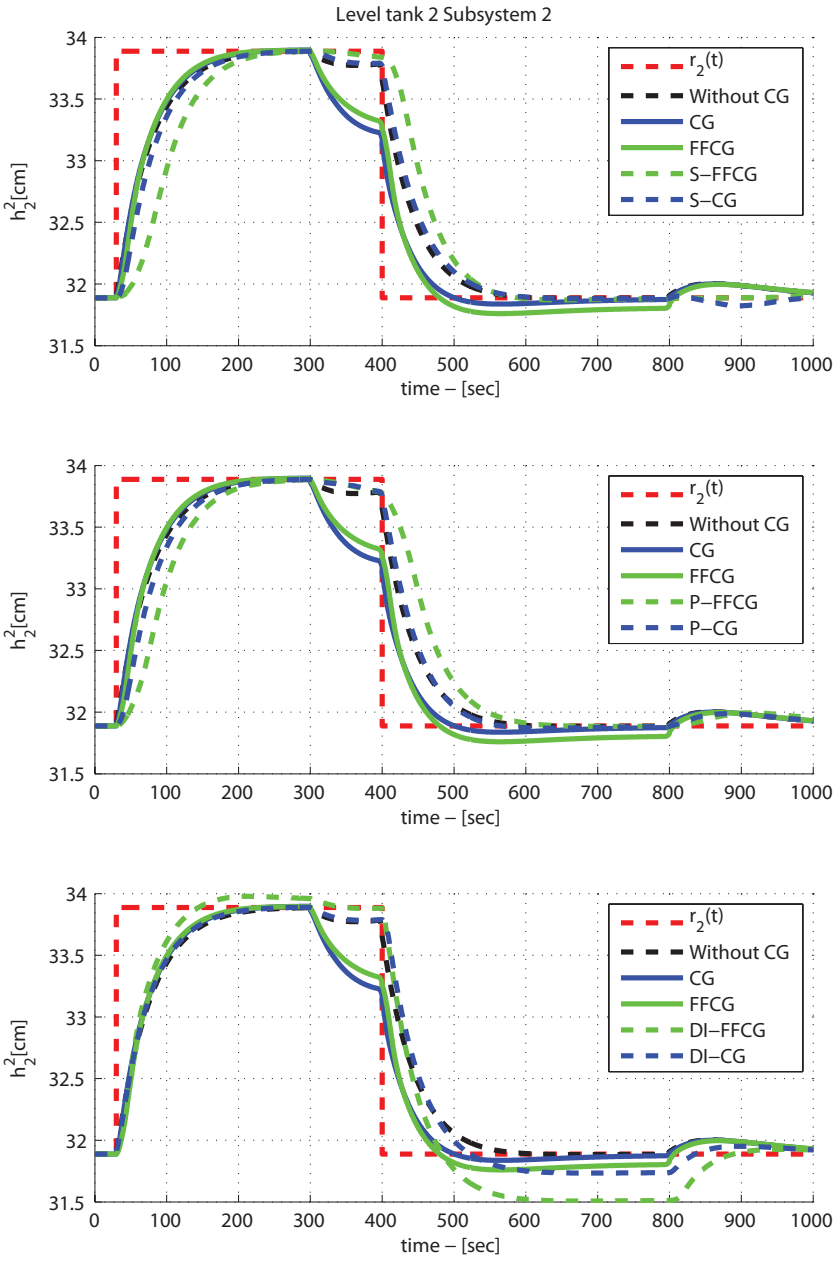

Fig. 8.11. Water level in the downstream tank of Subsystem 2

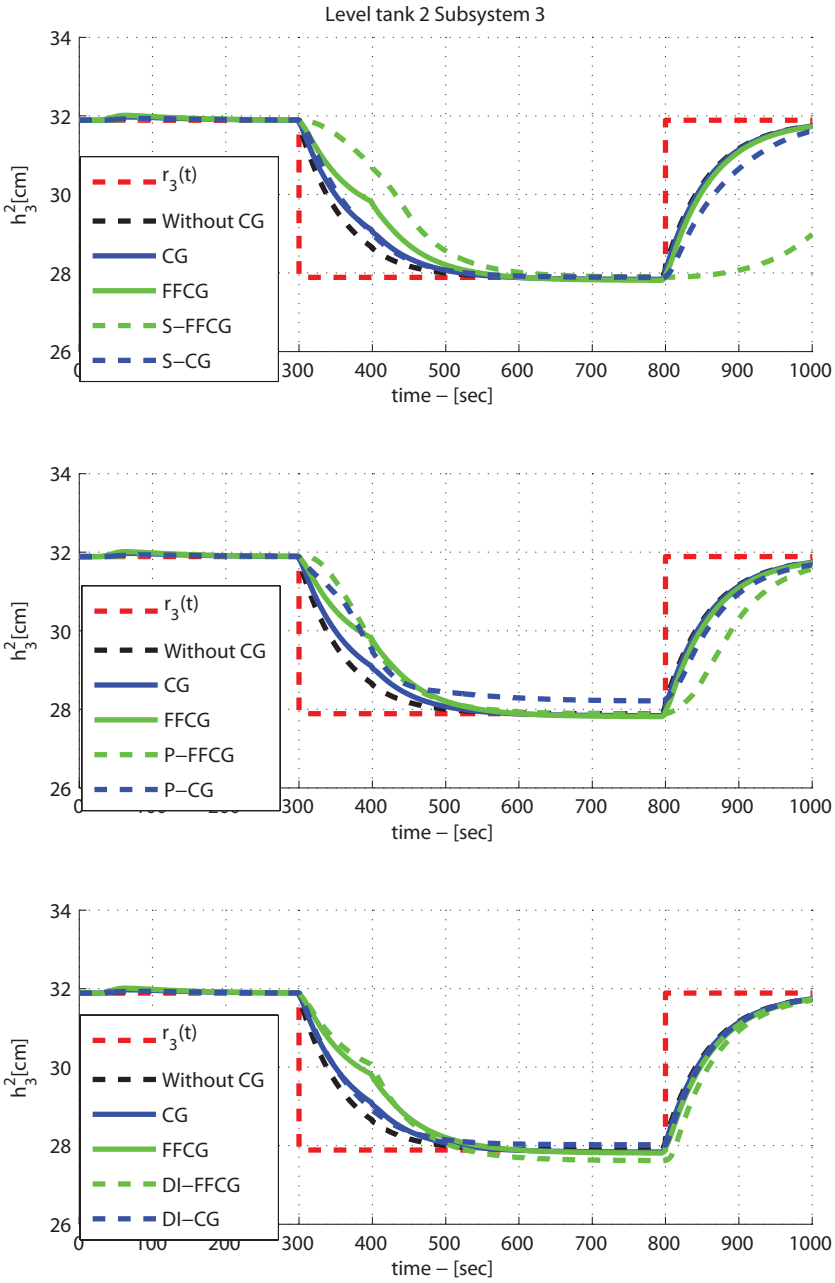

Fig. 8.12. Water level in the downstream tank of Subsystem 3

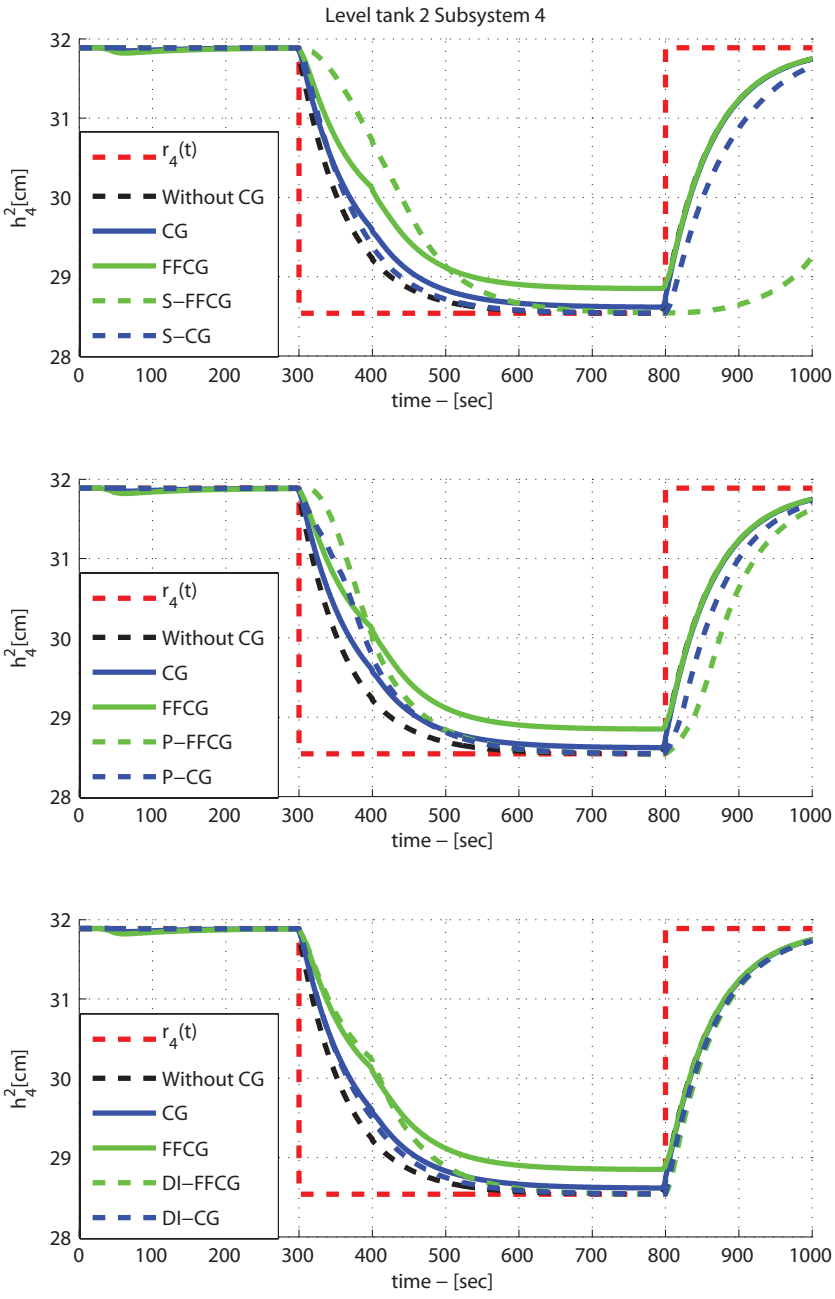

Fig. 8.13. Water level in the downstream tank of Subsystem 4

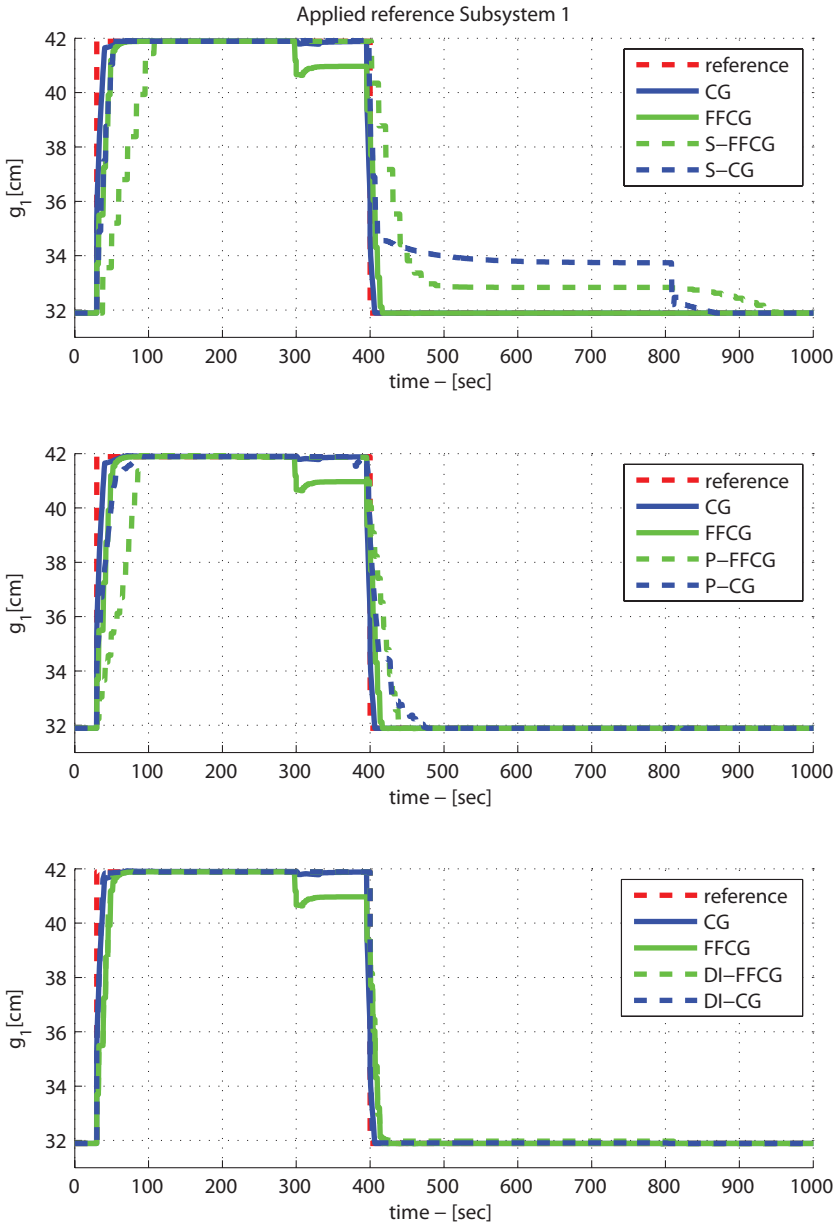

Fig. 8.14. Applied $g(t)$ Subsystem 1

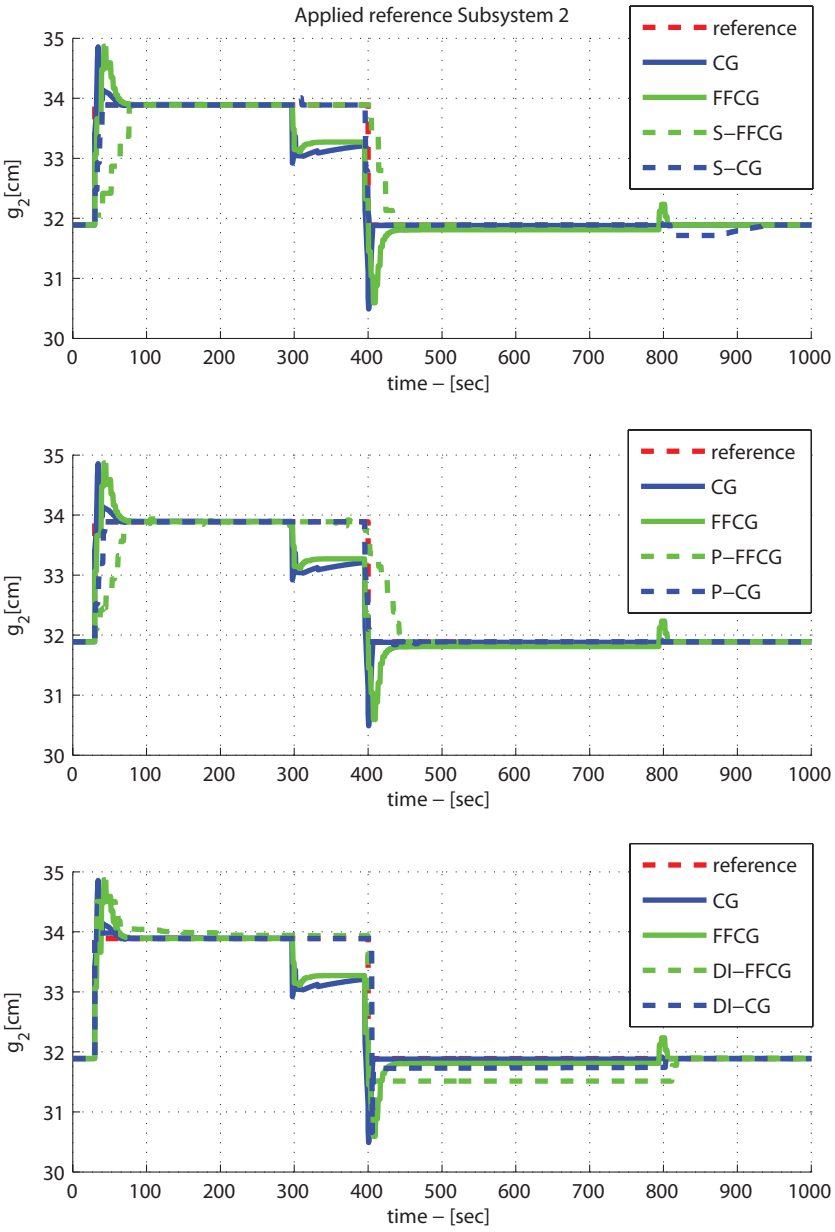

Fig. 8.15. Applied $g(t)$ Subsystem 2

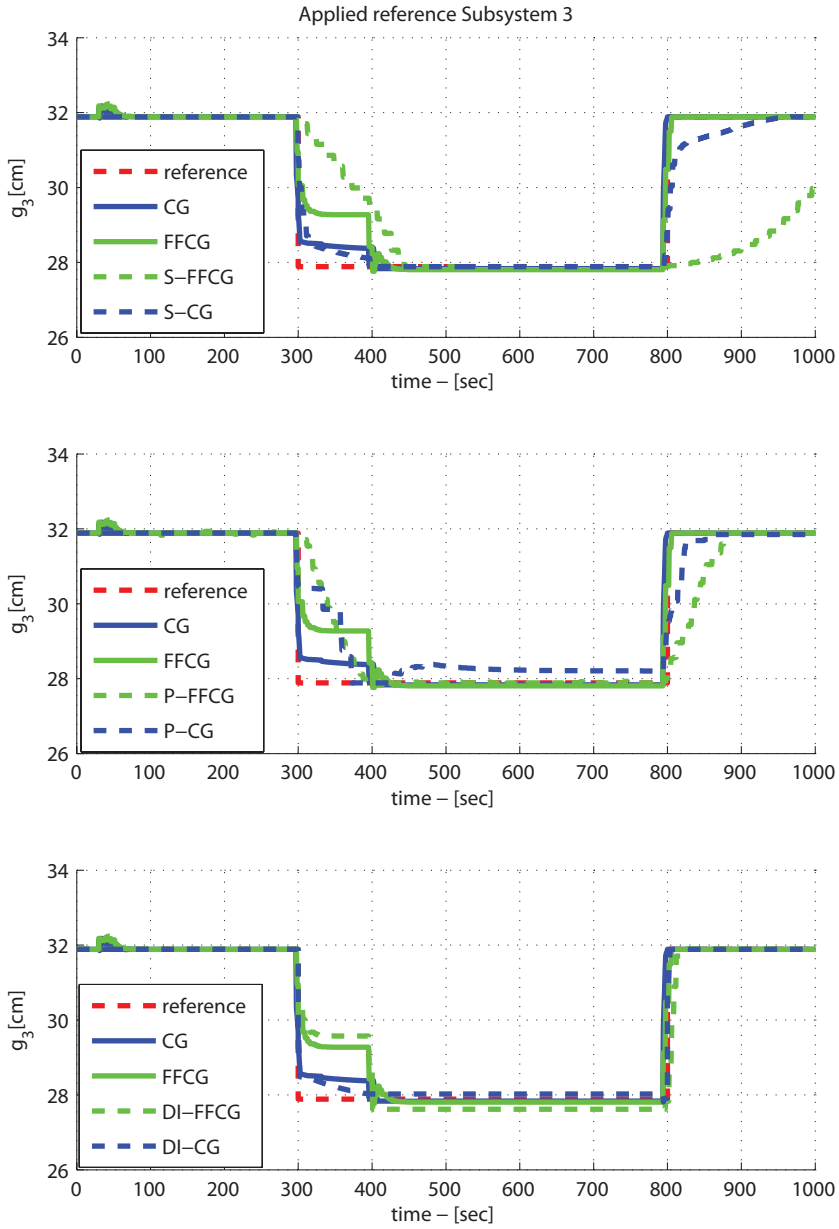

Fig. 8.16. Applied $g(t)$ Subsystem 3

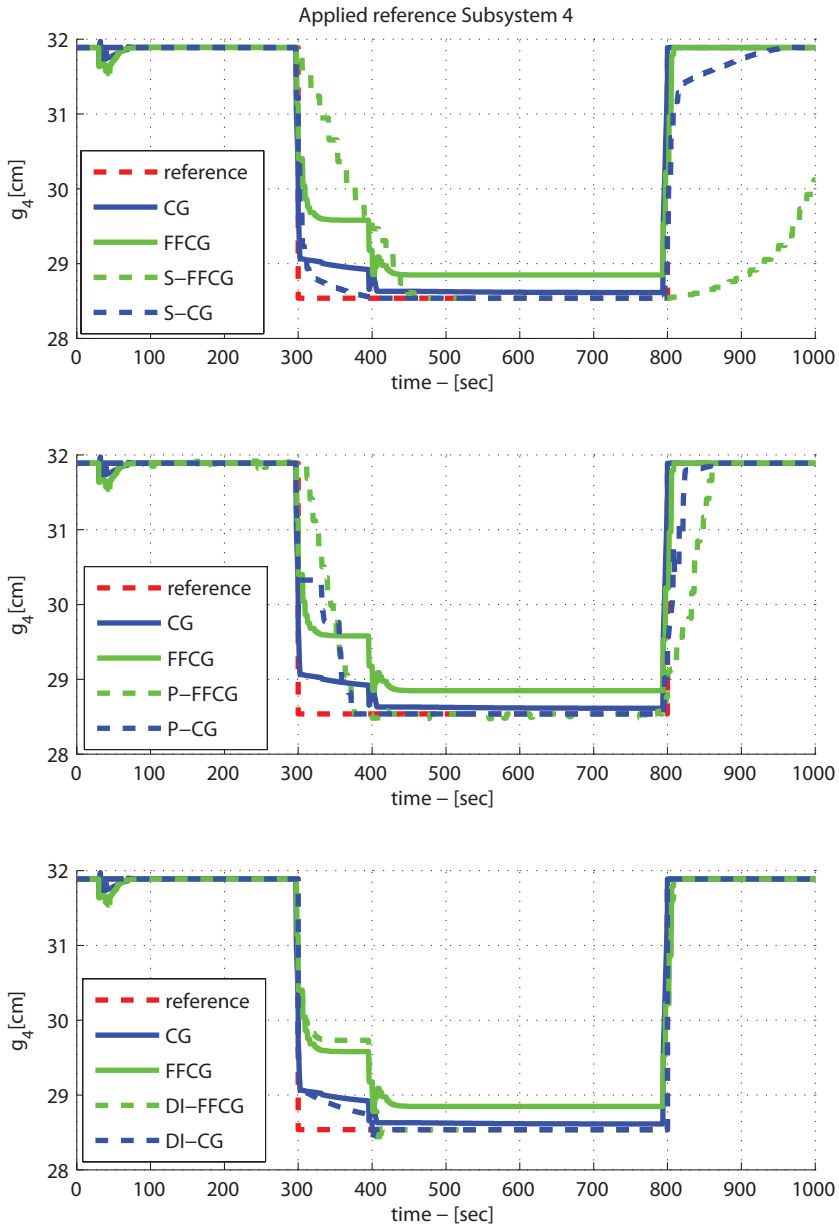

Fig. 8.17. Applied $g(t)$ Subsystem 4

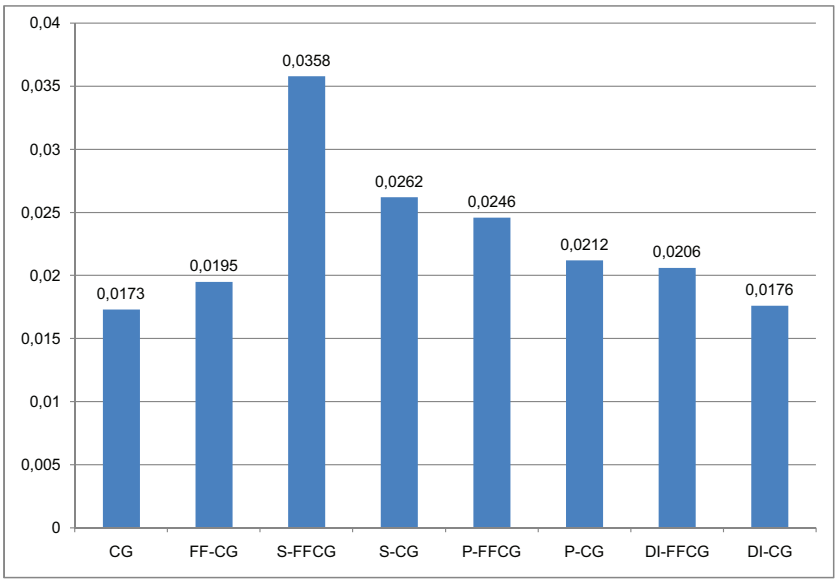

Fig. 8.18. Performance computation J

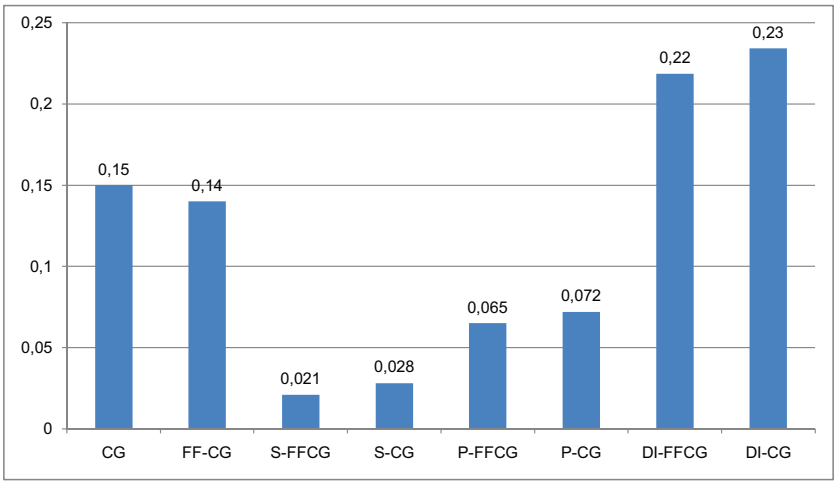

Fig. 8.19. CPU usage during simulations ([ms])

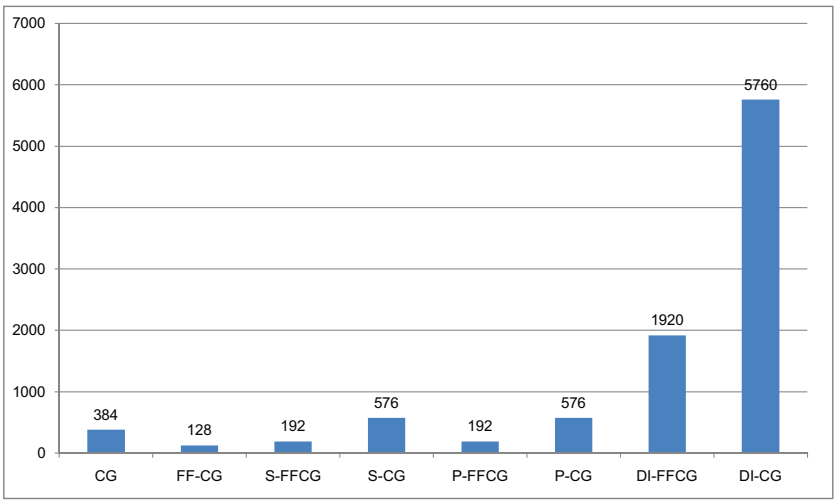

Fig. 8.20. Tx/Rx Data [bits/agent]

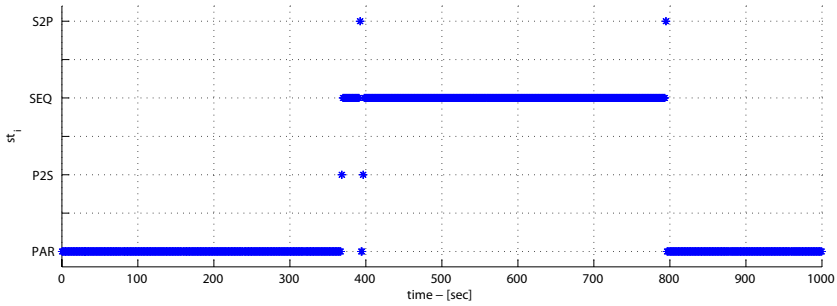

Fig. 8.21. Operative scenarios during P-FFCG action

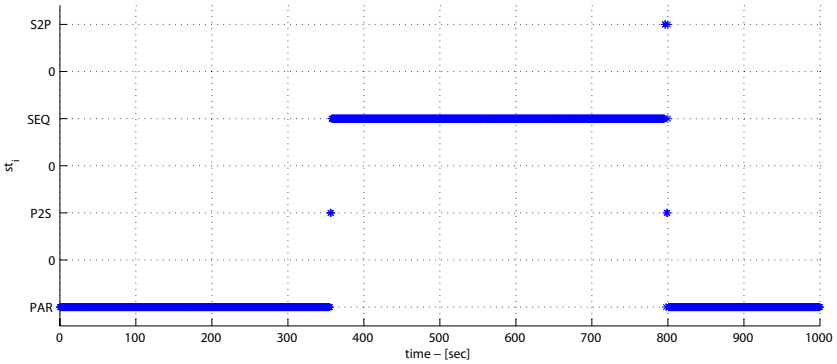

Fig. 8.22. Operative scenarios during P-CG action

Conclusions and Future works

The purpose of this dissertation was to discuss multi-agent Command Governor supervision schemes for distributed interconnected dynamical systems. The main contributions are here summarized and future research direction established.

A. Conclusions

The main contributions of this thesis with respect to the existing literature on supervision approaches are as follows:

- **Sensorless Command Governor schemes**. In Chapters 3 and 4 novel FeedForward CG schemes have been proposed which, thanks to the asymptotic stability of the precompensated interconnected systems, don't require for their action computation an explicit measure or estimate of the actual state. Their actions are derived under a constraint on the maximal allowable reference incremental variations which is instrumental to maintain the state trajectory "not too far" from the space of the admissible steady-state equilibria. In Chapter 3, such a behavior was shown to be achievable by forcing the FFCG action to stay constant for a precomputed number of sampling steps.
  In Chapter 4, an enhanced FFCG method is presented where the supervising action is updated at each sampling step. All the properties (stability, feasibility and computability) of the proposed algorithms have been carefully analyzed and the differences and similarities with standard CG approaches pointed out. Comparisons with the classical CG solutions have been presented and commented by means of several examples.
- **Distributed Command Governor schemes**. The second main contribution of this thesis relies in the presentation of a class of distributed supervision schemes based on centralized FFCG and CG ideas for a network of dynamically interconnected subsystems. It has been assumed that

such systems are connected with a communication data network and are supervised according to the *coordination-by-constraint* paradigm by a set of coordination and supervising agents. In Chapters 5 and 6 we have discussed non-iterative schemes and have proposed a sequential approach which, besides its own relevance, was shown to be useful also for the development of a more efficient parallel approach. Furthermore, we provided some conditions to guarantee feasibility of the schemes and the Pareto optimality of the solutions. In particular, distributed FFCG and CG schemes have been fully analyzed in Chapter 5 and, respectively, in Chapter 6. Finally, distributed FFCG and CG iterative schemes based on a distributed optimization procedure borrowed from the literature have been proposed in Chapter 7. All technical details required for the implementation of the proposed distributed schemes have also been provided and used in the final example of Chapter 8 where the supervision of a water network is considered. Comparisons amongst the methods have also been reported.

B. Future research

Although the work underlying this thesis has achieved the main goal of proposing and investigating distributed algorithms based on Command Governor strategies, several issues remain unsolved and need further investigation.

With respect to the FFCG approach addressed in this thesis, some challenging issues that require further investigations concerns

- the consideration of **uncertain systems**.
- **time-delay case: teleoperation**. In this case, there exists a non negligible physical distance from the master side, where the CG action is computed, and the slaves' sides where it is applied with a certain delay.
- **faulty situations** set-point reconfiguration strategies when a subset of actuators or sensors could fail.

Furthermore, concerning the distributed schemes, more general fundamental directions would include:

- **Asynchronous scenarios**. In this case agents have not to wait for other agents to solve their problems and decide on which action to take. However, agents have to include newly received information from neighborhood at any time while solving their own optimization problems.
- **Enhanced FFCG-based schemes**. Future work will involve the achievement of distributed iterative and not-iterative strategies based on the more effective FFCG method discussed in Chapter 4.
- **Scalability**. It remains to be addressed how the required CPU time and the amount of data exchanged change when applied to supervision structures with a large number of agents. If the approaches do not scale well, then ways to make them scalable should be investigated.

- **Robustness**. Future research should address how the schemes could be made robust to time-delays and packet loss. In addition, fault-tolerance against failing control agents is still an unsolved issue.

More in detail, with respect to non iterative techniques discussed in Chapters 5 and 6, several issues need to be further investigated:

- Although the asymptotically convergence to a Pareto Optimal solution has been proved only when the desired reference is constant, all performed numerical experiments show that the agents **converge to a Pareto Optimum in finite time**. Actually, no formal proofs are available and the investigation is ongoing.
- The Constraints Qualification property of the multi-box approximations discussed in Section 5.3.1 has been proved only for mono-dimensional decision set cases. Then, it would be desirable to arrive to a more power inner approximation method capable of generating **Constrained Qualified** polyhedrons in the general multi-dimensional case.
- In order to compute its local action each agent has to be provided with the model of the overall system. In general, for very large systems, a complete model could be not available or storable due to the inherent complexity. Then, an improvement could be the achievement of similar techniques where agents make use of a partial local simplified model.

In addition, future general application-oriented researches should investigate the use of the discussed approaches in other fields besides water networks. In this respect, several studies are ongoing concerning the set-point reconfiguration strategies for the supervision and coordination of smart-grid subsystems in addressing voltage and power generation control problems in the presence of faults or uncertainties due to distributed generation.

Further domains in which the supervision approaches here presented could be applied include road traffic networks, railway networks, gas distribution networks, but also the pro- cess industry, supply chains and autonomous guided or flying vehicles.

References

1. E.G. Gilbert and K. Tin-Tan (1991), Linear system with state and control constraints: The theory and application of maximal output admissible sets, IEEE Transaction on Automatic Control, Vol. 36: 1008-1020.
2. E.G. Gilbert, I. Kolmanovsky and K.T. Tan (1995), Discrete-time Reference Governors and the Nonlinear Control of Systems with State and Control Constraints, International Journal on Robust and Nonlinear Control, Vol. 5:487-504.
3. E.G. Gilbert and I. Kolmanovsky (1999), Fast reference governors for systems with state and control constraints and disturbance inputs, International Journal on Robust and Nonlinear Control, Vol. 9:1117-1141.
4. E.G. Gilbert and I. Kolmanovsky (2001), A generalized reference governor for nonlinear systems, Proc. IEEE CDC01 (2001), pp. 4222-4227.
5. A. Casavola, E. Mosca and D.Angeli (2008), Robust command governors for constrained linear systems, IEEE Transaction on Automatic Control, Vol. 45: 2071-2077.
6. A. Bemporad and E. Mosca (1998), Fulfilling Hard Constraints in Uncertain Linear Systems by Reference Managing, Automatica, Vol. 34: 451-461.
7. A. Angeli, A. Casavola and E. Mosca (2001), On feasible set-membership state estimators in constrained command governor control, Automatica, Vol. 37: 151-156.
8. A. Bemporad (1998), Reference Governor for Constrained Nonlinear Systems, IEEE Transactio on Automatic Control, Vol. 43: 415-419.
9. D. Angeli, A.Casavola and E. Mosca (1999), Command Governors for Constrained Nonlinear Systems: Direct Nonlinear vs. Linearization-based Strategies, International Journal of Robust and Nonlinear Control, Vol. 9: 677-699.
10. A. Angeli and E. Mosca (1999), Command governor for nonlinear systems under constraints, IEEE Transaction on Automatic Control, Vol. 44: 816-820.
11. A. Casavola, M. Papini and G. Franzè (2006), Supervision of networked dynamical systems under coordination constraints, IEEE Transaction on Automatic Control, Vol. 51(3): 421-437.
12. P. Falcone, F. Borrelli, J. Peckar and G. Steward (2009), A Reference Governor Approach for Constrained Piecewise Affine Systems, Proceedings of European Control Conference 2009, Budapest, Hungary.

13. T. Sugie and H. Yamamoto (2001), Reference Management for Closed Loop Systems with State and Control Constraints, Proceedings of IEEE American Control Conference 2001, Arlington, Virginia, U.S..
14. T. Hatanaka and K. Takaba (2005), Output Feedback Reference Governor for Nonlinear Systems, Proceedings of IEEE Conference on Decision and Control 2005, San Diego, CA, U.S..
15. C. M. Cirre, M. Berenguel (2009), L. Valenzuela and R. Klempous, Reference Governor Optimization and Control of a Distributed Solar Collector Field, European Journal of Operational Research, Vol. 193: 709-717.
16. M. Mattei, D. Famularo and C.V. Labate (2009), Constrained Plasma Shape Control in Iter, Proceedings of European Control Conference 2009, Budapest, Hungary.
17. A. Bemporad, (1997), Reference Governors: On-line Set-Point Optimization Techniques for Constraint Fulfillment, PhD thesis, Universit di Firenze. Firenze.
18. Mayne D. Q., Rawlings J. B.,Rao C. V., and Scokaert P. O. M. (2000), Constrained model predictive control: Stability and optimality, Automatica, vol. 36(6): 789-814.
19. C.E. Garcia, D.M. Prett, M. Morari (1989), Model predictive control: theory and practice: a survey, Automatica, vol. 25: 335-348.
20. D.Q. Mayne, M.M. Seron, V. Rakovic (2005), Robust model predictive control of constrained linear systems with bounded disturbances, Automatica 41: 219224.
21. R. Scattolini (2009), Architectures for distributed and hierarchical model predictive control - a review, Journal of Process Control, vol. 19(5): 723731.
22. T.S. Basar, J.G. Olsder (1999), Dynamic Noncooperative Game Theory, SIAM, Philadelphia.
23. Garone E., Tedesco F., Casavola A. (2010), A FeedForward Command Governor Strategy for Constrained Linear Systems, 8th IFAC Symposium on Nonlinear Control Systems (2010).
24. Garone E., Tedesco F., Casavola A. (2011), Sensorless Supervision of Linear Dynamical Systems: The Feed-Forward Command Governor Approach, Automatica, in press.
25. N.R. Sandell, P. Varaiya, M. Athans, M.G. Safonov (1978), Survey of decentralized control methods for large scale systems, IEEE Transactions on Automatic Control, Vol. 23: 108-128.
26. A. Iftar (1993), Decentralized estimation and control with overlapping input state and output decomposition, Automatica, vol. 29: 511-516.
27. M. Ikeda, D.D. Siljak, D.E. White (1981), Decentralized control with overlapping information sets, Journal of Optimization Theory and Application, Vol. 34: 279-310.
28. L. Magni, R. Scattolini (2006), Stabilizing decentralized model predictive control of nonlinear systems, Automatica, Vol. 42: 1231-1236.
29. D.M. Raimondo, L. Magni, R. Scattolini (2007), Decentralized MPC of nonlinear systems: an input-to-state stability approach, International Journal of Robust and Nonlinear Control, Vol. 17: 1651-1667.
30. L. Acar (1995), Boundaries of receding horizon control for interconnected systems, Journal of Optimization Theory and Applications, Vol. 84(2): 251-271.
31. D. Jia, B. Krogh (2001), Distributed model predictive control, in: Proceedings of the IEEE American Control Conference, Arlington, VA, USA.

32. E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar (2002), Distributed model predictive control, IEEE Control Systems Magazine, vol. 22(1): 44-52.
33. A.N. Venkat, J.B. Rawlings, S.J. Wright (2005), Stability and optimality of distributed model predictive control, in Proceedings of the 46th IEEE Conference on Decision and Control, Seville, Spain.
34. W. B. Dunbar (2007), Distributed receding horizon control of dynamically coupled nolinear systems, TAC, vol. 52(7): 1249-1263.
35. W. B. Dunbar and R. M. Murray (2006), Distributed receding horizon control for multi-vehicle formation stabilization, Automatica, Vol. 42: 549-558.
36. T. Keviczky, F. Borrelli, and G. J. Balas (2006), Decentralized receding horizon control for large scale dynamically decoupled systems. Automatica, Vol. 42(12): 2105-2115.
37. R. R. Negenborn, B. De Schutter, and J. Hellendoorn (2008), Multi-agent model predictive control for trans- portation networks: Serial versus parallel schemes, Engineering Applications of Artificial Intelligence, Vol. 21(3): 353-366.
38. A. Richards and J. P. How (2007), Robust distributed model predictive control, International Journal of Control, vol. 80(9): 1517-1531.
39. A. Alessio, A. Bemporad (2007), Decentralized model predictive control of constrained linear systems, in: Proceedings of the European Control Conference, Kos, Greece.
40. A. Alessio, A. Bemporad (2008), Stability conditions for decentralized model predictive control under packet drop communication, in Proceedings of the American Control Conference, Seattle, Washington, U.S..
41. A.N. Venkat (2006), Distributed Model Predictive Control: Theory and Applications, PhD Thesis, University of Wisconsin, USA.
42. Casavola A., Garone E., Tedesco F. (2011), Improved Feed-Forward Command Governor Strategies for Constrained Discrete-time Linear Systems, IEEE Transactions on Automatic Control, (Submitted).
43. Garone E., Tedesco F., Casavola A. (2010), Distributed Coordination Strategies for Interconnected Multi-Agent Systems, 8th IFAC Symposium on Nonlinear Control Systems.
44. Garone, E., Tedesco, F., Casavola, A. (2009), Distributed Coordination-by-Constraint Strategies for Networked Control Systems, 1st IFAC Workshop on Estimation and Control of Networked Systems, Venice, Italy.
45. Tedesco F., Raimondo, D. M., Casavola A., Lygeros J. (2010), Distributed collision avoidance for interacting vehicles: a command governor approach, 2nd IFAC Workshop on Estimation and Control of Networked Systems, Annecy, France.
46. Casavola A. , Franzè G. , Garone E. , Tedesco F. (2011) , Distributed Coordination-by-Constraint Strategies in Networked Multi-Area Power Systems, IEEE ISIE 2011, Gdansk, Poland.
47. Casavola A. , Garone E. , Tedesco F. (2011), Distributed Reference Management Strategies for a Networked Water Distribution System, 18th IFAC World Congress, Milan (Italy).
48. Casavola A. , Garone E. , Tedesco F. (2011) , Distributed Coordination-by-Constraint Strategies for Multi-agent Networked Systems, CDC 2011, Orlando, USA.
49. Casavola A. , Garone E. , Tedesco F. (2011), A Liveliness Analysis of a Distributed Constrained Coordination Strategy for Multi-Agent Linear Systems, CDC 2011, Orlando, USA.

50. A. Kurzhanski, and I. Valyi (1997), Ellipsoidal calculus for estimation and control, Birkhauser, Berlin.
51. A. Casavola, (2001), The Command Governor Approach for Dummies, DEIS.
52. Horst, R., and N.V. Thoai (1999), DC programming: overview. Journal of Optimization Theory and Applications 103, no. 1.
53. R. Horst, P. M., Pardalos, and N.V. Thoai (1995), Introduction to Global Optimization, Kluwer Academic Publishers, Dordrecht, Holland.
54. A. Bemporad, C. Filippi, and F. D. Torrisi (2004), Inner and outer approximations of polytopes using boxes, Computational Geometry, Vol. 27: 151-178.
55. E. Garone, F. Tedesco, and A. Casavola (2009), Distributed Coordination-by-Constraint Supervision Strategies for Networked Control Systems, 1st IFAC Necsys, Venezia, Italy.
56. M. Kvasnica, P. Grieder, M. Baotic, M. Morari (2004), Multi-Parametric Toolbox (MPT) Hybrid Systems: Computation and Control, Lecture Notes in Computer Science Vol. 2993: 121-124.
57. D. P. Bertsekas (2007), Separable Dynamic Programming and Approximate Decomposition Methods, IEEE Trans. on Aut. Control, Vol. 52: 911-916.
58. A. Casavola and Franzè (2007), Dealing with Time-Delay in Coordination by Constraints Methods, First International Workshop on Control over Communication Networks - CONCOM 2007, Ciprus.
59. K. M. Miettinen (1999), Nonlinear Multiobjective Optimization, Kluwer Academic.
60. G. Inalhan (2004), Decentralized optimization across independent decision makers with incomplete models, Ph.D. Thesis, Stanford University USA.
61. G. Inalhan, D.M. Stipanovic, C. J Tomlin, (2002) Decentralized optimization, with application to multiple aircraft coordination, Proceedings of the 41st IEEE Conference on Decision and Control.
62. D. G. Luenberger (1989), Linear and Nonlinear Programming, Addison Wesley, Reading, MA, second edition.
63. D. D. S̃iljak (1991), Decentralized Control of Complex Systems, Academic Press, New York, USA.
64. D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert (2000), Constrained model predictive control: Stability and optimality, Automatica, Vol. 36(6): 789-814.
65. B. T. Wahlin and A. J. Clemmens (2006), Automatic downstream water-level feedback control of branching canal networks: Simulation results, Journal of Irrigation and Drainage Engineering, Vol. 132(3): 208-219.
66. O. Begovich, V. M. Ruiz, G. Besancon, C. I. Aldana, and D. Georges (2007), Predictive control with constraints of a multi-pool irrigation canal prototype, Latin American Applied Research, Vol. 37(3): 177-185.
67. C. Ocampo-Martinez,V. Puig, G. Cembrano, R. Creus, M. Minoves (2009), Improving water management efficiency by using optimization-based control strategies: the Barcelona case study, Water science and technology: water supply,Vol. 9(5): 565-575.
68. G. Georges (1999), Decentralized adaptive control for a water distribution system, in Proceedings of the 3rd IEEE Conference on Control Applications, Glasgow, UK, pp. 1411-1416.
69. H. El Fawal, D. Georges, and G. Bornard (1998), Optimal control of complex irrigation systems via decomposition-coordination and the use of augmented

Lagrangian, in Proceedings of the 1998 International Conference on Systems, Man, and Cybernetics, San Diego, California, pp. 3874-3879.
70. R.R. Negenborn, A. Sahin, Z. Lukszo, B. De Schutter, and M. Morari (2009), A non-iterative cascaded predictive control approach for control of irrigation canals, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA.
71. Y. Zhu, P.R. Pagilla (2006), Decentralized output feedback control of a class of large-scale interconnected systems, IMA Journal of Mathematical Control and Information, 24(1):57-69.