Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A Cooperative Game Theoretical Approach to Distributed Iterative Command Governor Schemes

Francesco Tedesco and Alessandro Casavola

Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Università degli Studi della Calabria, Via Pietro Bucci, Cubo 42-c, Rende (CS), 87036, Italy,
{ftedesco, casavola}@dimes.unical.it

Abstract: A novel distributed Command Governor (CG) supervision strategy relying on cooperative game theoretical concepts is presented for multi-agent networked systems subject to pointwise-in-time coordination constraints. Unlike non-cooperative distributed CG schemes, here all agents contribute individually to the minimization of a global performance index. As a result, these methods are able to achieve Pareto-optimal solutions, not only in steady-state conditions as the non-cooperative ones, but also during transients and are not hampered by the presence of undesirable non-Pareto Nash-equilibria or deadlock situations. Other noticeable difference with respect to non-cooperative methods is that all agents need to exchange data among them several times within a decision step in order to coordinate their local optimization procedures and arrive to the optimal solution. The properties of the algorithm are fully investigated. A final example is presented where the proposed distributed solution is contrasted with both centralized CG solutions and distributed CG methods based on non-cooperative game theoretical concepts.

Keywords: Nonlinear Control, Distributed Predictive Control, Cooperative Game Theory, Command Governor.

1. INTRODUCTION

The problem of interest here is the design of distributed supervision strategies for large-scale multi-agent systems in situations where the use of centralized solutions might result impracticable. The distributed context under consideration is depicted in Figure 1, where the supervisory task is distributed amongst many agents which are assumed to be able to communicate each other through a communication network. There, each agent is in charge of supervising and coordinating one specific subsystem. In particular, r_i, g_i, x_i, y_i and c_i represent respectively: the nominal reference, the applied reference, the state, performance and the coordination related output vectors of the $i-$th subsystem.

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 subsystem and/or on the overall network evolutions. To this end, each $i-$th supervisor is in charge of modifying its nominal local reference r_i into the feasible one g_i, when the joint application of all nominal references would produce constraint violations and hence loss of coordination.

[1] This work has been partially supported by the European Commission, the European Social Fund and the Calabria Region. The authors are solely responsible for the content of this paper and the European Commission and Calabria Region disclaim any responsibility for the use that may be made of the information contained therein.

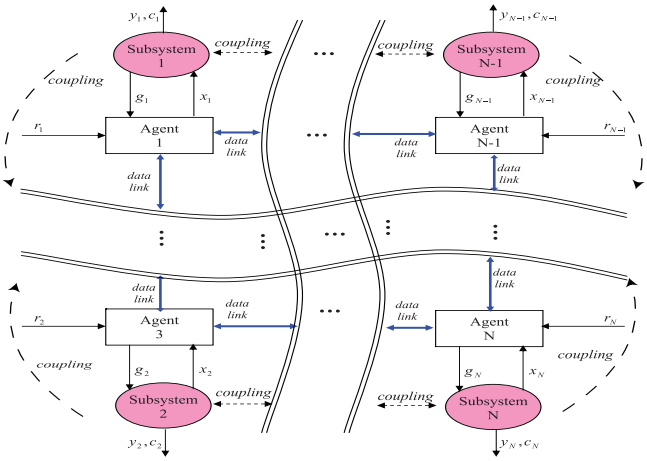

Fig. 1. Multi-agent architectures

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 in this distributed context.

Earlier works on distributed CG strategies relied on a non-cooperative game theoretical approach and several strategies have been proposed, both *sequential* (Casavola *et al.*. [2011-A]),(Tedesco *et al.*.A [2012]) where, according to a prefixed order, only one agent at each sampling time is allowed to update its control action while all others keep applying their previous applied commands, and *parallel* (Tedesco *et al.*.B [2012]), (Casavola *et al.* [2014]), (Tedesco *et al.*.C [2012]) where, on the contrary, all agents

update their control actions simultaneously at each sampling time.

One peculiarity of all above schemes is that each agent computes its control action by solving a local optimization problem and the needs of agents coordination is limited, accomplished by a few data exchange occurring only one time per decision step. For this reason their have been referred to as *non-Iterative* in (Scattolini [2009]). Also note that mixed sequential-parallel approaches are possible (Tedesco *et al.*.C [2012]).

In this paper we move towards the class of cooperative methods, which differ for the fact that all agents contribute individually to the minimization of a global performance index. In fact, the performance index that each agent locally minimizes, besides local self-fish objectives, contains also terms related to the collective goal to achieve and, unlike non-cooperative methods, lead to the achievement of Pareto-optimal coordination performance. Notice that, in order to achieve this global utility, all agents need to exchange data many times during solving their local optimization problems for coordinating the solutions. For this reason such schemes have been referred to as *Iterative* in (Scattolini [2009]). Cooperative games theory concepts have been also preliminary considered in (Chinchuluun et al. [2008]).

Thus, the goal here is to recast the CG supervision design problems already introduced in (Bemporad *et al.* [1997]) into an iterative multi-objective optimization scheme and solve them in a distributed way. Several methods for solving such a kind of problems have been reported in the literature (see Miettinen [1999]). Amongst many, we resort here to the iterative distributed optimization procedure presented in (Inalhan *et al.* [2002]).

The main benefit related to the proposed class of distributed cooperative iterative CG strategies is the ability to achieve Pareto optimal coordination performance not only in steady-state conditions but also during transients. Moreover, deadlock situations and/or undesirable Nash-equilibria are avoided from the outset without requiring that the constraints fulfill any particular geometrical conditions, as the liveness property required for the non-cooperative CG strategies of (Casavola *et al.*. [2011-B]). As a further advantage, agents don't need to know the entire model of the network but only the models of the "neighboring" subsystems, that is the agents with which they exchange data.

2. NOTATIONS AND DEFINITIONS

$\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{Z}_+$ denote respectively the real, non-negative real and non-negative integer numbers. The Euclidean norm of a vector $x \in \mathbb{R}^n$ is denoted by $\|x\| = \sqrt{x_1^2 + ... + x_n^2}$ whereas $\|x\|_\Psi^2, \Psi = \Psi^T > 0$, denotes the quadratic form $x^T \Psi x$. Let A be a matrix in $\mathbb{R}^{m\times n}$, then $a_i^T \in \mathbb{R}^n$ denotes the i-th line of A and $a_i^{(j)}$ its j-th element.

3. 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

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

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

$$c_i(t) \in \mathcal{C}^i, \ \forall t \in \mathbb{Z}_+, \quad (2)$$

$\mathcal{C}^i$ 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 (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

$$\begin{cases} x(t+1) = \Phi x(t) + Gg(t) \\ y(t) = H^y x(t) \\ c(t) = H^c x(t) + Lg(t) \end{cases} \quad (3)$$

where $\Phi = [\Phi_{ij}]_{i,j=1,...,N}$ $G = diag(G_1, ..., G_N)$, $H^y = diag(H_1^y, ..., H_N^y)$, $H^c = [(H_1^c)^T, ..., (H_N^c)^T]^T$ and $L = [(L_1)^T, ..., (L_N)^T]^T$.

Hereafter, it is assumed that: (**A1**) Φ is a Schur matrix.

The CG design problem consists of locally determining, 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. $c_i(t) \in \mathcal{C}^i, \forall t \in \mathbb{Z}_+, i \in \mathcal{A}$, under the application of a suitably computed sequence of feasible $\{g_i(t)\}_{t=0}^\infty$.

A centralized solution of the above stated CG design problem have been proposed in (Bemporad *et al.* [1997]), whose architecture is depicted in Figure 2. There, at each time instant t, the CG action $g(t)$ is determined as a function of the current reference $r(t)$ and measured state $x(t)$ so that $g(t)$ results the best feasible approximation of $r(t)$ under

$$c(t) \in \mathcal{C} \quad (4)$$

where $\mathcal{C}$ is the constrained region.

In particular, the centralized CG action is chosen as the solution of the following constrained optimization problem

$$\hat{g}(t) = \arg \min_{g\in\mathcal{V}(x(t))} \|g - r(t)\|_\Psi^2 \quad (5)$$

where $\Psi = \Psi^T > 0_m$ and $\mathcal{V}(x)$ is defined as

$$\mathcal{V}(x) = \mathcal{W}_\delta \cap \mathcal{G}(x) \quad (6)$$

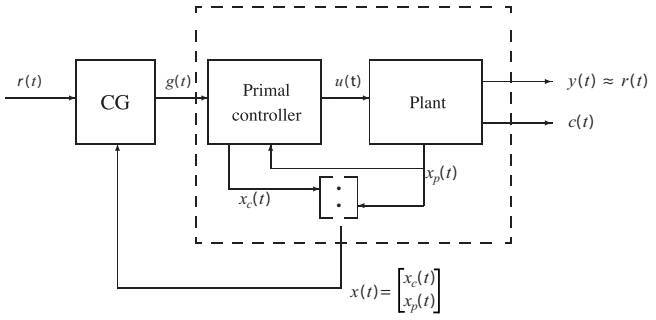

Fig. 2. Command Governor Control Scheme

with $\mathcal{W}_\delta$ representing the set of all constant commands whose corresponding steady-state c-vectors

$$c_g := H_c(I_n - \Phi)^{-1}Gg + Lg \tag{7}$$

satisfy the constraints with margin $\delta > 0$, i.e.

$$\mathcal{W}_\delta := \{g \in \mathbb{R}^m : \ c_g \in \mathcal{C} \sim \mathcal{B}_\delta\} \tag{8}$$

with $\mathcal{B}_\delta^q$ denoting hereafter the ball of radius δ centered at q (in this case $\mathcal{B}_\delta$ is centered at the origin). Moreover, $\mathcal{G}(x)$ describes the set of all constant commands $g \in \mathbb{R}^m$ whose corresponding virtual c-evolutions

$$c(k, x, g) := H_c\Phi^k x(t) + \sum_{i=0}^{k-1} \Phi^{k-i-1}Gg + Lg \tag{9}$$

starting at time $k = 0$ from the current state x satisfy the constraints for all $k \in \mathbb{Z}_+$, i.e.

$$\mathcal{G}(x) = \{g \in \mathbb{R}^m : c(k, x, g) \in \mathcal{C}_k, \ \forall k \in \mathbb{Z}_+\}. \tag{10}$$

In particular, because $\mathcal{V}(x)$ is finitely determined and assuming $\mathcal{C}_i$ being polytopes, $\mathcal{V}(x)$ can be characterized by a finite number z of inequalities in $\mathbb{R}^m$ (Kolmanovsky and Gilbert [1995])

$$\mathcal{V}(x) := \{g \in \mathbb{R}^m : Ag + \tilde{A}x - b \leq 0\} \tag{11}$$

By denoting $h(g, x) := Ag + \tilde{A}x - b$, h being an affine function $h : \mathbb{R}^m \to \mathbb{R}^z$, an equivalent form for (5) is

$$\begin{array}{ll} \min\limits_{g \in \mathbb{R}^m} & \|g - r(t)\|_\Psi^2 \\ s.t. & h(g, x(t)) \leq 0 \end{array} \tag{12}$$

The solution for the above problem (12) represents the *best* feasible approximation of $r(t)$ which, if constantly applied from t onwards, never produces constraints violation. It is worth noticing that, if $r(t) \notin \mathcal{V}(x)$, the solution of (12) strongly depends on the shape of weighting matrix Ψ that, according to the designer preferences, *selects* a suitable solution from the set of all Pareto-optimal solutions related to the more general multi-objective optimization program

$$\begin{array}{ll} \min\limits_{g \in \mathbb{R}^m} & [f_1(g_1, r_1), ..., f_i(g_i, r_i), ..., f_N(g_N, r_N)] \\ s.t. & h(g, x) \leq 0 \end{array} \tag{13}$$

with $f_i(g_i, r_i) = \| \ g_i - r_i \ \|_{\Psi_i}^2, \forall i \in \mathcal{A}$. In this respect Problem (12) represents a scalarized version of (13). A particular scalarization case, of interest here, envisages the use of a block-diagonal matrix Ψ in (12), that is

$$\Psi := diag(\alpha_1\Psi_1, ..., \alpha_N\Psi_N), \ \ \Psi_i = \Psi_i^T > 0_{m_i}, \alpha_i > 0, \ \forall i \in \mathcal{A} \tag{14}$$

Then a PO solution g^* for (13) is the optimal solution for (12) iff it satisfies the following Karush-Khun-Tucker (KKT) conditions (Miettinen [1999])

$$\begin{array}{l} \sum\limits_{i=1}^{N} \alpha_i \ \dfrac{\partial}{\partial g} f_i(g_i, r_i)\Big|_{g_i = g_i^*} + \sum\limits_{j=1}^{z} \mu_j \ \dfrac{\partial}{\partial g} h^j(g, x)\Big|_{g = g^*} \\ \mu^T h(g^*, x) = 0 \\ \mu_j \geq 0, j = 1, ..., z \end{array} \tag{15}$$

with $\mu = [\mu_1, ..., \mu_z] \in \mathbb{R}^z$ and h^j representing the j-th rows of h.

4. DISTRIBUTED ITERATIVE CG (DI-CG)

As already seen in the previous section, a centralized implementation of the multi-objective optimization problem (13) requires a central computational facility with access to all system information. On the contrary, here we are interested in the implementation of N computational nodes, each one with restricted information about the whole system. Their main goal is the determination of a PO solution for the problem (13) at each time instant. Furthermore, it can be shown that such a solution, under certain conditions, is also the optimal solution for the scalarized centralized problem (12). For such comparisons, it is hereafter assumed that matrices Ψ of the form (14) are used in both problems (12) and (13), with each α_i being known to the related agent $i, \forall i \in \mathcal{A}$.

In order to consider a limited information scope for all agents, the notion of neighborhood of a given agent i is required:

Definition 4.1. (**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 involved with g_i in some constraints and have a direct communication link with node i, that is $\mathcal{N}_i = \{j \in \mathcal{A} :$ in (6) $\exists$ at least a line a_p in A or a line a_x in $\tilde{A}$ such that ($a_p^{(i)} \neq 0$ and $a_p^{(j)} \neq 0$), ($a_x^{(i)} \neq 0$ and $a_x^{(j)} \neq 0$) respectively$\}$

As an immediate consequence of Definition 4.1, the sets of all commands g_j and state x_j respectively associated to the i-th agent can be characterized in the following way

$$[g]_i = \{\text{All subvectors } g_j \text{ of } g \text{ such that } j \in \mathcal{N}_i\} \tag{16}$$

$$[x]_i = \{\text{All subvectors } x_j \text{ of } x \text{ such that } j \in \mathcal{N}_i\} \tag{17}$$

Then, the problem of interest can be defined as a distributed optimization problem for each agent i having the following form

$$\begin{array}{ll} \min\limits_{g_i \in \mathbb{R}^{n_i}} & f_i(g_i) \\ s.t. & h_i(g_i, x_i | [g]_i, [x]_i) \leq 0 \end{array} \tag{18}$$

where the notation $h_i(\cdot|[g]_i, [x]_j)$ is used to represent sets of rows of $h(g, x)$ in (12) as functions of g_i, given that the local state x_i, the neighborhood decision variables $[g]_i$ and the state $[x]_i$ are fixed.

4.1 Distributed Optimization Based on Penalty Methods

In this section the distributed optimization method presented in (Inalhan *et al.* [2002]) is recalled and it will be used to solve in a distributed way problem (18) and, in turn, problem (13). To this end, consider for each i-th agent the following augmented cost function, related to problem (18), reinforced by the penalty function P_i

$$\begin{aligned} F_i(g_i, \beta_i | [g]_i) &:= \beta_i f_i(g_i) + P_i(g_i | [g]_i, [x]_j) \\ &= \beta_i [f_i(g_i) + \frac{1}{\beta_i} P_i(g_i | [g]_i, [x]_j]), \beta_i \neq 0 \end{aligned} \tag{19}$$

where $\beta_i \geq 0$ is a local penalty parameter. It is also assumed that:

(**A2**) $F_i(g_i^{l+1}, \beta_i^{l+1}|[g]_i)$ embeds all constraints $h_i(g_i, [g]_i) \leq 0$ associated to g_i. □

The local penalty functions, P_i, are chosen as

$$P_i(g_i|[g]_i) := \sum_{j=1}^{z_i} \max(0, h_i^j(g_i, x_i|[g]_i, [x]_i))^\gamma, \gamma \geq 2 \quad (20)$$

With this choice, the local penalty function penalizes violations of the constraints given in (18) (i.e. $P_i(g_i|[g]_i) = 0 \Longleftrightarrow h_i(g_i, x_i|[g]_i, [x]_i) \leq 0$). It is finally assumed that :

(**A3**) Common global constraints (i.e. interconnection of agent constraints) and their penalty functions enter each associated agent optimization problem identically: $h_i^j(g_i, x_i|g_k, ...) = h_k^l(g_k|g_i, ...)$ and $P_{h_i^j}(g_i|g_k, ...) = P_{h_k^l}(g_k|g_i, ...)$ where h_i^j represents a constraint denoted as the j-th constraint of the i-th agent. □

By using the above approach, local optimization problems for each agent can then be defined as follows:

$$\min_{g_i \in \mathbb{R}^{n_i}} F_i(g_i, \beta_i|[g]_i, [x]_j) \quad (21)$$

with the optimal solution denoted by

$$[g_i^*|\beta_i, [g]_i] = \arg \min_{g_i \in \mathbb{R}^{n_i}} F_i(g_i, \beta_i|[g]_i, [x]_j) \quad (22)$$

It is simple to show (Inalhan *et al.* [2002]) that, for a local optimization, the values of F_i e P_i decrease as β_i decreases. As a consequence, each agent can use the local β_i as a selection tool to achieve possibly less constraints violation (and indirectly, more cooperation) without resulting in an increase in $F_i(g_i, \beta_i|[g]_i)$.

In this work, a particular policy is used to select β_i over the iterations. At the beginning of the optimization process, β_i are initialized as

$$\beta_i(0) = \alpha_i, \forall i \in \mathcal{A} \quad (23)$$

with α_i the same as in (14). Therefore, at each iteration they are all decreased by the same factor $\lambda \in (0, 1)$

$$\begin{aligned} \beta_i(1) &= \lambda\beta_i(0), \forall i \in \mathcal{A} \\ &\vdots \\ \beta_i(k+1) &= \lambda\beta_i(k) = \lambda^k\beta_i(0), \forall i \in \mathcal{A} \end{aligned} \quad (24)$$

In this way, all β_i decrease with the same speed and this fact is instrumental to prove some properties of the method (see in the next section).

Moving from these considerations, an algorithm that combines local subsystem optimizations with a bargaining scheme between subsystems can be implemented. Such a scheme uses iterative optimizations locally, where each agent i solves a sequence of local programs involving its neighborhood $\mathcal{N}_i$ only. In particular, at each iterate it optimizes the cost (21) by a local selection of β_i which is then made aware to all other agents $j \in \mathcal{N}_i$.

During the evolution of the distributed optimization process, the subsystems are actually negotiating: 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. The selection of β_i gives each subsystem a way 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.

Below, a pseudo-code implementing the algorithm for the generic agent i with neighborhood $\mathcal{N}_i$ is reported

Algorithm 1 (Distributed Optimization)

INPUTS: x_i, $[x]_i$, $[g]_i$, g_i
OUTPUTS: $g_i^{(k)}$
initialization
[1.1] SET $k = 0$, $C_i = 0$, $g_i^{(0)} = g_i$, $[g(0)]_i = [g]_i$, $\beta_i(0) = \alpha_i$
main:
[1.1]IF $(C_i == 0)$
[1.1.1] IF $k > 0$
[1.1.1.1] RECEIVE $g_j^{(k)}$ FROM EACH $j \in \mathcal{N}_i$
[1.1.1.2] SET $g_j^{(k)} = g_j^{(k-1)}$ FOR AGENTS THAT HAVE NOTIFIED CONVERGENCE
[1.1.1.3] COMPOSE $[g(k)]_i$
[1.1.2] SELECT $\beta_i(k+1) = \lambda\beta_i(k)$
[1.1.3] $g_i^{(k+1)} = \text{argmin}\, F_i(g_i^{(k)}, \beta_i(k+1)|[g]_i(k), [x]_i)$
[1.1.4] $\Delta F_i(k+1) = F_i(g_i^{(k)}, \beta_i(k)|[g]_i(k)) - F_i(g_i^{(k+1)}, \beta_i(k+1)|[g]_i(k))$
[1.1.5] IF $\Delta F_i(k) < \epsilon$
[1.1.5.1] SET $C_i = 1$
[1.1.5.2] NOTIFY LOCAL CONVERGENCE TO $\mathcal{N}_i$
[1.1.6] SET $k = k + 1$
[1.1.7] TRANSMIT $g_i^{(k)}$ TO $\mathcal{N}_i$
[1.1.8] GO TO **main**

Remark 4.2. In the original algorithm of (Inalhan *et al.* [2002]), at each iteration and within a certain polling time T_i, agents generate a new optimizing thread whenever new data arrive. In our implementation, thread generation is avoided by assuming that at the beginning of each iteration agents have all the needed information from their neighbors. In this way the algorithm here presented keeps the same properties as the original one. □

Further details on the above algorithm can be found in (Inalhan *et al.* [2002]). Basically, under Assumptions (**A2**) and (**A3**), it can be proved that Algorithm 1 will terminate in a finite number of iterations and return a Pareto Optimal solution for the centralized problem (13) that is ϵ-close to satisfying each of the constraints $h_j(g) \leq 0, \forall j = 1, ..., z$, $\epsilon > 0$.

Proofs of the above propositions can be found in (Inalhan *et al.* [2002]).

4.2 Main Results

The above described formulation allows us to present the main properties of the following DI-CG algorithm when executed at each time instant by all agents.

Algorithm 2 (Distributed Iterative CG Algorithm (DI-CG) - Agent i)

AT EACH TIME t
1.1 RECEIVE $g_j(t-1)$, $x_j(t)$ FROM EACH $j \in \mathcal{N}_i$
1.2 BUILD VECTORS $[g]_i(t-1)$ AND $[x]_i(t)$
1.2 COMPUTE $g_i(t)$ BY MEANS OF **Algorithm 1** WITH INPUTS $x(t)$, $[x(t-1)]_i$, $[g(t-1)]_i$, $g_i(t-1)$
1.3 APPLY $g_i(t)$

1.4 TRANSMIT $g_i(t)$ AND $x_i(t)$ TO $\mathcal{N}_i$

Finally, for the above stated DI-CG scheme its main properties can be summarized in the following Theorem.

Theorem 1. Consider the distributed **DI-CG** Algorithm (**Algorithm 2**) and let assumptions **A1-A3** be fulfilled and $\mathcal{V}(x(0))$ be non empty at time $t = 0$. Then

1) At each time $t \in \mathbb{Z}_+$, the global action of all agents in $\mathcal{A}$ solving **Algorithm 2**, and in turn **Algorithm 1**, generates a feasible solution g^* for problem (13).
2) At each time $t \in \mathbb{Z}_+$, the global action of all agents in $\mathcal{A}$ solving **Algorithm 2**, and in turn **Algorithm 1**, leads to a PO solution $g^*(t)$ for Problem (13) in a finite time. In particular, if the termination criterium of the algorithm are satisfied by all agents at the same iteration step, the achieved solution is also optimal for the associated problem (12) characterized by a weighting matrix Ψ of the form (14).
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 $g(t) = [g_1^T(t),\ldots,g_N^T(t)]^T$ computed by means of **Algorithm 2** converges in finite time to a PO stationary (constant) solution for the following problem

$$\hat{r} := \arg\min_{g}[f_1(g_1,r_1),\ldots,f_1(g_i,r_i),\ldots,f_1(g_N,r_N)] \quad \text{s. t. } g=[g_1^T,..,g_i^T,...,g_N^T]^T\in\mathcal{W}_\delta \tag{25}$$

which is given by $\hat{r} = r$, whenever $r \in \mathcal{W}_\delta$, or by any other PO solution $\hat{r} \in \mathcal{W}_\delta$ otherwise.

For space reasons the proof is missing. However it will be available in a journal version of the paper as soon as possible. □

5. AN EIGHT-TANK WATER DISTRIBUTION SYSTEM APPLICATION

Consider the water tank network depicted in Figure 3. The system consists of the interconnection of four cascaded two-tank models. Each cascaded subsystem is described by the following non-linear equations

$$\begin{cases} \rho S_i^1 \dot{h}_i^1 = -\rho A_i^1 \sqrt{2gh_i^1} + u_i \\ \rho S_i^2 \dot{h}_i^2 = -\alpha_i \rho A_i^2 \sqrt{2gh_i^2} + \rho A_i^1 \sqrt{2gh_i^1} + \sum_{j\in\mathcal{S}^i} \alpha_j \rho A_j^1 \sqrt{2gh_i^1} \end{cases}$$

where u_i is the water flow supplied by the pump whose command is the voltage V_i, $i \in \mathcal{A} := \{1,..,4\}$. Moreover, for each $q = 1,2$, $i = 1,...4$, $S_i^q = 2500\ [cm^2]$ are the tank cross sectional area, h_i^q, the water levels in the tanks, $A_q^i = 8\ [cm^2]$ the sections of pipes connecting the tanks, and $g = 980[cm/sec^2]$ and $\rho = 10^{-3}\ [Kg/cm^3]$ the gravity constant and the water density respectively. Finally, parameters $\alpha_1 = 1, \alpha_2 = \alpha_3 = \alpha_4 = 1/2$ model the splitting water flows between upper and lower tanks. With $\mathcal{T}^i$ we denote the set of subsystems which provide water to the downstream tank of the i-th subsystem; in our case $\mathcal{T}^1 := \{2\}$, $\mathcal{T}^2 = \{3\}$, $\mathcal{T}^3 = \{4\}$ and $\mathcal{T}^4 = \emptyset$. 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 properly modifying their set-points and by exchanging relevant data with the other agents by means of a communication network.

Local decentralized tracking LQ output feedback controllers (Zhu and Pagilla [2006]) are implemented, which act properly on the incoming water flows $u_i(t)$ so that an offset property is satisfied on the tracking error. A simple static equation is used to model the relationship between the input voltage $V_i(t)$ and the incoming mass of water.

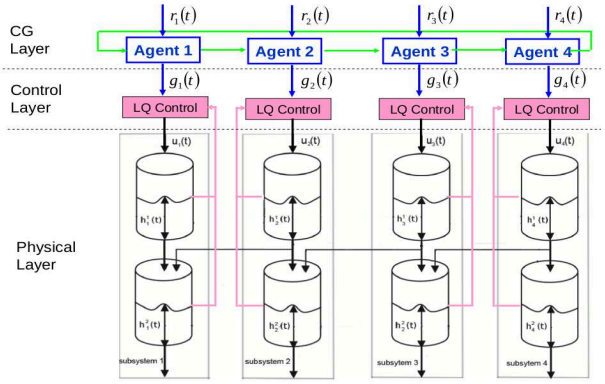

Fig. 3. A four cascaded two-tank water system.

$$u_i(t) = \begin{cases} V_i(t) & if\ V_i(t) \geq 0 \\ 0 & if\ V(t) < 0 \end{cases}$$

The following local and global constraints are to be enforced at each time instant

$$1[cm] \leq h_i^1 \leq 80[cm],\quad 1[cm] \leq h_i^2 \leq 70[cm],\quad 0 \leq V_i \leq 4[V],\ \forall i \in \mathcal{A},$$
$$|h_1^1 - h_2^1| <= 5\ [cm],\ |h_2^1 - h_3^1| <= 5\ [cm],\ |h_3^1 - h_4^1| <= 5\ [cm]$$

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 $T_c = 0.8\ sec.$ The reported simulations investigate the behavior of the overall system when the desired set-points to the water levels of the downstream tanks have the profiles depicted in Figure 5 (red dashed line).

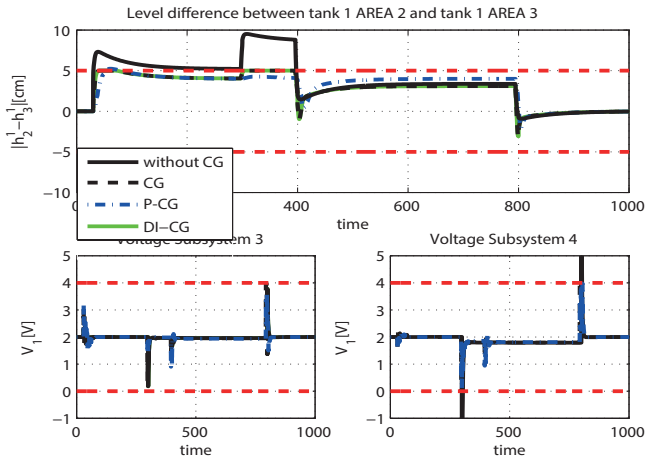

Fig. 4. Applied Voltages

Three CG based supervision methods have been considered: the standard centralized CG (Bemporad *et al.* [1997]) scheme, the parallel non-cooperative non-iterative distributed CG (P-CG) scheme presented in Tedesco *et al.*.B [2012], Casavola *et al.* [2014], and the proposed DI-CG scheme. The local functional costs considered by the distributed strategies are $J_i(t) = \|g_i - r_i(t)\|^2, \forall i \in \mathcal{A}$ while, the centralized CG scheme minimizes the global performance index $J(t) = \sum_i^4 J_i(t)$.

In Figure 4, some components of the constrained vector response can be observed. It is important to note how such a vector violates the constraints at several time

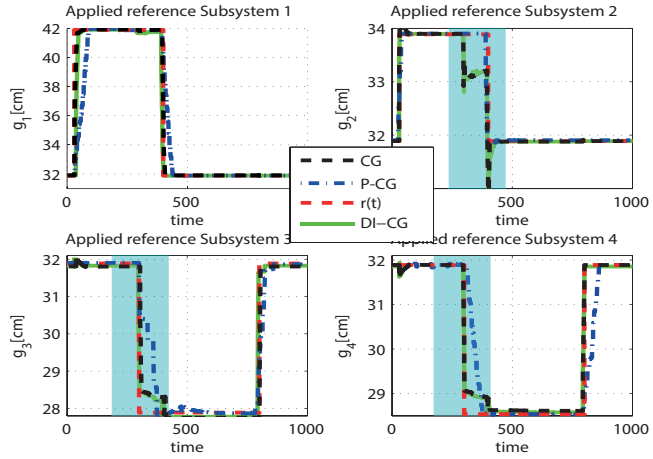

Fig. 5. CG actions

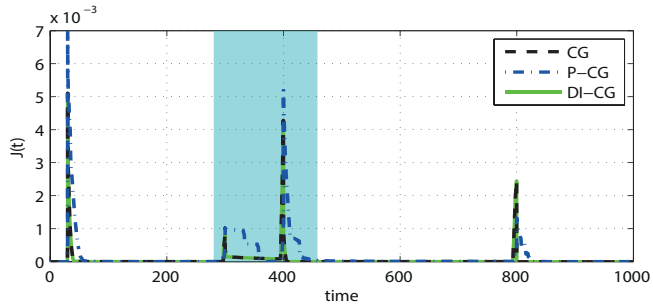

Fig. 6. Performance index $J(t)$

instants when no CG unit is used. On the contrary, this never happens when a CG unit is used. In particular, the responses of the system governed by the classical centralized CG (Bemporad *et al.* [1997]), the P-CG (Casavola *et al.* [2014], Tedesco *et al.*.B [2012]) and the DI-CG schemes are all reported for comparisons.

In Figure 5 the various CG actions are reported. The standard CG centralized scheme has the better performance and, as expected, P-CG exhibits the slowest response to changed conditions. Nevertheless, the related performance, especially during the equilibrium phases, are quite good even if compared to the centralized algorithm. The proposed DI-CG scheme outperforms P-CG and exhibits a behavior very similar to that pertaining to the standard centralized approach. In order to better appreciate this aspect, in Figure 6 the above strategies have been compared in terms of the performance index $J(t)$. From that Figure strongly arises the cooperative nature of the DI-CG in contrast with the non-cooperative behavior of P-CG. In fact, DI-CG is able to generate PO solutions also in the transient phase. This can be observed in the time interval $[300-450]s$, where the prescribed references $r_i(t)$ is time-varying, by noticing that the performance index related to DI-CG is very close to that associated to CG. On the contrary, P-CG approaches a PO solution only during the equilibrium phases.

6. CONCLUSIONS

In this work a novel cooperative iterative distributed CG scheme has been described for the supervision of dynamically coupled interconnected linear systems subject to local and global constraints and used for solving constrained coordination problems arising in networked control systems. The resulting distributed iterative coordination algorithm has been outlined and its main properties concerning optimality, stability and feasibility highlighted. Moreover, in a final example the performance of the cooperative scheme has been compared with those pertaining to both centralized and a non-cooperative distributed schemes.

With respect to the latter approaches, the proposed scheme achieves better performance than the non- cooperative distributed scheme and is even able, under certain conditions, to achieve the same solution of the centralized scheme with similar computational burdens.

REFERENCES

A. Bemporad, A. Casavola and E. Mosca (1997). "Nonlinear Control of Constrained Linear Systems via Predictive Reference Management", *IEEE Trans. Automat. Control*, 42, pp. 340-349.

A. Casavola, E. Garone, F. Tedesco (2011-A), "Distributed Coordination-by-Constraint Strategies for Multi-agent Networked Systems", *IEEE CDC 2011*, Orlando, USA.

A. Casavola, E. Garone and F. Tedesco (2011-B), "A Liveliness Analysis of a Distributed Constrained Coordination Strategy for Multi-Agent Linear Systems", *Proc. of IEEE ECC-CDC*, Orlando, USA.

A. Casavola, F. Tedesco, E. Garone (2014), "The Distributed Command Governor approach in a nutshell", in *Distributed MPC Made Easy*, Chapter 14, Springer - Negenborn R.R. and Maestre J. (Eds.), pp. 259-274.

A. Chinchuluun, P.M. Pardalos, A. Migdalas, L. Pitsoulis (2008), *Pareto optimality, game theory and equilibria*. Springer.

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*, Las Vegas, NV.

I. Kolmanovsky and E. G. Gilbert (1995). "Maximal output admissible sets for discrete-time systems with disturbance inputs". *ACC*, Seattle, pp. 1995-1999.

K. M. Miettinen (1999), *Nonlinear Multiobjective Optimization*, Kluwer Academic.

R. Scattolini (2009), "Architectures for distributed and hierarchical model predictive control - a review", *Journal of Process Control*, vol. 19(5): 723–731.

F. Tedesco, A. Casavola, E. Garone (2012-A), "Distributed Command Governor Strategies for Constrained Coordination of Multi-Agent Networked Systems", *Proc. of IEEE American Control Conference*, Montreal, Canada.

F. Tedesco , A. Casavola , E. Garone (2012-B), "A Distributed parallel Command Governor strategy for the coordination of Multi-agent Networked Systems", *NMPC*, Noordwijkerhout (Netherlands).

F. Tedesco, A. Casavola, E. Garone (2012-C), "A Parallel Distributed Coordination-by-Constraint Strategy for Multi-agent Networked Systems", *IEEE CDC 2012*, Maui Hawaii (USA).

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):57-6.