Proceedings of the 1st IFAC Workshop on Estimation and Control of Networked Systems, Venice, Italy, September 24-26, 2009

Preprint

Distributed Coordination-by-Constraint Strategies for Networked Control Systems

Emanuele Garone * **Francesco Tedesco** * **Alessandro Casavola** *

* *Dipartimento di Elettronica, Informatica e Sistemistica, Universitá della Calabria, Via P.Bucci, 42-C, Rende (CS), 87036, ITALY*

Abstract: In this paper we present preliminary ideas on how to develop distributed supervision strategies for networked control systems subject to coordination constraints to be enforced on-line. Such a coordination paradigm, hereafter referred to as *coordination-by-constraint*, 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. In order to evaluate the distributed method here proposed, the distributed coordination of coupled autonomous vehicles under input-saturation and formation accuracy constraints is presented as an example.

1. INTRODUCTION

The problem of interest here is the design of distributed supervision strategies based on Command Governor (CG) ideas for multi-agent systems where the use of a centralized coordination unit is impracticable because requiring unrealistic or unavailable communication infrastructures. Examples of relevant applications include groups of vehicles cooperatively converging to a desired formation (Keviczky [2005]), large scale chemical processes (Venkat [2006]) and coordination of generators in networked power systems (Kumar and Kothari [2005]) to mention a few. From a theoretical point of view, distributed control policies for dynamically coupled systems have been recently studied in Camponogara et al. [2002], D'Andrea and Dullerud [2003], Venkat et al. [2005], Magni and Scattolini [2006], Dumbar [2007].

The CG approach (see Gilbert et al. [1995], Casavola et al. [2006]) is a well known and established methodology that provides a simple and effective way to enforce pointwise-in-time constraints along the trajectories of a closed-loop system. Namely, the CG is a nonlinear device which is added to a pre-compensated control system that, whenever necessary and on the basis of the knowledge of the actual measured state, modifies the reference to the closed-loop system so as to avoid constraints violation. In this paper, we will introduce a new solution to the CG problem, hereafter referred to as the *Steady-State* CG (SS-CG) approach, that, at the price of some additional conservativeness, is able to accomplish the CG task in 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, then the state will not differ too much from the steady-state equilibrium. Clearly, the scheme remains a *closed-loop* strategy, because the reference modification is undertaken on the basis of the *expected* value of the current state. Extensions to the case of systems subject to bounded disturbances can be directly obtained by following the standard lines of (Casavola et al. [2000]) and will be reported in future works.

The peculiarities of the proposed SS-CG scheme make it an attractive solution for distributed frameworks because alleviate the need to make the entire aggregate state known to all agents at each time instant, the latter could being unrealistic or requiring unrealistic communication infrastructures in some large scale applications. In this paper we will present two preliminary communication-based distributed supervisory algorithms. Feasibility and stability of the presented approaches will be highlighted.

2. PROBLEM FORMULATION

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

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

where: $t \in \mathbb{Z}_+$, $x_i \in \mathbb{R}^{n_i}$ is the state vector (which includes the controller states under dynamic regulation), $g_i \in \mathbb{R}^{m_i}$ is 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

$$c_i(t) \in \mathcal{C}_i, \ \forall t \in \mathbb{Z}_+, \tag{2}$$

$\mathcal{C}_i$ being a convex and compact 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 $x = [x_1^T, \dots, x_N^T]^T \in \mathbb{R}^n$, with $n = \sum_{i=1}^N n_i$, and $g = [g_1^T, \dots, g_N^T]^T \in \mathbb{R}^m$, with $m = \sum_{i=1}^N m_i$. Moreover, we denote by $r = [r_1^T, \dots, r_N^T]^T \in \mathbb{R}^m$, $y = [y_1^T, \dots, y_N^T]^T \in \mathbb{R}^m$ and $c = [c_1^T, \dots, c_N^T]^T \in \mathbb{R}^{n^c}$, 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

$$\Phi = \begin{pmatrix} \Phi_{11} & \dots & \Phi_{1N} \\ \vdots & \ddots & \vdots \\ \Phi_{N1} & \dots & \Phi_{NN} \end{pmatrix}, G = \begin{pmatrix} G_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & G_N \end{pmatrix},$$
$$H^y = \begin{pmatrix} H_1^y \\ \dots \\ H_N^y \end{pmatrix}, H^c = \begin{pmatrix} H_1^c \\ \dots \\ H_N^c \end{pmatrix}, L = \begin{pmatrix} L_1 \\ \dots \\ L_N \end{pmatrix}.$$

It is further assumed that
A1. The overall system (3) is asymptotically stable.
A2. System (3) is off-set free i.e. $H^y(I_n - \Phi)^{-1}G = I_m$.

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. $c_i(t) \in \mathcal{C}_i, \forall t \in \mathbb{Z}_+, i \in \mathcal{A}$.

Classical centralized solutions of the above stated CG design problem (see Bemporad et al. [1997], Casavola et al. [2000]) have been achieved by finding, at each time t, a CG action $g(t)$ as a function of the current reference $r(t)$ and measured state $x(t)$

$$g(t) := \underline{g}(r(t), x(t)) \quad (4)$$

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. Here we will focus on a slight different approach to the CG design problem in which the explicit dependence on the state vector disappears. This is a convenient solution to be used in a decentralized environment because it eliminates the need to share the state vector amongst the agents that, as well known (see Negenborn et al. [2008], Dumbar [2007]), is one of the main difficulties in defining decentralized schemes. Such an approach, hereafter referred to as *Steady-State* CG (SS-CG), will be described in next sections. The main idea is that, if the manipulable reference signal $g(\cdot)$ is "slow enough" w.r.t. system dynamics, then, because of **A1** and **A2**, the state $x(t)$ will not differ too much from the closed-loop steady-state equilibrium that would correspond to the application of a constant set-point $g(t-1)$ for a sufficient number of steps, i.e.

$$x(t) \simeq x_{g(t-1)} := (I_n - \Phi)^{-1}Gg(t-1) \quad (5)$$

The latter allows us to replace the dependence on the measured state $x(t)$ with $x_{g(t-1)}$ in (4). Moreover, because $x_{g(t-1)}$ univocally depends on the command signal applied at the previous time step $g(t-1)$, we can finally reformulate the Steady-State CG problem as the one of finding a command signal $g(t)$ as a function of $r(t)$ and $g(t-1)$

$$g(t) = \underline{g}(r(t), g(t-1)) \quad (6)$$

where $g(t)$ is the best approximation of $r(t)$ to be computed as in the standard CG approach with the additional requirement that, at the next sampling time, the state $x(t+1)$ will be not far away from the new steady state solution, i.e. $x(t+1) \simeq x_{g(t)}$.

3. THE STEADY-STATE CG APPROACH

In order to make precise statements and comparisons, we describe the basic CG centralized approach of Bemporad et al. [1997], Casavola et al. [2000] first. For a constrained closed-loop system of the form (3) satisfying assumptions **A1-A2**, the standard CG design problem (4) can be solved by introducing, for a given $\delta > 0$, the sets:

$$\begin{aligned} \mathcal{C}^\delta &:= \mathcal{C} \sim \mathcal{B}_\delta \\ \mathcal{W}^\delta &:= \{g \in \mathbb{R}^m : c_g \in \mathcal{C}^\delta\} \end{aligned} \quad (7)$$

where $\mathcal{B}_\delta$ is the ball of radius δ centered at the origin and $\mathcal{A} \sim \mathcal{E}$ is the Pontryagin set difference defined as $\{a : a + e \in \mathcal{A}, \forall e \in \mathcal{E}\}$. In particular, $\mathcal{W}^\delta$, which we assume non-empty, is the convex and closed set of all constant commands g whose corresponding equilibrium points $c_g := H^c(I_n - \Phi)^{-1}Gg + Lg$ satisfy the constraints with margin δ. Let introduce also the *virtual evolutions* of the c-variable

$$\hat{c}(k, x(t), g(t)) := H^c\left(\Phi^k x(t) + \sum_{i=0}^{k-1} \Phi^{k-i-1} G g(t)\right) + Lg(t) \quad (8)$$

along the *virtual time* k, from the initial condition $x(t)$ at time $k = 0$ under the application of a constant command $g(t)$, $\forall k$. Then, for any given state x, we can define

$$\mathcal{V}(x) = \{g \in \mathcal{W}^\delta : \hat{c}(k, x, g) \in \mathcal{C}, \forall k \in \mathbb{Z}_+\}. \quad (9)$$

As a consequence, $\mathcal{V}(x)$ represents, if non-empty, the set of all constant commands in $\mathcal{W}^\delta$ whose virtual c-evolutions starting from x at virtual time $k = 0$ satisfy the constraints also during transients. Then, the standard CG design problem can be solved by the following algorithm

The standard CG Algorithm
REPEAT AT EACH TIME t

1.1 SOLVE

$$g(t) = \arg \min_{g \in \mathcal{V}(x(t))} \| g - r(t) \|_\Psi^2 \quad (10)$$

1.2 APPLY $g(t)$

The idea underlying the SS-CG approach is that of ensuring that any admissible variation of the manipulated reference $g(\cdot)$ always produces a guaranteed bounded perturbation on the actual closed-loop state around a suitable feasible equilibrium state. Such a property is ensured if the following technical expedients are adopted during the SS-CG computation:

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

$$g(t) - g(t-\tau) \in \Delta\mathcal{G}, \quad (11)$$

where the integer $\tau > 0$ and the closed and convex set $\Delta\mathcal{G} \subset \mathbb{R}^m$ are computed from the outset as detailed below. Let us rewrite the virtual evolution of the c-variable (8) as

$$\hat{c}(k, x(t), g(t)) = c_{g(t)} + H^c\Phi^k(x(t) - x_{g(t)}). \quad (12)$$

The above expression shows that the predictions can be divided in two amounts: a steady-state component represented by $c_{g(t)}$ and the transient evolution $H^c\Phi^k(x(t) -$

$x_{g(t)}$). Then, because $g(t) \in \mathcal{W}^\delta$ and, in turn, $c_{g(t)} \in \mathcal{C}^\delta$ at each time t, the key idea behind the SS-CG approach is that constraints can be satisfied, although in a quite arbitrary and conservative way, by only taking care of the transient component, on which the following condition

$$\|H^c\Phi^k(x(t) - x_{g(t)})\| \leq \delta, \forall k \geq 0 \tag{13}$$

has to be enforced. One way to achieve (13) without assuming state availability is based on the idea that, after a sufficient long time after from the application of a new SS-CG command, the transient contribution decreases and can be bounded within a certain percentage of its initial magnitude. More formally, let us introduce the following notion of *dwelling time*:

Definition *(Dwelling Time)* - The integer $\tau > 0$ is said to be the *dwelling time with parameter* γ, $0 < \gamma < 1$, for the pair (H^c, Φ), if it is the smallest integer such that

$$\begin{array}{c} \|H^c\Phi^k x\| \leq M(x), \quad \forall k \geq 0 \\ \Downarrow \\ \|H^c\Phi^{\tau+k} x\| \leq \gamma M(x), \forall k \geq 0 \end{array} \tag{14}$$

holds true for each $x \in \mathbb{R}^n$ with the real $M(x) > 0$ any arbitrarily chosen upper-bound. □

Such a definition implies that if at time $t - \tau$ a certain command $g(t - \tau)$, such that

$$\| H^c\Phi^k(x(t-\tau) - x_{g(t-\tau)}) \| \leq \delta, \ \forall k \geq 0 \tag{15}$$

is constantly applied to the system, the transient contribution from t onwards can be bounded as follows

$$\| H^c\Phi^k(x(t) - x_{g(t-\tau)}) \| \leq \gamma\delta, \quad \forall k \geq 0 \tag{16}$$

because the following relationship

$$\Phi^\tau(x(t-\tau) - x_{g(t-\tau)}) = (x(t) - x_{g(t-\tau)}) \tag{17}$$

obviously holds true because of the above arguments. Consider now the transient contribution to the constrained vector at time t depending on the new SS-CG command $g(t)$ to be determined

$$\| H^c\Phi^k(x(t) - x_{g(t)}) \| . \tag{18}$$

The latter, by introducing the τ-step incremental vector $\Delta x_{g(t)} = x_{g(t)} - x_{g(t-\tau)}$, can be rewritten as

$$\| H^c\Phi^k(x(t) - x_{g(t-\tau)}) - H^c\Phi^k \Delta x_{g(t)} \| . \tag{19}$$

Moreover, if at time instant $t - \tau$, a SS-CG command $g(t - \tau)$ complying with (15) is applied, then one has

$$\begin{array}{r} \|H^c\Phi^k(x(t) - x_{g(t-\tau)}) - H^c\Phi^k \Delta x_{g(t)}\| \leq \\ \leq \gamma\delta + \|H^c\Phi^k \Delta x_{g(t)} \| . \end{array} \tag{20}$$

The latter allows one to simplify the SS-CG design control problem into the one of selecting, every τ steps, a new command $g(t)$ satisfying

$$\| H^c\Phi^k \Delta x_{g(t)} \| \leq (1 - \gamma)\delta, \forall k \geq 0. \tag{21}$$

Finally, because Δx_g univocally depends on $\Delta g(t) = g(t) - g(t - \tau)$, we can formulate the Steady-State CG algorithm as follows.

The SS-CG Algorithm

REPEAT AT EACH TIME $t = \kappa\tau, \kappa = 0, 1, \ldots$

1.1 SOLVE

$$g(t) = \arg\min_g \| g - r(t) \|^2_\Psi \tag{22}$$

$$subject\ to: \begin{cases} g \in \mathcal{W}^\delta \\ (g - g(t-\tau)) \in \Delta\mathcal{G} \end{cases} \tag{23}$$

1.2 APPLY $g(t)$

where $\Psi = \Psi^T > 0$ is a weighting matrix and $\Delta\mathcal{G}$ is the closed and convex set of all the possible τ-step incremental commands ensuring the inequality (21) to hold true:

$$\Delta\mathcal{G} = \{\Delta g : \|H^c\Phi^k(I - \Phi)^{-1}G\Delta g\| \leq (1-\gamma)\delta, \forall k \geq 0\}. \tag{24}$$

It is worth to note that the sets $\mathcal{W}^\delta$, $\Delta\mathcal{G}$ and the dwelling time τ can be computed off-line from the outset. The following main properties can be proved for the above described SS-CG strategy Casavola et al. [2009]

Proposition *1.* - Let assumptions **A1-A2** be fulfilled. Consider system (3) along with the **SS-CG** selection rule and let an admissible command signal $g(0) \in \mathcal{W}^\delta$ be applied at $t = 0$ such that (13) holds true. Then:

(1) the minimizer in (22) uniquely exists every τ steps and can be obtained by solving a convex constrained optimization problem;
(2) constraints are fulfilled for all $t \in \mathbb{Z}_+$;
(3) the overall system is asymptotically stable and whenever $r(t) \equiv r$, the sequence of $g(t)$ converges in finite time either to r or to its best steady-state admissible approximation: $g(t) \to \hat{r} := \arg\min_{g \in \mathcal{W}^\delta} \|g - r\|^2_\Psi$. □

4. DISTRIBUTED SS-CG

Here we will focus on two distributed CG schemes where agents are connected by a communication network. Such a network is modeled by means of a *communication graph*: an undirected graph $\mathcal{G} = (\mathcal{A}, \mathcal{B})$, where $\mathcal{A}$ denotes the set of the N subsystems and $\mathcal{B} \subset \mathcal{A} \times \mathcal{A}$ the set of edges representing communication links amongst agents. More precisely the edge (i, j) will belong 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 τ sampling times. The communication graph is assumed to be connected, i.e. for each couple of agents $i \in \mathcal{A}, j \in \mathcal{A}$ it exists at least one sequence of edges connecting i and j and the minimum number of edges connecting two agents will be denoted by $d_{i,j}$. The set of the agents with a direct connection with the i-th agent will be referred to as *Neighborhood of the i-th agent* $\mathcal{N}_i = \{j \in \mathcal{A} : d_{i,j} = 1\}$.

4.1 Sequential Procedure (S-SSCG)

Let $\mathcal{G}$ be an Hamiltonian graph and, without loss of generality, the sequence $\mathcal{H} = \{1, 2, ..., N-1, N\}$ an Hamiltonian cycle. The idea behind the approach is that only one agent per decision time is allowed to manipulate its local command signal $g_i(t)$ while all others are instructed to hold their previous values. After each decision, 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 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 SS-CG algorithm:

Sequential-SSCG Algorithm (S-SSCG) *- Agent i*

REPEAT AT EACH TIME $t = \kappa\tau, \kappa = 0, 1, \ldots$

1.1 IF $(\kappa mod N) == i$

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

1.1.2 SOLVE

$$\begin{aligned} &g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel^2_{\Psi_i} \\ &subject\ to: \\ &\begin{cases} g(t) = [g_1^T(t-\tau),..,g_i^T,...,g_N^T(t-\tau)]^T \in \mathcal{W}^\delta \\ (g_i - g_i(t-\tau)) \in \Delta\mathcal{G}_i^0 \end{cases} \end{aligned} \quad (25)$$

1.1.3 APPLY $g_i(t)$

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

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

1.2 ELSE

1.2.1 APPLY $g_i(t) = g_i(t-\tau)$

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

$$\Delta\mathcal{G}_i^0 = \left\{ \Delta g_i \,\middle|\, [0, 0, \ldots, \Delta g_i^T, \ldots, 0]^T \in \Delta\mathcal{G} \right\}$$

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

Proposition *2.* - Let assumptions **A1-A2** be fulfilled. Consider system (3) as the composition of N subsystems in form (1) along with the distributed **S-SSCG** selection rule 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 (13) holds true. Then

(1) for each agent $i \in \mathcal{A}$ the algorithm produces at each time step a feasible local command $g_i(t)$;
(2) constraints are fulfilled for all $t \in \mathbb{Z}_+$;
(3) the overall system is asymptotically stable. In particular, whenever $r(t) \equiv r$, the sequence of the aggregate vectors $g(t) = [g_1^T(t), \ldots, g_N^T(t)]^T$ converges in finite time either to r or to an admissible approximation in $\mathcal{W}^\delta$ which is Pareto-optimal w.r.t. the local objective functionals $||g_i - r_i||^2_\Psi, i = 1, ..., N$. □

Remark 3 - The proof of Proposition (2) is here omitted for brevity. See Casavola et al. [2009] for details. □

4.2 Parallel SSCG (P-SSCG)

The main drawback of the S-SSCG algorithm is that, every τ time instants, only one agent at a time is allowed to modify its local command. In order to overcome such a limitation, any agent should be enabled to select its local command every τ time instants. The two key points to build up such a kind of strategy are i) the definition of the information available to each agent and ii) the determination of a set of "selection rules" such that the composition of all feasible local commands satisfies the overall constraints (23). Here, 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, the i-th agent has knowledge of the following vector:

$$\xi_i(t-\tau) = [g_1^T(t-d_{i,1}\tau), \ldots, g_i^T(t-\tau), \ldots, g_N^T(t-d_{i,N}\tau)]^T$$

It is important to highlight that, under the above assumption, the most recent information on the applied commands shared by all agents at each decision time t is given by the vector

$$\xi(t-\tau) = [g_1^T(t - d_{max,1}\tau), ..., g_N^T(t - d_{max,N}\tau)]^T$$

where $d_{max,i} = max_{j\in\mathcal{A}} d_{i,j}$. In the following, a simple selection rule is presented based on the idea that agents can autonomously select their own commands whenever their states are in a "safe" area of the constrained feasible set whereas, on the contrary, they have to resort to the Sequential SSCG algorithm when autonomous decisions can yield to hazardous situations. More precisely:

- if the shared information $\xi(t-\tau)$ is such that any possible (past and present) choice of the agent never makes the aggregate command to violate the constraints, i.e $\xi(t-\tau) \in W^{com}$ where $W^{com} \triangleq W^\delta \sim (d_{max,1}\Delta\mathcal{G}_1 \times \ldots \times d_{max,N}\Delta\mathcal{G}_N)$, then each agent can select its local command accordingly to the solution of the following optimization problem

$$\begin{aligned} &g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel^2_{\Psi_i} \\ &subject\ to: \{g_i - g_i(t-\tau) \in \Delta\mathcal{G}_i \end{aligned} \quad (26)$$

  where $\Delta\mathcal{G}_1 \subseteq \mathbb{R}^{m_1}, ..., \Delta\mathcal{G}_N \subseteq \mathbb{R}^{m_N}$ are the sets of possible reference variations for each agent predetermined in order to accomplish $\Delta\mathcal{G}_1 \times ... \times \Delta\mathcal{G}_N \subseteq \Delta\mathcal{G}$ (see Casavola et al. [2009] for details).
- otherwise, after a proper initialization phase, a switch to the "sequential" S-SSCG Algorithm should be undertaken and the commands selected on its basis.

We can then present the following "parallel" version of the distributed SS-CG algorithm:

Parallel-SSCG Algorithm (P-SSCG) *- Agent i*

REPEAT AT EACH TIME $t = \kappa\tau, \kappa = 0, 1, \ldots$

1.1 IF $\xi(t-\tau) \in W^{com}$
  SOLVE (26)
  $\mu = 1$;
  ELSE IF $\kappa mod N \neq i$
  $g_i(t) = g_i(t-\tau)$
  $\mu++$;
  ELSE
  SOLVE

$$\begin{aligned} &g_i(t) = \arg\min_{g_i} \parallel g_i - r_i(t) \parallel^2_{\Psi_i} \\ &subject\ to: \\ &\begin{cases} (g_i - g_i(t-\tau)) \in \Delta\mathcal{G}_i^0 \\ g(t) = [\bar{g}_1^T,..,g_i^T,...,\bar{g}_N^T(t-\tau)]^T \in \mathcal{W}^\delta \\ \qquad\qquad \forall \bar{g}_j \in \bar{\mathcal{G}}_{i,j}(\xi_i(t-\tau), \mu) \end{cases} \end{aligned} \quad (27)$$

  $\mu++$;

1.2 APPLY $g_i(t)$

1.3 UPDATE $\xi_i(t)$

1.4 TRANSMIT $\xi_i(t)$ TO NEIGHBORHOOD $\mathcal{N}_i$

1.5 RECEIVE $\xi_j(t)$ FROM ALL NEIGHBORS $j \in \mathcal{N}_i$

where μ is a counter initialized as $\mu = 1$ that is incremented for each time passed in the sequential mode and $\bar{\mathcal{G}}_{i,j}(\xi_i(t-\tau), \mu)$ is the set of all the possible values the j-th command could have assumed from the i-th viewpoint:

$$\begin{aligned} &\bar{\mathcal{G}}_{i,j}(\xi_i(t-\tau), \mu) = \\ &= \{\bar{g}_j \,|\bar{g}_j - g_j(t - d_{i,j}\tau) \in (\min\{0, d_{i,j} - \mu\})\Delta\mathcal{G}_i\}. \end{aligned} \quad (28)$$

Please note that the introduction of such a set and of the associate condition in the optimization problem (27) forces the initialization of the sequential algorithm. In fact, for $\mu > d_{i,j}$, it turns out that $\bar{\mathcal{G}}_{i,j}(\xi_i(t-\tau), \mu) = \{g_j(t - d_{i,j}\tau)\}$ reduces to a singleton in (28) and the algorithm coincides with the S-SSCG scheme seen in the previous Subsection.

Remark 4 - It has been proved in Casavola et al. [2009] that the same properties pertaining to the S-SSCG scheme and reported in Proposition 2 still apply for the P-SSCG algorithm. Details are here omitted for space reasons.

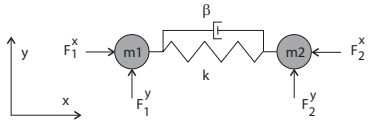

Fig. 1. Planar system of two dynamically coupled masses.

Notice also that other different selection rules can be defined and will be subject of further investigations. □

5. ILLUSTRATIVE EXAMPLE: COORDINATION OF AUTONOMOUS VEHICLES

In Fig. 1, a system consisting of two dynamically coupled masses is presented. Such a system, which will be referred to as the *m2-system*, represents in essence a pair of unmanned vehicles transporting an elastic membrane. The system is described by the following equations

$$\begin{aligned} m_i\ddot{x}_i &= -k(x_i - \sum_{j\in\mathcal{N}_i} x_j) - \beta(\dot{x}_1 - \sum_{j\in\mathcal{N}_i} \dot{x}_j) + F_i^x \\ m_i\ddot{y}_i &= -k(y_i - \sum_{j\in\mathcal{N}_i} y_j) - \beta(\dot{y}_1 - \sum_{j\in\mathcal{N}_i} \dot{y}_j) + F_i^y \end{aligned} \tag{29}$$

where (x_i, y_i), $i \in \mathcal{A} = \{1, 2\}$ are the coordinates of the i-th mass position w.r.t a cartesian reference and (F_i^x, F_i^y), $i \in \mathcal{A}$ the components, along the same reference frame, of the forces acting as inputs for the two slave systems. $\mathcal{N}_i$ denotes the neighborhood of the i-th agent, in this case $\mathcal{N}_1 = \{2\}$ and $\mathcal{N}_2 = \{1\}$. The following system parameters are assumed $\beta = 1$ $[\frac{N\cdot sec}{m}]$, $k = 1$ $[\frac{N}{m}]$, $m_i = 1$ $[Kg]$, $\in \mathcal{A}$ and a sampling time of $T_c = 0.1$ $[sec]$ is employed in the simulations. Each subsystem has been precompensated by an optimal LQ state feedback local controller. The problem we consider here is the coordination of the planar motions of those two masses along two continuously parameterized paths $r_1(\alpha_1) \in \mathbb{R}^2, r_2(\alpha_2) \in \mathbb{R}^2$ where $\alpha_1 \in [0, \bar{\alpha}], \alpha_2 \in [0, \bar{\alpha}]$ are real parameters. It is required that each agent tracks its own path by progressively increasing the value of $\alpha_i(t)$ at the maximum speed complying with the following local and coupling constraints

$$\begin{aligned} |F_i^j(t)| &\le 0.5\ [N] \qquad j = x, y,\ i \in \mathcal{A}, \\ |\nu_i(t) - r_i(\alpha_i(t))| &\le 0.05\ [m] \qquad i \in \mathcal{A}, \\ |\alpha_i(t) - \alpha_j(t)| &\le 0.06, \qquad i \in \mathcal{A}, j \in \mathcal{N}_i. \end{aligned} \tag{30}$$

The first set of inequalities represents input-saturation constraints on the four forces F_i^x and F_i^y, $i \in \mathcal{A}$, acting as inputs of the vehicles. The second set of constraints represents the component-wise accuracy of the vehicle positions $\nu_i(t) = [x_i(t), y_i(t)]$ with respect to the target motion $r_i(\alpha_i(t))$. Finally, the third group of constraints represents the coordination constraints between the agents: the two agents have never to be "too far" in the parameter progression in order to maintain the formation shape. In this example we will consider the rigid motions $r_1(\alpha), r_2(\alpha)$ for $\alpha \in [0, 4]$ reported in Figure 3. and we compare in Figures 3.b -3.d, the real system trajectories achieved by the use of the centralized Standard CG strategy (10) and of the two presented decentralized approaches for the first 1000 simulation steps. Those trajectories corresponds to the evolution of $\alpha_i(t), i = 1, 2$ shown in Figure 4. As expected, the use of decentralized schemes introduces a certain level of conservativeness w.r.t. standard centralized CG approaches. However, the performance, especially for the P-SSCG method, could be adequate in certain applications. In order to evaluate the scalability of the proposed

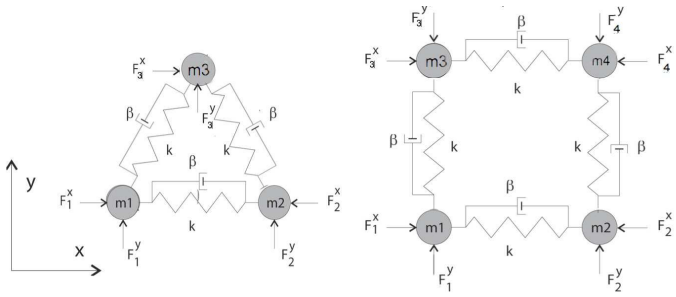

Fig. 2. a) Planar system of three dynamically coupled masses (*m3-system*) b) Planar system of four dynamically coupled masses (*m4-system*)

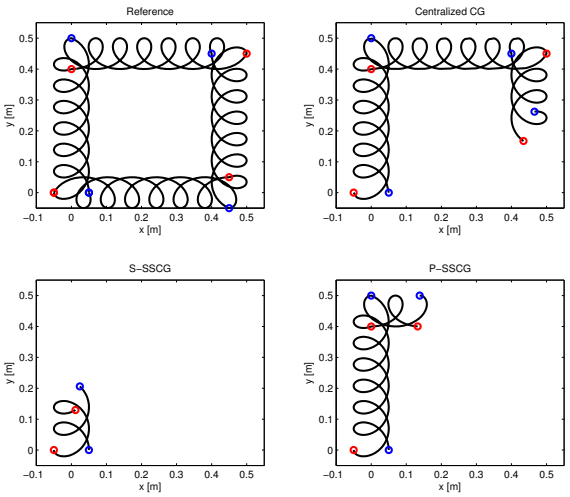

Fig. 3. Tracking trajectories for *m2-system*: a) Reference Trajectories, b) Standard CG, c) S-SSCG d) P-SSCG

algorithms, formations of three and four masses, hereafter denoted as the *m3-system* and the *m4-system*, have been considered with masses on the vertices, of an equilateral triangle for *m3-system* (see Figure 2.a) and of a square for *m4-system* (see Figure 2.a), and coupling elements on the edges. Simulative results on the trajectories are depicted in Figures 5 and Figures 6 respectively. Finally, in Figure 7 some comparisons regarding the amounts of CPU usage are reported. It is possible to note that the standard CG algorithm requires a much higher computational time than the two decentralized counterparts and that the increment of the number of the agents produces an approximately linear increment in the computational burdens.

6. CONCLUSIONS

In this paper, a distributed implementation of CG schemes has been developed for dynamically coupled linear systems subject to local and global constraints. The key point was to resort to a new CG scheme that, thanks to the asymptotical stability of the pre-compensated system, does not need an explicit measure of the state. Two algorithms for the case where communication amongst the agents are allowed have been presented and results on constraints fulfillment and stability highlighted. Comparisons with central solutions have been presented and commented in the final illustrative example. The presented results are encouraging and stimulate further research on the topic.

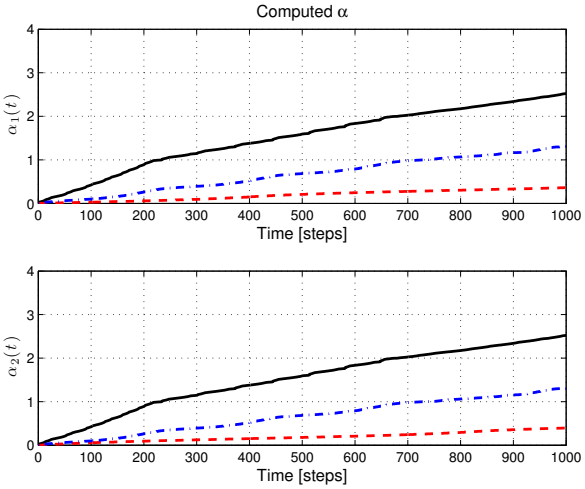

Fig. 4. Computed $\alpha_1(t), \alpha_2(t)$ for *m2-system*: Standard CG (-), P-SSCG (-.-), S-SSCG(- -).

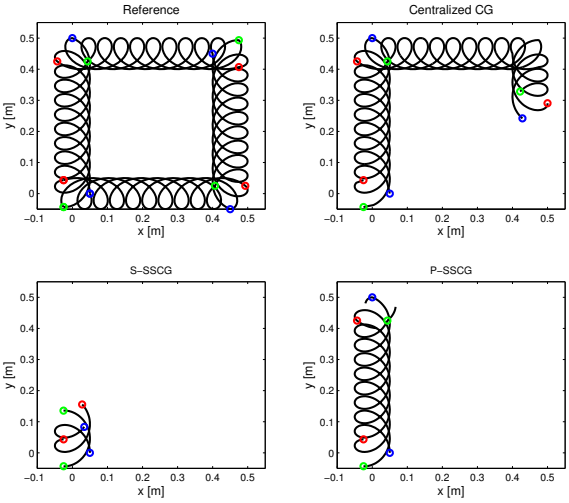

Fig. 5. Tracking trajectories for *m3-system*: a) Reference Trajectories, b) Standard CG, c) S-SSCG d) P-SSCG

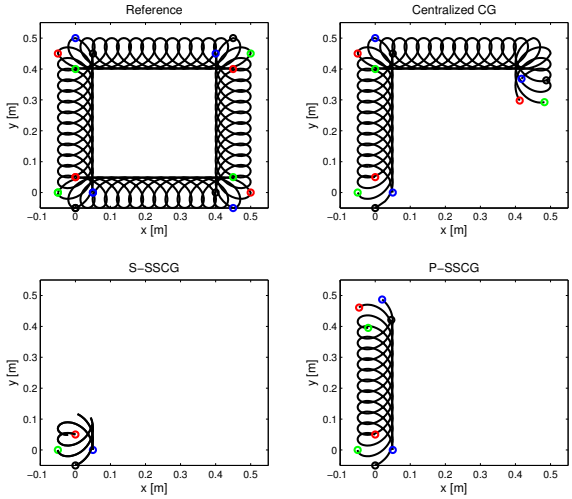

Fig. 6. Tracking trajectories for *m4-system*: a) Reference Trajectories, b) Standard CG, c) S-SSCG d) P-SSCG

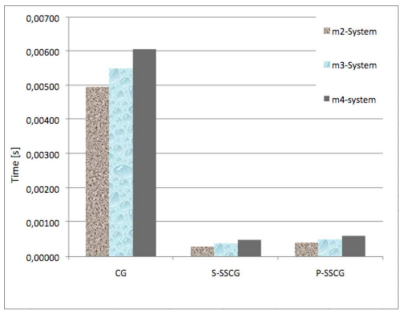

Fig. 7. Mean CPU usage for each agent (sec).

R. R. Negenborn, B. De Schutter, and J. Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Appl. of Artificial Intelligence*, 21(3), pp.353-366, 2008.

L. Magni and R. Scattolini. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*, 42, pp. 1231-1236, 2006.

W.B. Dunbar. Distributed receding horizon of dynamically coupled nonlinear systems. *IEEE Trans. Automat. Control*, 52(7), pp. 1249-1263, 2007.

E.G. Gilbert, I. Kolmanovsky and K. Tin Tan. Discrete-time Reference Governors and the Nonlinear Control of Systems with State and Control Constraints. *International Journal on Robust and Nonlinear Control*, 5, pp. 487-504, 1995.

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

A. Casavola, M. Papini and G. Franzè. Supervision of networked dynamical systems under coordination constraints. *IEEE Trans. Automat. Control*, 51(3), pp. 421-437, 2006.

A. Casavola, E. Mosca and D. Angeli. Robust command governors for constrained linear systems. *IEEE Trans. Automat. Control*, 45, pp. 2071-2077, 2000.

A. Casavola, E. Garone, F. Tedesco. Decentralized Steady State Command Governor, *DEIS-UNICAL, Technical Report*, DEIS-18/02, University of Calabria, 2009.

REFERENCES

T. Keviczky. Decentralized Receding Horizon Control of Large Scale Dynamically Decoupled Systems. *PhD Thesis*, University of Minnesota, USA, 2005.

A.N. Venkat. Distributed Model Predictive Control: Theory and Applications. *PhD Thesis*, University of Wisconsin, USA, 2006.

Ibraheem, P. Kumar, and D.P. Kothari, Recent Philosophies of Automatic Generation Control Strategies in Power Systems, *IEEE Trans. on Power Systems*, 20(1), 2005.

E. Camponogara, D. Jia, B.H. Krogh and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, February, pp. 44-52, 2002.

R. D'Andrea and G.E. Dullerud, Distributed Control Design for Spatially Interconnected Systems. *IEEE Trans. Automat. Control*, Vol. 48(9), pp. 1478-1495, 2003.

A.N. Venkat, J.B. Rawlings and S.J. Wright. Stability and optimality of distributed model predictive control. *IEEE CDC-ECC '05*, Sevilla, Spain, pp. 6680-6685, 2005.