INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust. Nonlinear Control 2017; **00**:1–19
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rnc

A Distributed Command Governor based on Graph Colorability Theory

Alessandro Casavola[1], Emanuele Garone[2], Francesco Tedesco[1]

[1] *Università degli Studi della Calabria - DIMES - Via P. Bucci, Cubo 42 C - 87036 Rende (CS) - Italy*
[2] *Université Libre de Bruxelles, 50 Av. F.D. Roosvelt, B-1050 Brussels - Belgium*

SUMMARY

In this paper a distributed Command Governor (CG) strategy is introduced that, by the use of graph colorability theory, improves the scalability property and the performance of recently introduced distributed non-cooperative sequential CG strategies. The latter are characterized by the fact that only one agent at a decision time is allowed to update its command, while all others keep applying their previously computed commands. The scalability of these early CG distributed schemes and their performance are limited because the structure of the constraints is not taken into account in their implementation. Here, by exploiting the idea that agents that are not directly coupled by constraints can simultaneously update their control actions, the agents in the network are grouped into particular subsets (Turns). At each time instant, on the basis of a round-robin policy, all agents belonging to a turn are allowed to update simultaneously their commands, while agents in other turns keep applying their previous commands. Then, a Turn-Based distributed CG strategy is proposed and its main properties analyzed. Graph colorability theory is used to determine the minimal number of turns and to distribute each agent in at least a turn. A novel graph colorability problem which allows one to maximize the frequency at which agents can update their commands is proposed and discussed. A final example is presented to illustrate the effectiveness of the proposed strategy.

Received ...

1. INTRODUCTION

This paper focuses on the design of a distributed supervision strategy based on multi-agent Command Governor (CG) ideas for networked interconnected systems in situations where the use of a centralized coordination unit is impracticable because requiring unrealistic or unavailable communication and/or computation infrastructures.

The CG approach is a well-known constrained supervision strategy that selects its actions by solving on-line a constrained convex optimization problem based on future system predictions. A CG unit (Figure 1) is a nonlinear device which is added to a plant, regulated by a *primal controller* that is separately designed w.r.t. the CG to guarantee stability and tracking performance in linear regimes. The CG main objective is that of modifying, whenever necessary, the *desired 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 state and input predictions.

*Correspondence to: [1]Università degli Studi della Calabria - DIMES - Via P. Bucci, Cubo 42 C - 87036 Rende (CS) - Italy - email `ftedesco@dimes.unical.it`

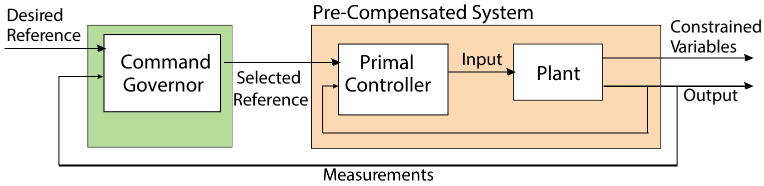

Figure 1. Command Governor basic architecture

In this paper such a supervision design problem is considered in the distributed context depicted in Figure 2. There, the supervisory task is allocated amongst many agents, which are assumed to be able to share information among them. Moreover, each agent is in charge of supervising and coordinating one specific subsystem.

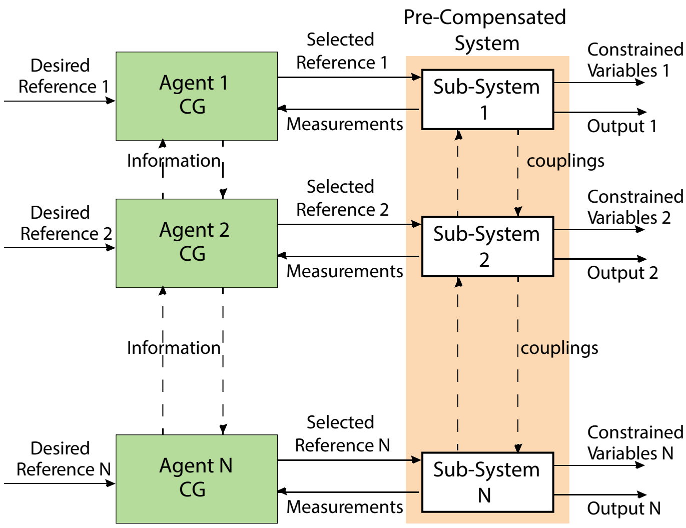

Figure 2. Multi-agent architectures

Several works dealt with the above distributed CG design problem. They were mainly based on the use of non-cooperative game theory concepts and several strategies were singled out. In particular, the *Sequential* CG approach of [6] and [23] is a strategy where, according to a predetermined order, only one agent at each decision time is allowed to update its control action, while all others keep applying their previous applied commands. On the contrary, the *Parallel* CG strategy of [5] is a scheme where all agents update their control actions simultaneously at each decision time.

A further decentralized scheme based on the similar approach denoted as *Reference Governor* has been presented in [15]. For an extensive survey on basic and advanced advanced *Reference Governor* schemes the reader is referred to the survey paper [9].

In this paper, a Turn-Based distributed CG approach is proposed that dramatically improves the performance and the scalability property of the Sequential distributed CG scheme introduced in [23]. To this end, the idea is that agents not jointly involved in coupling constraints can simultaneously update their control actions without violating them. Then, the main contribution of this paper is to show that the use of graph colorability theory is instrumental to systematically group agents into particular subsets (Turns) that allow this kind of parallelization.

More precisely, in this paper we will discuss and highlight the link between the grouping policy and the graph vertex coloring problem applied to the graph describing the couplings induced by the constraints among couples of agents. As well known, the minimal graph vertex coloring problem [12] aims at assigning a color to each vertex of a graph such that no two adjacent vertices may have the same color. The objective is to make this using the fewest possible number of colors. Such a

Prepared using rncauth.cls
DOI: 10.1002/rnc

problem has been widely studied in the last century (see [12] for a survey on the subject). Efficient algorithms and heuristics exist and several interesting results have been proved. Among many, of particular interest here is the fact that, for many classes of graph topologies, the minimal number of colors of a graph is bounded (and often by a small integer) regardless of the number of nodes of the graph.

The use of such graph theoretical concepts allows one to remarkably improve the scalability and the performance of the proposed supervision scheme with respect to the earlier version described in [23]. In the final example we will show that the resources (CPU time and data exchanges) required to accomplish the coordination task do not increase with the number of agents, while the performance remains very close to the one pertaining to centralized solutions.

A very preliminary version of this paper has been presented in [24] for systems having no dynamic interactions. Here, the proposed approach has been generalized to the case where dynamic interactions are present. Furthermore, this paper treats more systematically the connections with the graph colorability theory and formulates the optimal turn-definition problem as a new vertex multi-coloring problem [18]. Such a formulation is crucial to achieve more performing turn configurations of the network where agents may belong to multiple, not conflicting, turns. To the best of our knowledge, the specific problem introduced in this paper has not been studied before. Moreover, being the problem NP-hard, this paper proposes a heuristic to compute a sub-optimal multi-coloring round-robin policy. We also wish to remark that, although in [20] some hints are given along this direction, a formal connection between graph colorability theory and distributed control literature is missing. This paper seems to be first attempts to do so.

The paper is organized as follows. Section 2 introduces some preliminary notations and definitions. In Section 3 the main problem is stated. In Section 4 the proposed Turn-Based distributed CG approach is introduced. In Section 5 the main properties of the proposed approach are discussed. Section 6 highlights the connections with the graph colorability theory. In Section 7 an illustrative example consisting of the coordination of a robots' formation is reported. Some conclusions end the paper.

2. NOTATIONS AND PRELIMINARIES

$\mathbb{R}$, $\mathbb{R}_+$ 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 $x^T \Psi x$. A generic ball in an Euclidean n-space $\mathbb{R}^n$ is defined as $\mathcal{B}_\delta := \{x \in \mathbb{R}^n : \|x\| \leq \delta\}$. Let $g = [g_1^T, .., g_N^T]^T \in \mathbb{R}^m$ be a vector composed by N sub-vectors $g_i \in \mathbb{R}^{m_i}$. Then a generic matrix $A \in \mathbb{R}^{z \times m}$, can be represented either as

$$A = \begin{bmatrix} a_1^1, & a_1^2, & \dots, & a_1^N \\ a_2^1, & a_2^2, & \dots, & a_2^N \\ \vdots & \vdots & \vdots & \vdots \\ a_z^1, & a_z^2, & \dots, & a_z^N \end{bmatrix}$$

with $a_i^j \in \mathbb{R}^{1 \times m_i}$ denoting the j-th sub-vector of the i-th row of A, or as $A = \left[A_1 | A_2 | \dots | A_N\right]$ where $A_i \in \mathbb{R}^{z \times m_i}$ is the i-th sub matrix of A.

Let $A_{ij} \in \mathbb{R}^{n_i \times n_j}, i, j = 1, ..., N$ be a collection of N^2 matrices. Then the notation $[A_{ij}]_{i,j=1,\dots,N}$ denotes a matrix composed as

$$[A_{ij}]_{i,j=1,\dots,N} := \begin{bmatrix} A_{11}, & A_{12}, & \dots, & A_{1N} \\ A_{21}, & A_{22}, & \dots, & A_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{N1}, & A_{N2}, & \dots, & A_{NN} \end{bmatrix}$$

Given a matrix $A \in \mathbb{R}^{n \times n}$, let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of A. Then A is strictly Schur if $\max_i |\lambda_i| < 1$.

DOI: 10.1002/rnc

Definition 2.1
(Admissible direction) - Let $\mathcal{S} \subset \mathbb{R}^m$ be a convex set and 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 positive scalar $\bar{\lambda} > 0$ such that $(g + \lambda v) \in \mathcal{S}, \lambda \in [0, \bar{\lambda}]$.

Definition 2.2
(Pontryagin Set Difference) - For given sets $\mathcal{A}, \mathcal{E} \subset \mathbb{R}^n$, the set determined as $\mathcal{A} \sim \mathcal{E} := \{a : a + e \in \mathcal{A}, \forall e \in \mathcal{E}\}$ is the *Pontryagin Set Difference* of $\mathcal{A}$ with respect to $\mathcal{E}$.

Definition 2.3
(Decision Set of agent i)[6] - The Decision Set $\mathcal{V}_i^{\mathcal{S}}(g)$ of the i-th agent at a point $g \in \mathcal{S}$ is the set of all admissible directions along which it can move, under the assumption that all other agents are maintaining unchanged their commands, in updating its action viz. $\mathcal{V}_i^{\mathcal{S}}(g) := \{v \in \mathbb{R}^{m_i} : [0_1^T, \ldots, 0_{i-1}^T, v^T, 0_{i+1}^T, \ldots, 0_N^T]^T \textit{ is an admissible direction for } g \in \mathcal{S}\}$.

Definition 2.4
(Viability property)[6] - A point $g \in \mathcal{S}$ is "*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 $\sum_{i=1}^N m_i = m$, there exists at least one agent i such that the sub-vector $v_i \neq 0$ belongs to its Decision Set, i.e. $v_i \in \mathcal{V}_i^{\mathcal{S}}(g)$.

Definition 2.5
Pareto Optimal solution (PO): Consider the following multi-objective problem

$$\begin{array}{l} \min_g [f_1(g_1), f_2(g_2)..., f_N(g_N)] \\ \text{subject to } g = [g_1^T, .., g_i^T, ..., g_N^T]^T \in \mathcal{S} \end{array} \tag{1}$$

The vector $g^{*_p} \in \mathcal{S}$ is a PO solution if there exist no other $g \in \mathcal{S}$ such that: $f_i(g_i) \leq f_i(g_i^{*_p}), \quad i = 1, ..., N$ for which there exists a j such that $f_j(g_j) < f_j(g_j^{*_p})$.

Definition 2.6
Graph: A graph is an ordered pair $\Gamma(\mathcal{A}, \mathcal{E})$, such that

- $\mathcal{A}$ is the set of nodes;
- $\mathcal{E}$ is a subset of pairs of **V** known as the set of edges connecting two nodes, i.e. $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$.

Definition 2.7
Degree of a node Given a graph $\Gamma(\mathcal{A}, \mathcal{E})$ $\Delta_i : \mathcal{A} \to \mathbb{Z}_+$ is the number of edges incident to a node.

Definition 2.8
(**Neighborhood of the i-th node**) The neighborhood $\mathcal{N}_i$ of the i-th node in $\Gamma(\mathcal{A}, \mathcal{E})$ consists of all its adjacent nodes i.e.

$$\mathcal{N}_i = \{i\} \cup \{j \in \mathcal{A} : (i, j) \in \mathcal{E}\}. \tag{2}$$

2.1. Basic Centralized CG

Let us consider the following pre-compensated plant model depicted in Fig. 3:

$$\left\{ \begin{array}{rcl} x(t+1) & = & \Phi x(t) + G g(t) \\ y(t) & = & H^y x(t) \\ c(t) & = & H^c x(t) + L g(t) \end{array} \right. \tag{3}$$

where: Φ is a strictly Schur matrix and G, H^y, H^c, L have proper dimensions, $x(t) \in \mathbb{R}^n$ is an extended state including both plant and primal controller states, $g(t)$ is the CG action, i.e. a suitably modified version of the reference signal $r(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^m$ the plant output which is required to track $r(t)$ and $c(t) \in \mathbb{R}^{n_c}$ the constrained output vector

$$c(t) \in \mathcal{C}, \quad \forall t \in \mathbb{Z}_+ \tag{4}$$

with $\mathcal{C}$ a specified convex and compact set. Moreover it is assumed that:

DOI: 10.1002/rnc

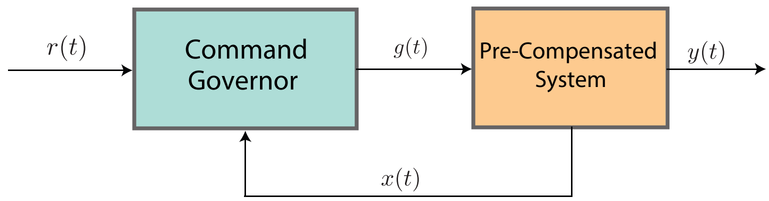

Figure 3. Command governor (CG) centralized architecture

(**A1**) Φ is a Schur matrix.

The main idea of the classical centralized solution to the CG design problem (see [2]) is to choose at each time instant a set-point $g(t)$ approximating $r(t)$ such that:

1) for $g(t) \equiv g, \forall t$, the associated steady-state constrained output

$$c_g = H^c(I - \Phi)^{-1}Gg \tag{5}$$

satisfies the constraints with a margin $\delta > 0$, i.e. $c_g \in \mathcal{C} \sim \mathcal{B}_\delta$.

2) if the command $g(t) \equiv g$ were kept constant from t onward, constraints would never be violated by the predictions of c, i.e. $c(k, x(t), g) \in \mathcal{C}, \forall k \geq 0$, where

$$c(k, x, g) := H^c \left(\Phi^k x + \sum_{\tau=0}^{k-1} \Phi^{k-\tau-1} Gg \right) + Lg \tag{6}$$

As proven in [10] and [2], the latter two conditions translate into confining all admissible g in a convex set defined as follows

$$\mathcal{V}(x) := \{g \in \mathcal{W}_\delta : c(k, x, g) \in \mathcal{C}, \ \forall k \in \{0, 1, \ldots, k_0\}\} \tag{7}$$

where k_0 is an integer that can be computed off-line and

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

Then the Centralized CG design problem can be summarized as the one of finding the command $g(t)$ by solving at each time instant t the following optimization problem

$$g(t) := \arg \min_{g \in \mathcal{V}(x(t))} \|g - r(t)\|^2_\Psi \tag{9}$$

with matrix $\Psi = \Psi^T > 0$ being a striclty positive defined matrix. For details please refer to [2]. It has been proven that this scheme always ensures constraints satisfaction. Moreover, under constant references r, the applied command $g(t)$ converges in finite time to a constant reference $\hat{r}$, which is the best admissible approximation of r.

3. SYSTEM DESCRIPTION AND PROBLEM FORMULATION

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

$$\begin{cases} x_i(t+1) &= \Phi_{ii} x_i(t) + G_i g_i(t) + \sum_{j \in \mathcal{A} \setminus \{i\}} \Phi_{ij} x_j(t) \\ y_i(t) &= H_i^y x_i(t) \\ c_i(t) &= H_i^c x(t) + L_i g(t) \end{cases} \tag{10}$$

DOI: 10.1002/rnc

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 and $y_i \in \mathbb{R}^{m_i}$ is the output vector which is required to track the local nominal reference $r_i \in \mathbb{R}^{m_i}$. Moreover, $c_i \in \mathbb{R}^{n_i^c}$ represents the local constrained vector depending on the aggregate state vectors $x = [x_1^T, \ldots, x_N^T]^T \in \mathbb{R}^n$, $n = \sum_{i=1}^N n_i$ and $g = [g_1^T, \ldots, g_N^T]^T \in \mathbb{R}^m$, $m = \sum_{i=1}^N m_i$.

Let the aggregate constrained vector $c = [c_1^T, \ldots, c_N^T]^T \in \mathbb{R}^{n^c}$, with $n^c = \sum_{i=1}^N n_i^c$ be defined as well. It is assumed that at each time instant $c(t)$ must be constrained in (4) where $\mathcal{C} \subset \mathbb{R}^{n^c}$ is hereafter considered a convex and compact polytopic set. Note that $\mathcal{C}$ describes both local and globally coupling constraints. Then, the distributed CG design problem can be stated as follows

Problem *1*

Given systems (10) *and constraints* (4)*, locally determine, at each time step t and for each agent $i \in \mathcal{A}$, a suitable reference signal $g_i(t)$ that is the best approximation of $r_i(t)$ and such that its application do not produce constraints violation, i.e. $c(t) \in \mathcal{C}$, $\forall t \in \mathbb{Z}_+$.*

4. TURN-BASED DISTRIBUTED CG

In this section the above stated problem will be solved by introducing a novel distributed CG strategy based on a particular coordination of agents that, on the basis of the constraints structure, will be grouped into turns. Agents belonging to each turn are instructed to modify their reference according to a precise scheduling procedure established offline.

4.1. Constraints associated to the i-th Agent

The first step towards the development of a distributed CG is to note that, under local dynamics (10), the aggregated system can be written as in (3) where $x(t) = [x_1^T(t), ..., x_N^T(t)]^T, g(t) = [g_1^T(t), ..., g_N^T(t)]^T$, $y(t) = [y_1^T(t), \ ..., y_N^T(t)]^T$, and $\Phi = [\Phi_{ij}]_{i,j=1,...,N}$, $G = diag(G_1, ..., G_N)$, $H^y = diag(H_1^y, ..., H_N^y)$, $H^c = [(H_1^c)^T, ..., (H_N^c)^T]^T$ and $L = [(L_1)^T, ..., (L_N)^T]^T$.

Φ is assumed to satisfy property (**A1**). Such an assumption in this distributed context implies, according to Figure 2, that the local regulators of each subsystem (10) are capable to asymptotically stabilize the overall system (3).

In the case $\mathcal{C}$ is a polytope, $\mathcal{V}(x)$ is a polyhedron as well and can be characterized by a set of linear inequalities in $\mathbb{R}^m$ [14]

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

A very important observation for the goals of this paper is that, once computed $\mathcal{V}(x)$ as in (11), the i-th agent may concur only to the violation of the constraints that directly depend on the command $g_i(t)$. Then each inequality of (11) involves just a limited number of agents. To put in evidence the constraints involving the i-th agent only, for each $i \in \mathcal{A}$, it is possible to define the block-matrices $A_i = [A_{i,1}|......|A_{i,N}]$, with $A_{i,j} \in \mathbb{R}^{z_i \times m_i}$, $Q_i = [Q_{i,1}|......|Q_{i,N}]$, with $Q_{i,j} \in \mathbb{R}^{z_i \times n_i}$ and vector $\tilde{b}_i \in \mathbb{R}^{z_i}$ collecting respectively the all and only (say z_i) rows a_j of A, rows q_j of Q and b_j of b in which the i-th agent is involved, i.e. such that the sub-vectors $a_j^i \neq 0_{m_i}$. As a consequence, the inequalities in $\mathcal{V}(x(t))$ associated to the i-th agent can be described as

$$A_i g(t) + Q_i x(t) \leq \tilde{b}_i \tag{12}$$

At this point, it is important to remark that, in some cases (e.g. for systems with no or weak dynamic interaction and coupling constraints), the matrix A can be quite sparse and only a subset of agents are involved in the constraints associated to the i-th agent.

It is worth emphasizing that the above constraints structure can be associated to an undirect graph $\Gamma(\mathcal{A}, \mathcal{E})$, whose set of nodes coincides with the set of agents $\mathcal{A}$ and where $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$ is the set of edges connecting agents (nodes) whose subsystem evolutions are jointly constrained with the i-th subsystem evolution as defined in (12), i.e. $j \in \mathcal{N}_i$ in $\Gamma(\mathcal{A}, \mathcal{E})$ if $A_{i,j} \neq 0$, where $A_{i,j} \in \mathbb{R}^{z_i \times m_i}$ is the j-th sub-matrix of A_i.

DOI: 10.1002/rnc

In this formulation, from the perspective of the i-th agent, constraints (4) become

$$\tilde{A}_i \tilde{g}_i(t) + \tilde{Q}_i \tilde{x}_i(t) \leq \tilde{b}_i \tag{13}$$

where $\tilde{g}_i(t) = S_{g,\mathcal{N}_i} g(t)$ and $\tilde{x}_i(t) = S_{g,\mathcal{N}_i} x(t)$ denote respectively the commands and the states associated to the neighbors of the i-th agent, with $\tilde{A}_i = A_i S^T_{g,\mathcal{N}_i}$ and $\tilde{Q}_i = Q_i S^T_{x,\mathcal{N}_i}$ the associated matrices. $S_{g,\mathcal{N}_i}$ and $S_{x,\mathcal{N}_i}$ are selection matrices that respectively extracts all rows of g and x related to the agents in $\mathcal{N}_i$.

Next, assume that at time t the i-th agent receives from its neighbors the values of their states and of their previously applied commands, i.e. $\tilde{x}_i(t)$ and $\tilde{g}_i(t-1)$. If at time t all agents in $\mathcal{N}_i$ except the i-th were holding the commands applied at time $t-1$, then the i-th agent could select a local command g_i satisfying constraints (12) by fulfilling the following inequalities

$$\tilde{A}_i \tilde{g}_i \leq \tilde{b}_{i,\delta} - \tilde{Q}_i \tilde{x}_i(t) \tag{14}$$

where $\tilde{g}_i$ is set equal to $\tilde{g}_i(t-1)$ for all entries except g_i. This idea can be extended to a larger number of agents using the notion of Turn:

Definition 4.1
(**Turn**) A turn $\mathcal{T} \subset \mathcal{A}$ is a *subset of non-neighboring nodes*, i.e. $\forall i, j \in \mathcal{T}$ such that $i \neq j$, $j \notin \mathcal{N}_i$ (none of them is a neighbor of the others).

The following proposition can be proved

Proposition *1*
Let $\mathcal{T}_t \subset \mathcal{A}$ denote a turn selected at time t. Then, under the assumption that $Ag(0) + Qx(0) \leq b$, if at each time t all agents not in $\mathcal{T}_t$ keep applying their previously applied commands, i.e. $g_i(t) = g_i(t-1), \forall i \notin \mathcal{T}_t$ and the agents in $i \in \mathcal{T}_t$ update their commands g_i accordingly to (14), then the overall constraints (4) are never violated.

Proof: It directly follows from the structures of set $\mathcal{V}(x)$ and constraints (14). □

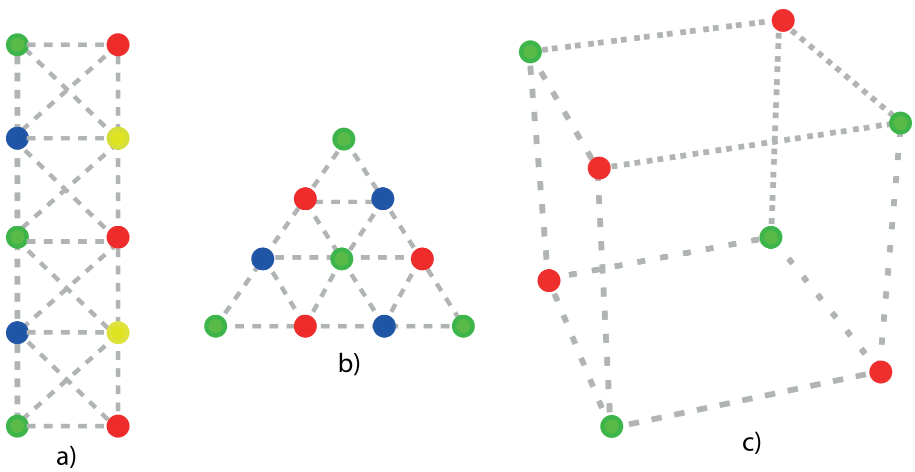

Figure 4. Particular lattice structures: a) four colors are needed to cover the graph, b) three colors are needed, c) only two colors are required

4.2. *The Overall Algorithm*

At this point, given a sequence of *turns* $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3...$, and assuming that each agent knows to which turn it belongs to, the problem of locally determining at each time t the best $g_i(t)$ approximating $r_i(t)$ such that global constraints are satisfied can be solved by allowing only the agents in the current turn $\mathcal{T}_t$ to update their commands in accordance with constraints (14). This idea can be formalized as follows:

DOI: 10.1002/rnc

Algorithm 1: The Turn-Based CG (TB-CG)

(Agent i) repeat at each time t

if $i \in \mathcal{T}_t$ **then**
  receive $\tilde{x}_i(t), \tilde{g}_i(t-1)$ from neighbors;
  solve
$$g_i(t) = \arg\min_{g_i} ||g_i - r_i(t)||^2_{\Psi_i} \quad \text{subject to } (14) \tag{15}$$
else
  set $g_i(t) = g_i(t-1)$
  apply $g_i(t)$
  transmit $g_i(t)$ and $x_i(t)$ to the neighboring agents
end if

where $\Psi_i = \Psi_i^T > 0, \forall i \in \mathcal{A}$. Please note that the proposed algorithm satisfies the constraints using only local data. In fact, each agent in the turn $\mathcal{T}_t$ only needs to know the constraints in which it is involved and the commands, and the states of its neighbors. This represent a major improvement with respect to previous sequential [6], [23] and parallel [5] schemes were global knowledge of the system dynamics was assumed and the overall system state had to be broadcasted.

5. PROPERTIES

Thanks to the following Lemma 1, it is possible to prove that the scheme here proposed can be seen as a special instance of the sequential scheme (S-CG) proposed in [23].

Lemma *1*
Given a turn $\mathcal{T}_t$, the solutions of all local optimization problems
$$\begin{array}{l} \min_{g_i} ||g_i - r_i(t)||^2_{\Psi_i} \\ subject\ to\ (14), \forall i \in \mathcal{T}_t \end{array} \tag{16}$$
is equivalent to the solution of the following single optimization problem
$$\begin{array}{l} \min_{g_i, i \in T_t} \sum_{i \in \mathcal{T}_t} ||g_i - r_i(t)||^2_{\Psi_i} \\ subject\ to \\ \quad \tilde{A}_i \tilde{g}_i \leq \tilde{b}_{i,\delta} - \tilde{Q}_i \tilde{x}_i(t), \quad \forall i \in \mathcal{T}_t \end{array} \tag{17}$$

Proof: Each optimization problem (16) governs a single command g_i. Thanks to the definition of turns, if a constraint is influenced by $g_i, i \in \mathcal{T}_t$ then it is not influenced by any other $g_j, j \in \mathcal{T}_t$. As a consequence, a vector collecting local feasible solutions for (16) for all $i \in \mathcal{T}_t$ is also a feasible solution for problem (17). Moreover, because each agent deals with a decoupled objective function in (16), clearly
$$\sum_{i \in \mathcal{T}_t} \min_{g_i} ||g_i - r_i(t)||^2_{\Psi_i} = \min_{g_i, i \in T_t} \sum_{i \in \mathcal{T}_t} ||g_i - r_i(t)||^2_{\Psi_i}$$
which means that problem (17) is an aggregation of $|\mathcal{T}_t|$ decoupled optimization problems. □

The main consequence of Lemma 1 is that the proposed scheme will share the same properties of the sequential CG schemes introduced in [23] and [6]. The most relevant properties of the proposed Turn based-CG scheme are summarized in the following Theorem 1.

Theorem 1
Consider systems (10) along with the distributed **TB-CG** (Algorithm 1) selection rule performed by agents in $\mathcal{A}$ distributed into turns $\mathcal{T}_i$ such that periodically, all agents of the network are selected, i.e. $\exists t' : \forall t > 0, \cup_{i=0}^{t'} \mathcal{T}_{t+i} = \mathcal{A}$. Assume that at time $t = 0$ an admissible solution $g(0)$ exists for problem (4). Then:

DOI: 10.1002/rnc

1) the overall system acted by the agents implementing the TB-CG policy never violates the constraints (4), $\forall t \in \mathbb{Z}_+$;

2) under the assumption that the set $\mathcal{W}_\delta$, defined in (8), is viable according to Definition 2.4, 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 (PO) stationary (constant) solution of the following multi-objective optimization problem

$$\begin{array}{l} \min_g [\| g_1 - r_1 \|^2_{\Psi_1}, \ldots, \| g_i - r_i \|^2_{\Psi_2}, \ldots, \| g_N - r_N \|^2_{\Psi_N}] \\ \text{subject to } g = [g_1^T, .., g_i^T, ..., g_N^T]^T \in \mathcal{W}_\delta \end{array} \tag{18}$$

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

Proof: Item 1) follows directly from Proposition 1. For what it concerns Item 2), because of Lemma 1, the TB-CG scheme is equivalent to the S-CG carried out by agents managing all the commands updated within $\mathcal{T}_i$ which, under the viability property 2.4, converges to a PO solution for the problem [6]

$$\begin{array}{l} \min_g \left[\sum\limits_{i \in \mathcal{T}_1} ||g_i - r_i(t)||^2_{\Psi_i}, \ldots, \sum\limits_{i \in \mathcal{T}_q} ||g_i - r_i(t)||^2_{\Psi_i} \right] \\ \text{subject to } g = [g_1^T, .., g_i^T, ..., g_N^T]^T \in \mathcal{W}_\delta \end{array} \tag{19}$$

The proof is concluded by noticing that a PO solution for problem (19) is also PO for (18). □

6. TURNS AND GRAPH COLORABILITY THEORY

A crucial point of the proposed algorithm is the way the turn sets $\mathcal{T}_t$ are determined. As seen, the first general requirement to guarantee convergence is that, periodically, all agents of the network are selected, i.e. $\exists t' : \forall t > 0, \cup_{i=0}^{t'} \mathcal{T}_{t+i} = \mathcal{A}$.

In order to make the overall system behaving "fast" in response to fast changing reference signals, a further guideline is to choose a sequence of $\mathcal{T}_t$ that maximizes the frequency with which the agents are allowed to update their local commands.

From a practical implementation viewpoint the most reasonable choice is to resort to a periodic scheduling $\mathcal{T}_1, ..., \mathcal{T}_q$ of length q, with q as small as possible and such that $\cup_{i=1,..,q} \mathcal{T}_i = \mathcal{A}$. In this way, one can ensure that each agent can update its command with a frequency that is at least $1/q$. Furthermore, among the possible periodic definitions of turns that minimize q, it is convenient to chose the ones that maximize the sum of the frequencies with which each sensor is selected in the period. To formalize this idea, define the graph $\Gamma(\mathcal{A}, \mathcal{E})$, whose set of nodes coincides with $\mathcal{A}$ and where $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$ is the set of edges connecting neighbor agents, i.e. the edge (i, j) belongs to $\mathcal{E}$ if and only if $j \in \mathcal{N}_i$. The problem of determining a convenient set of turns can be defined as the following lexicographic problem

$$\left\{ \begin{array}{l} q^* = \min\limits_{q \in \{1,...,N\}, \mathcal{T}_1 \subseteq \mathcal{A}, ..., \mathcal{T}_q \subseteq \mathcal{A}} q \\ subject\ to \\ \bigcup_{t=1}^{q} \mathcal{T}_t = \mathcal{A} \\ (i,j) \notin \mathcal{E}, \ \ \forall i,j \in \mathcal{T}_t, t = 1, ..., q. \end{array} \right. \tag{20}$$

$$\left\{ \begin{array}{l} \max\limits_{\mathcal{T}_1 \subseteq \mathcal{A}, ..., \mathcal{T}_{q^*} \subseteq \mathcal{A}} \sum\limits_{t=1}^{q^*} |\mathcal{T}_t| \\ subject\ to \\ \bigcup_{t=1}^{q*} \mathcal{T}_t = \mathcal{A} \\ (i,j) \notin \mathcal{E}, \ \ \forall i,j \in \mathcal{T}_t, t = 1, ..., q^*. \end{array} \right. \tag{21}$$

The first step of the lexicographic problem (20), it is equivalent to the well known minimal vertex coloring problem [12], i.e. to determine an assignment color to each graph vertex such that no two adjacent vertices have the same color and the number of colors is minimized. Accordingly, all the properties of the minimal vertex coloring apply. The problem of minimal vertex coloring is a NP-Complete optimization problem [13] that has been widely studied in the last century [12]. For the purposes of this paper, the most relevant property of this problem is that the minimal number of colors of a graph is bounded (and often by a small integer) regardless of the number of nodes for many classes of graph topologies,. It is the case, for instance, for lattices where (e.g. see Figure 4) simple optimal coloring rules can be easily found. Moreover, the Brooks' Theorem [3] ensures that, for any graph with maximum degree $\Delta(\Gamma) := \max_{v \in \mathcal{A}} \Delta_i(v)$, the chromatic number (minimal number of colors) is at most $q = \Delta(\Gamma) + 1$. Simple polynomial-time distributed algorithms exist that are able to color a Δ-graph with at most $\Delta(\Gamma) + 1$ colors [11].

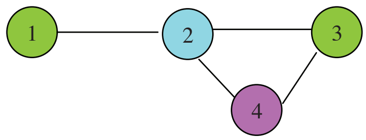

Figure 5. A colored graph

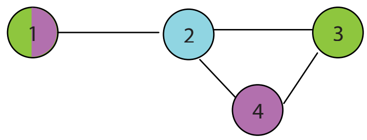

Figure 6. A multicolored graph

The second step of the lexicographic problem (21) is used to exploit the fact that some agent can be selected more often than q, or in terms of graph colorability, that is possible to assign more than one color to some nodes (see Figures 5 and 6). Although the proposed generalization is, at the best of our knowledge, currently not addressed in the Graph Theory literature, some research activities are moving in similar directions, see e.g. [18] where the problem of assigning a predefined number of colors to each vertex using the fewest possible number of colors is considered.

As far as the complexity arguments are concerned, it is quite trivial to show that the entire problem (20)-(21) is NP-hard. In fact Problem (20) is NP-complete. Moreover, starting from a solution of problem (20) a candidate solution for problem (21) can be checked in a polynomial time. The NP-completeness seems not to be so trivial to prove and it is still under investigation.

From a computational point of view, a possible exact solution consists in first using existing state-of-the-art algorithms [12] to solve (20), and then solving (21) via a backtracking procedure. Clearly this approach is viable only at the design phase and for moderately small graphs. To deal with the other situations, when for instance online reconfiguration might be needed, a heuristic solution based on a greedy refining an (exact or approximated) minimal vertex coloring is proposed here

Algorithm 2: Approximated Round-robin Policy

Solve (exact or approximated) (20) and get $\mathcal{T}_1, ..., \mathcal{T}_q$
for $i = 1, ..., N$ **do**
    for $t = 1, ..., q$ **do**
        $\mathcal{T}_t \leftarrow \mathcal{T}_t \cup \{i\}$
        if $\mathcal{T}_1, ..., \mathcal{T}_q$ is NOT a feasible round-robin **then**
            $\mathcal{T}_t \leftarrow \mathcal{T}_t \setminus i$
        end if
    end for
end for

Note that, even in the general case, using the distributed approximated coloring algorithm detailed in [11], the proposed heuristic can be also made distributed, and can be thus used for the online reconfiguration (e.g. following plug-in or plug-out of subsystems) of the round-robin policy. Please note that although approximated, this heuristic would guarantee that the frequency with which each agent is chosen is at least $1/(\Delta(\Gamma)+1)$, independently on the number of agents.

7. ILLUSTRATIVE EXAMPLES

In order to show the effectiveness of the proposed method, a set of 21 decoupled particle masses representing vehicles has been considered, as depicted in Figure 7.

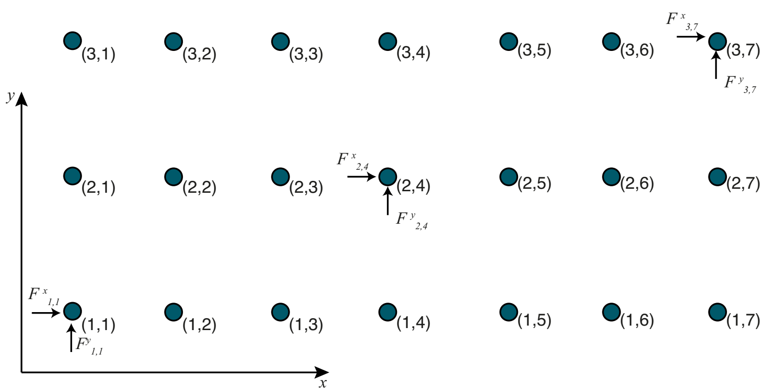

Figure 7. Planar particle masses.

The following equations describe the (i, j)-th mass dynamics

$$\begin{array}{rcl} m\ddot{x}_{i,j} & = & F^x_{i,j} \\ m\ddot{y}_{i,j} & = & F^y_{i,j} \end{array} \tag{22}$$

where $(x_{i,j}, y_{i,j})$, $i \in \{1, 2, ..7\}$, $j \in \{1, 2, 3\}$ are the coordinates of the (i, j)-th mass position w.r.t a fixed Cartesian reference frame and $(F^x_{i,j}, F^y_{i,j})$, $i \in \mathcal{A}$, the components along the same reference frame of the forces acting as inputs for the subsystems. The value $m = 1\ [Kg]$ will be assumed in the simulations. For CG design purposes the models have been discretized with a sampling time of $T_c = 0.1\ [s]$ and an optimal LQ state-feedback local controller is used as a precompensator for each mass.

Each mass is subject to the local constraints

$$\left|F^p_{i,j}(t)\right| \leq 2\ [N]\ \ p = x, y \tag{23}$$

representing input-saturation constraints on the forces $F^x_{i,j}$ and $F^y_{i,j}$, $i \in \{1, 2, ..., 7\}$, $j \in \{1, 2, 3\}$, acting as inputs to the vehicles.

Moreover each couple of masses is subject to proximity constraints involving either x position

$$0.125[m] \leq |x_{i,j+1}(t) - x_{i,j}(t)| \leq 0.375[m], \forall t \in \mathbb{Z}_+ \tag{24}$$

or y position

$$0.125[m] \leq |y_{i+1,j}(t) - y_{i,j}(t)| \leq 0.375[m], \forall t \in \mathbb{Z}_+ \tag{25}$$

or both x and y positions

$$\left\{ \begin{array}{l} 0.125[m] \leq |x_{i+1,j+1}(t) - x_{i,j}(t)| \leq 0.375[m] \\ 0.125[m] \leq |y_{i+1,j+1}(t) - y_{i,j}(t)| \leq 0.375[m], \\ \forall t \in \mathbb{Z}_+ \end{array} \right. \tag{26}$$

DOI: 10.1002/rnc

Along the presented simulations, each vehicle has been instructed to track a "circular" reference

$$\begin{aligned} r^x_{i,j}(t) &= \rho\cos\left(q_{i,j}\tfrac{2\pi}{75}t\right)+i\bar{r}, \quad t\in\mathbb{Z}_+ \\ r^y_{i,j}(t) &= \rho\sin\left(q_{i,j}\tfrac{2\pi}{75}t\right)+j\bar{r}, \quad t\in\mathbb{Z}_+ \end{aligned} \tag{27}$$

with $i\in\{1,2,...,7\}$, $j\in\{1,2,3\}$, $\rho=0.125[m]$ is the radius and $q_{i,j}$ denotes the spin direction evaluated in the following way

$$q_{i,j}=\begin{cases} 1, & \text{if } i+j \text{ is an even number} \\ -1, & \text{if } i+j \text{ is an odd number} \end{cases} \tag{28}$$

and scalar $\bar{r}=-0.0884[m]$.

7.1. *Example 1: Comparison with existing CG based schemes*

In this first scenario we assume that each mass is subject to constraints of the type (24)-(26). The resulting coupling structure can be then modeled by means of the graph depicted in Figure 8, where each mass represents a vertex, while the black dashed edges denote the existence of constraints between two masses. In this case the minimal vertex coloring problem (20) is solved by using only three colors (blu, red, green). As a consequence, the whole CG supervision action is spread among three groups of agents which adopt the Turn-Based policy introduced in Section 3.

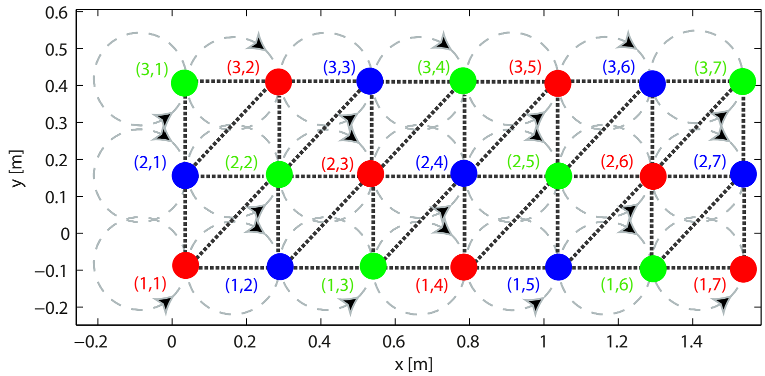

Figure 8. Example 1: Planar masses (colored spots) with desired references (black dashed lines)

Simulation results involve a comparison among three CG supervision strategies:

- the standard CG (Centralized CG) [2] where a unique CG device supervises the entire network by receiving/transmitting information to all subsystems;
- the distributed sequential agent-based CG (S-CG) [23] where only one agent (local CG) at a time is allowed to modify its command while all other agents keep applying their previously applied commands;
- the proposed Turn-Based distributed sequential CG, hereafter referred to as Color-CG, where agents perform Algorithm 1 on the basis of local and neighbors' informations only.

The simulation has been carried out over an horizon $T=150[steps]$ starting from initial mass positions $\left(x_{i,j}(0)=r^x_{i,j}(0), y_{i,j}(0)=r^y_{i,j}(0)\right)$, $(\dot{x}_{i,j}(0)=0, \dot{y}_{i,j}(0)=0)$, as depicted in Figure 8.

In Figures 9-11 the computed CG actions $g_{i,j}(t)$ for each agent have been depicted on a 2D plan, while Figure 12 depicts some constrained variables related to masses $(1,1)$ $(1,2)$ and $(2,1)$. A video showing the positions of the masses during the simulation (in that video $T=300[steps]$) for each applied supervising strategy can be downloaded at the url "http://youtu.be/msu1Lu8STus". From Figures 9-12 it is possible to observe that, within the given simulation horizon T, although all the strategies are capable to satisfy the imposed constraints, only the centralized CG and the Color-CG are able to cover the entire circle. On the contrary, masses supervised by the S-CG method are not able to track in a proper way the desired reference.

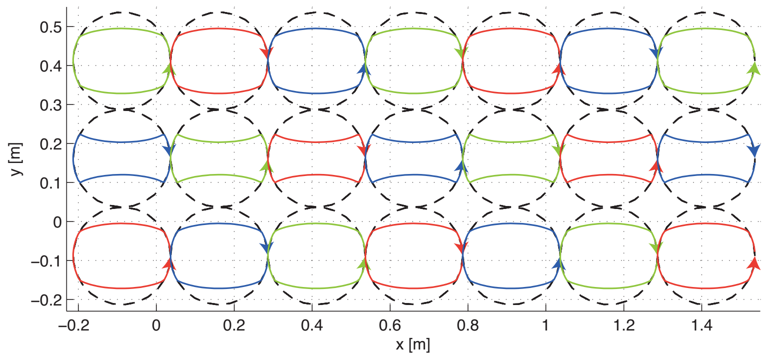

Figure 9. Example 1: Applied references under the standard centralized CG case

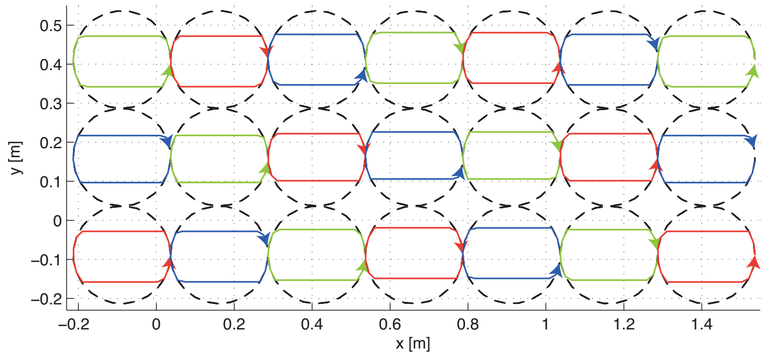

Figure 10. Example 1: Applied references under the distributed Color-CG case

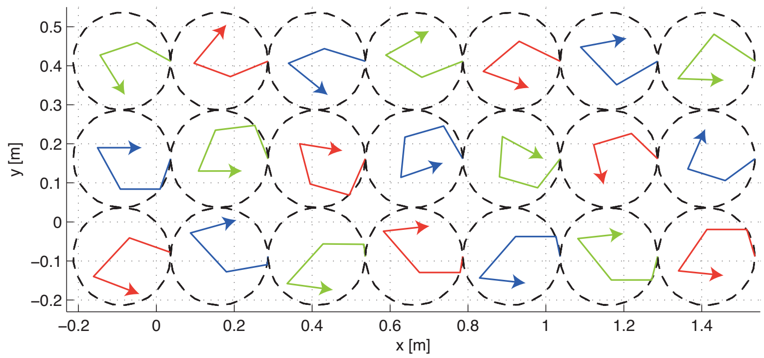

Figure 11. Example 1: Applied references under the S-CG case

A comparison under a different perspective can be accomplished by observing how the quantity $J(t) := \sum_{i=1}^{7} \sum_{j=1}^{3} (r_{i,j}(t) - g_{i,j}(t))^2$ varies during the simulations. In this respect, Figure 13 shows that Centralized CG and Color-CG exhibit a very similar behavior although the Color-CG makes use of a reduced amount of information/resources. In fact, see Figures 15-16 for the computational burdens and consider that each agent only exploit local and neighbors information to solve the underlying optimization problems. Finally, as expected, the S-CG exhibits the worst performance because the resulting command updating rate is too low for this application

7.2. *Example 2: Scalability Analysis*

Further simulations, involving the above compared strategies, have been carried out by considering scenarios characterized by an increasing number of masses having the same constraints structure as in the previous scenario. In particular, a system ranging from 4 masses up to 256 masses (agents)

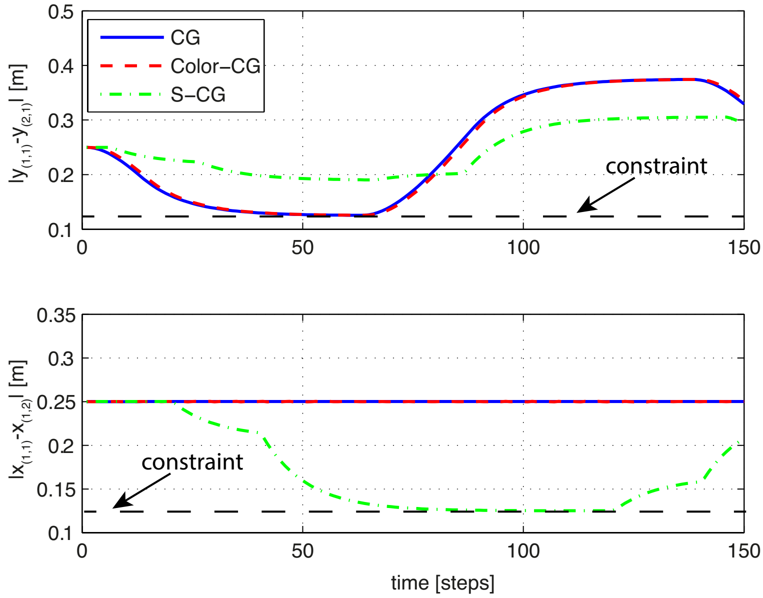

Figure 12. Example 1: Some constrained variables

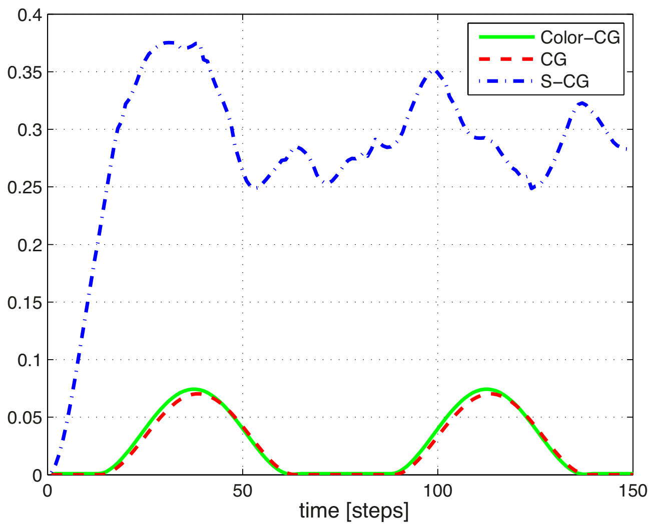

Figure 13. Example 1: Trend related to $J(t)$ during the simulation

have been taken into account. Comparisons in this case have been performed in terms of scalability of the proposed approaches. Simulation results have been reported in Figures 14-17. In this respect, in Figure 14 the average residual costs

$$\hat{J} := \frac{1}{T}\sum_{t=0}^{T} J(t) \tag{29}$$

have been reported. Also in this case, proposed Color-CG and centralized CG exhibit the best performance. Moreover, it is evident in Figures 15-17 that one of the main advantages of the proposed Turn-Based distributed schemes is in the low amount of data exchanged for its implementation and in the related negligible computational burdens that do not increase with the number of agents. Such a property could represent a clear advantage with respect to centralized implementation, whose required CPU time and the required amount of information transmitted/received increase linearly with the number of the masses.

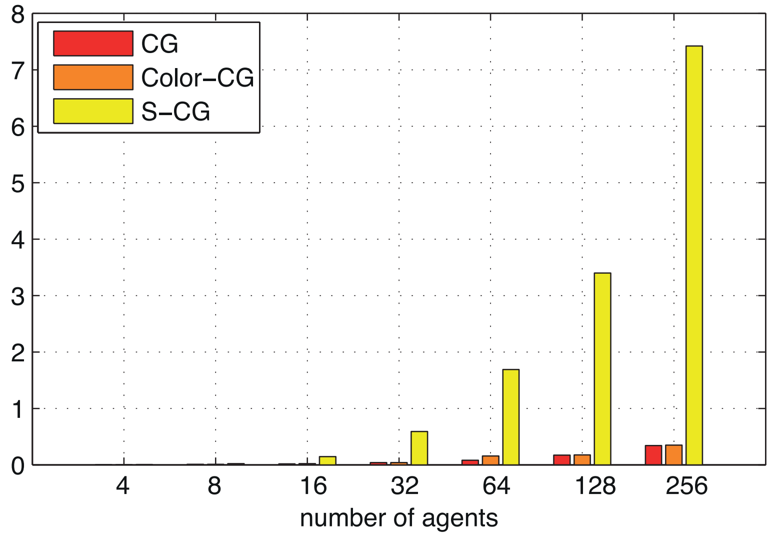

Figure 14. Example 2: Residual Cost $\hat{J}$

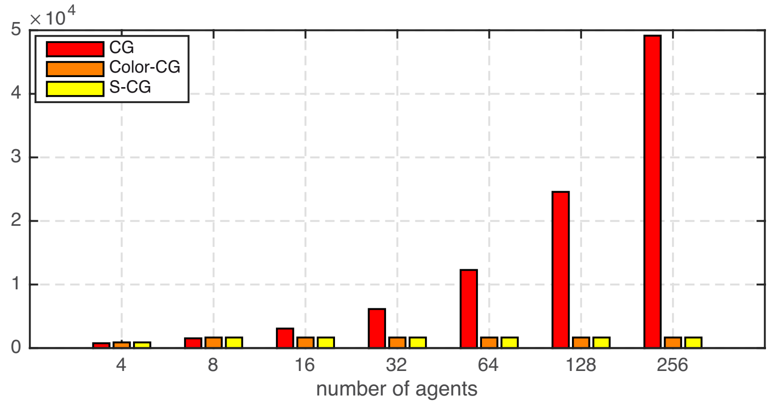

Figure 15. Example 2: Information received/transmitted by assuming that 32 bit are needed to encode each scalar

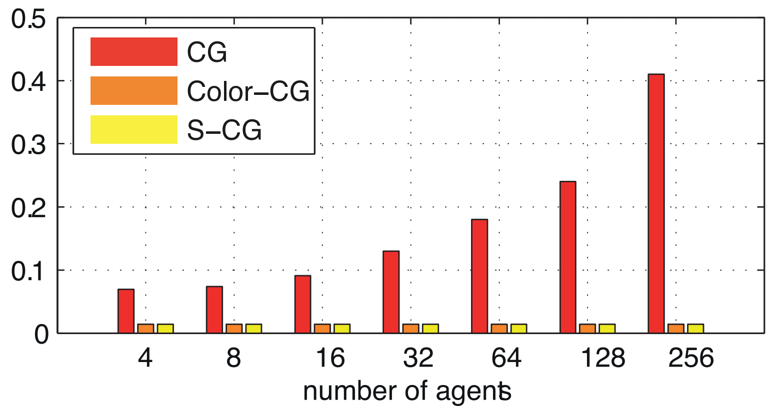

Figure 16. Example 2: Mean CPU time (seconds per step) per agent

DOI: 10.1002/rnc

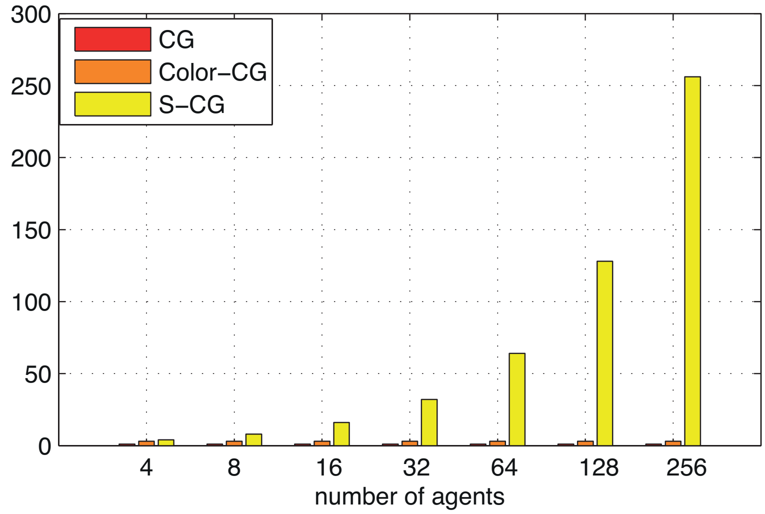

Figure 17. Example 2: Number of steps before next decisions vs number of agents

7.3. *Example 3: Coloring vs Multi-coloring*

In this section the benefits of exploiting multi-coloring approach mentioned in Section 6 to group the agents, are highlighted. To this end, the same masses and task of the previous example have been considered along with a different constraints coupling graph as depicted in Fig. 18. Fig. 16 also shows the coloring map generated by solving only the classical coloring problem formalized in Problem (20). Since this constraint structure is sparser than that taken into account in the previous example, it allows one to get (see Fig. 19) a multi-coloring map by executing Algorithm 2. The Turn-Based CG scheme is then implemented in both scenarios and compared with the centralized CG. The related results of this simulations appear in Fig. 20 and Table 1, where the acronym m-Color-CG is used to denote the Turn-Based CG applied to multi-colored graph in Fig. 19. In particular in Fig. 20 the cost $J(t)$ evaluated along the simulation is depicted. It is evident that the m-Color-CG gets better performance when compared to the Color-CG. Such an aspect is clearly related to the reduced average number of steps to the next decision reported in Table 1. In the same Table, by looking at the residual cost $\hat{J}$, it is possible to note that the m-Color-CG features about the 5% of cost reduction, which is a significant improvement if compared to the cost reduction of 8% obtained via the Centralized CG. This outcome is more relevant if we consider that is achieved by using the same amount of resources (see CPU Time and Average Info RX/TX in Table 1) used for the Color-CG scheme.

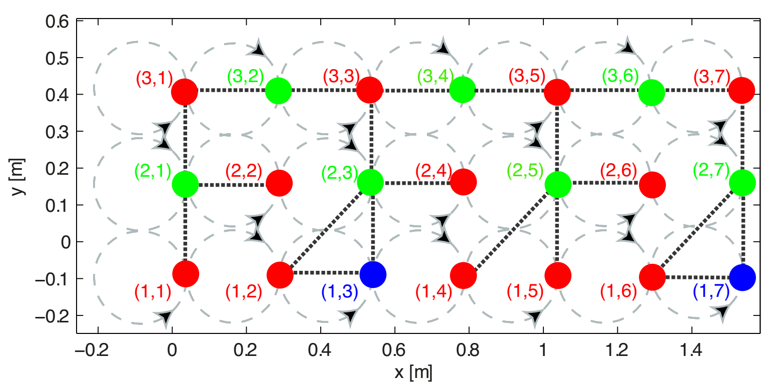

Figure 18. Example 3: Planar masses and coloring map

Int. J. Robust. Nonlinear Control (2017)
DOI: 10.1002/rnc

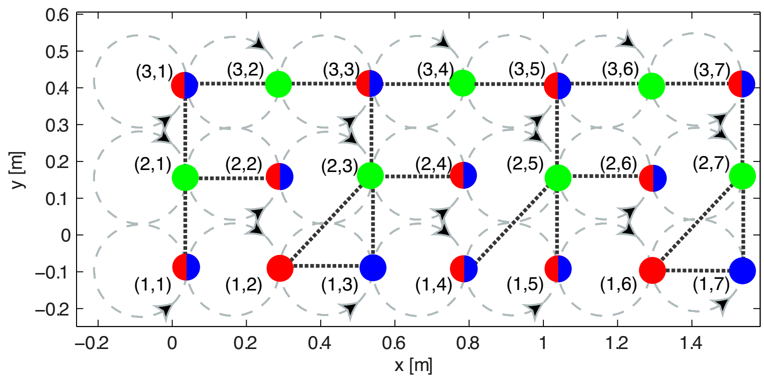

Figure 19. Planar masses and multi-coloring map

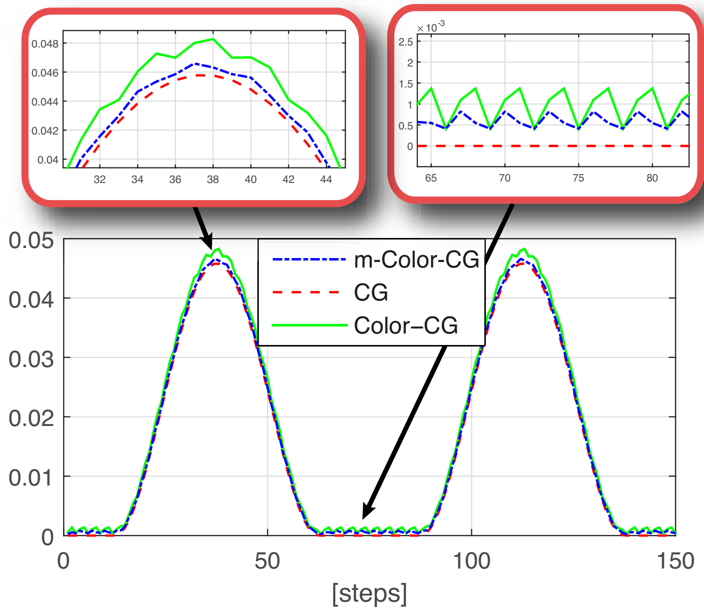

Figure 20. Example 3: Trend related to $J(t)$ during the simulation

Table I. Example 3: Simulation results

	CG	Color-CG	m-Color-CG
AVERAGE STEPS TO NEXT DECISION	1	3	2.28
RESIDUAL COST $\hat{J}$	0.0315	0.0342	0.0326
CPU TIME $[sec]$	0.0695	0.014	0.014
AVERAGE INFO RX/TX	4032	402	402

8. CONCLUSIONS

In this work a novel Turn-Based sequential distributed CG scheme has been proposed as a generalization of the early sequential distributed CG scheme of [23] with enhanced scalability and optimality properties. Unlike that earlier scheme, where, according to a prescribed periodic order, only a single agent at a time was instructed to update its command while all others were committed to keep applying their current commands, here such a polling strategy is applied turn-wise, where a turn is a group of agents that can simultaneously update their commands without consequences on the fulfillment of the constraints.

Graph minimal vertex coloring problems have been shown to be instrumental for the determination of turns and for the implementation of the Turn-Based CG strategy, whose main properties concerning optimality, stability and feasibility have been discussed. Specifically a new link between distributed control and graph multi-coloring problems has been formally presented and used as a special tool to include agents into multiple not conflicting turns in order to increase the performance.

In the final example, the performance of the proposed scheme has been compared with those pertaining to both the centralized and the agent-based distributed schemes presented in [23]. It was found that the proposed scheme achieves better scalability and performance. In fact, it has been there shown that, unlike the centralized scheme, the resources (CPU time and exchanged data) required to accomplish the coordination task remain almost constant and independent of the number of agents whereas the performance are very close to those achieved by the centralized scheme.

REFERENCES

1. T.S. Basar, J.G. Olsder, *Dynamic Noncooperative Game Theory*, SIAM, Philadelphia.
2. 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.
3. R. L. Brooks, "On colouring the nodes of a network", *Proc. Cambridge Philosophical Society, Math. Phys. Sci.*, 1941, 37, pp. 194197.
4. A. Casavola, E. Mosca and D. Angeli., "Robust command governors for constrained linear systems", *IEEE Trans. Automat. Control*, 45, pp. 2071-2077, 2000.
5. A. Casavola, F. Tedesco, E. Garone, "A parallel distributed supervision strategy for multi-agent networked systems", *Systems & Control Letters*, 97, pp. 115-124.
6. A. Casavola, F. Tedesco, E. Garone, "A Distributed Multi-Agent Command Governor Strategy for the Coordination of Networked Interconnected Systems", *IEEE Transactions on Automatic Control*, 2014, 59 (8), pp. 2099 - 2112.
7. A. Chinchuluun, P.M. Pardalos, A. Migdalas, L. Pitsoulis, *Pareto optimality, game theory and equilibria*. Springer, 2008
8. M. Farina , R. Scattolini, "Distributed predictive control: A noncooperative algorithm with neighbor-to-neighbor communication for linear systems", *Automatica*, 2012, Vol. 48(6), pp. 1088-1096.
9. E. Garone, S. Di Cairano, I. Kolmanovsky. "Reference and command governors for systems with constraints: a survey on theory and applications". *Automatica* 2017, 75, pp. 306-328
10. E.G. Gilbert, I. Kolmanovsky and K. Tin Tan. "Discrete-time Reference Governors and the Nonlinear Control of Systems with State and Control Constraints". *Int. Journ. on Robust and Nonlinear Control*, 1995, 5, pp. 487-504,
11. A.D. Grable, A. Panconesi, "Fast distributed algorithms for Brooks-Vizing colourings, "SODA '98: Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms", *Journal of Algorithms*, 1998, 37, pp. 473480.
12. T. Jensen, B. Toft, *Graph Coloring Problems*. Wiley-Interscience Ser. in Discrete Math. and Optim. John Wiley and Sons, Inc., N.Y. (1994).
13. R. M. Karp, "Reducibility Among Combinatorial Problems", *Complexity of Computer Computations*, Eds. R.E. Miller e J.W. Thatcher, New York, Plenum, 1972, pp. 85103.
14. I. Kolmanovsky and E. G. Gilbert. "Maximal output admissible sets for discrete-time systems with disturbance inputs". *American Control Conference*, Seattle, pp. 1995-1999, 1995.
15. U. Kalabić and I. Kolmanovsky. "Decentralized Constraint Enforcement Using Reference Governors". *Conference on Decision and Control*, Florence, Italy, pp. 6415-6421, 2013.
16. J. M. Maestre , D. Muñoz de la Peña, E. F.Camacho , Distributed model predictive control based on a cooperative game, Optimal Control Applications and Methods, 2011, Vol. 32(2), pp. 153-176.
17. M.S. Mahmoud, "Decentralized output-feedback stabilization for interconnected discrete systems with unknown delays", *Optimal Control Applications and Methods*, Vol 31, pp. 529-545, 2009
18. D. Marx, *Graph coloring with local and global constraints*, PhD thesis, Budapest University of Technology and Economics, 2004.
19. K. M. Miettinen (1999), *Nonlinear Multiobjective Optimization*, Kluwer Academic.
20. M.A. Müller, *Distributed and economic model predictive control: beyond setpoint stabilization*. Logos Verlag Berlin GmbH, 2014.
21. R. Scattolini (2009), "Architectures for distributed and hierarchical model predictive control - a review", *Journal of Process Control*, vol. 19(5), pp. 723–731.
22. B.T. Stewart, A.N. Venkat , J.B. Rawlings, S.J. Wright, G. Pannocchia, "Cooperative distributed model predictive control", *Systems & Control Letters*, 2012, Vol. 59(8), pp. 460-469.
23. F. Tedesco, A. Casavola, E. Garone, "Distributed Command Governor Strategies for Constrained Coordination of Multi-Agent Networked Systems", *Proc. of IEEE American Control Conference*, 2012, Montreal, Canada

24. "Scalability and performance improvement of distributed sequential command governor strategies via graph colorability theory", *Proceedings of 19th IFAC World Congress*, Cape Town, South Africa, August 24 - 29, 2014.
25. P. Trodden, A. Richards, "Distributed model predictive control of linear systems with persistent disturbances", *International Journal of Control*, 2010,Vol. 83(8), pp. 1653-1663.
26. A.N. Venkat, J.B. Rawlings , S.J. Wright, "Stability and optimality of distributed model predictive control", *Proc. of the 46th IEEE CDC*, 2005, Seville, Spain, pp. 6680-6685, 2005.
27. Y. Zhu, P.R. Pagilla, "Decentralized output feedback control of a class of large-scale interconnected systems", *IMA Journal of Mathematical Control and Information*, 2006;24(1):57-69.