# Chapter 2
# Generation of Trajectories Using Predictive Control for Tracking Consensus with Sensing and Connectivity Constraint

**Bernardo Ordoñez, Ubirajara F. Moreno, Jés Cerqueira and Luis Almeida**

**Abstract** This work presents a cooperation strategy for teams of multiple autonomous vehicles to solve the rendezvous problem. The approach is based on consensus algorithms, which are basically characterized by information exchange among the team members. The proposal is based on predictive control in order to compute decentralized control laws, considering constraints and different response requirements according to the application scenario, for example, constraints related to coverage and connectivity of the group. Our work allows considering together vehicles without and with non-holonomic restrictions while optimizing the sensing range, particularly that of fixed frontal cameras, by managing orientation in the way to the rendezvous point. We show the effectiveness of our strategy with simulation results.

**Keywords** Consensus algorithm · Cooperation strategies · Optimization · Non holonomic constraint

---

This work was supported in part by CAPES/Brazil (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and FCT/Portugal (Fundação para Ciência e Tecnologia).

---

B. Ordoñez (✉) · U. F. Moreno
Department of Automation and Systems Engineering, Federal University of Santa Catarina, BOX 88040-900, Florianópolis, Santa Catarina, Brazil
e-mail: ordonez@das.ufsc.br

U. F. Moreno
e-mail: moreno@das.ufsc.br

J. Cerqueira
Department of Electrical Engineering, Federal University of Bahia, BOX 40210-630, Salvador, Bahia, Brazil
e-mail: jes@das.ufsc.br

L. Almedia
IT—Faculty of Engineering, University of Porto, BOX 4200-465, Porto, Portugal
e-mail: lda@das.ufsc.br

A. Koubâa and A. Khelil (eds.), *Cooperative Robots and Sensor Networks*, Studies in Computational Intelligence 507, DOI: 10.1007/978-3-642-39301-3_2,

# 1 Introduction

The use of robotic vehicles to perform tasks autonomously is becoming widespread due to both technological and scientific advances, for example, the miniaturization of electromechanical systems and new sensing and control paradigms. It is natural to imagine that soon, teams of vehicles will be fully autonomous and capable of carrying out challenging tasks. The use of autonomous vehicles requires coordination through the use of cooperation strategies because there are tasks that one vehicle alone could not perform due to both its partial knowledge about the task and limited resources. A coordinated set of vehicles can share information in dynamic environments to perform challenging tasks [1, 2]. Examples can be found in applications that range from military systems to mobile surveillance sensor networks for monitoring roads and air transport systems [3].

The concept of cooperative control implies by definition that in a task performed by a team of vehicles these can communicate and collaborate [4]. Among the main techniques used to solve tasks in a cooperative way, this paper focuses on consensus algorithms, which are characterized by communication and information exchange within the team [5]. Another important feature of this technique is that the design of consensus algorithms is based on decentralized implementation.

The main studies about consensus were focused on the algorithm analysis while requirements such as control effort and tracking error were not considered. Therefore, this work presents a technique for synthesis of decentralized control laws to generate consensus trajectories that maximize the performance with respect to response requirements. The strategy was named by ACvO (Algorithm Consensus via Optimization). The contributions of our strategy are twofold, on one hand, non-holonomic constraints of vehicles motion are considered when defining the tracking trajectories, on the other hand, we optimize the sensing range, particularly of fixed front cameras, according to the rendezvous point, in the sense that we control the orientation at which the vehicles arrive at the rendezvous.

A preliminary version of this work was presented in [6], in which the main result was the development of feasible consensus trajectories for all vehicles, including non-holonomic constraints and sensing range optimization. In the current work we add a discussion on the convergence of consensus trajectories and the computational time for each algorithm iteration. Moreover, we show the advantage of using the method to add constraints in a straightforward way, and we illustrate it adding an important group connectivity constraint to achieve global consensus.

The paper is organized as follows. Section 1 presented the problem definition and the main related works about cooperative control and consensus algorithm. The aspects and definitions about the consensus algorithms are shown in Sect. 2. Section 3 presents the proposed formulation via optimization to perform consensus trajectories for the rendezvous problem. Section 4 presents numerical simulations in order to analyze the proposed strategy. Also it is introduced a discussion about the convergence and computational time of the ACvO protocol. Finally, Sect. 5 concludes the work with a discussion of results and future work.

## 1.1 Related Works

The consensus problem has a long history related to the field of computer science. It was one of the bases for the development of distributed computing. A historical perspective about consensus algorithms can be founded in [7] and [8], where the study of consensus problem was formalized. Although there is this historical relationship, the focus of this work is on the application of consensus theory to cooperative control for multi-vehicles group coordination.

Theoretical aspects about the definition and resolution of the consensus problem were introduced in [9] and [10]. In these works, the consensus problem in vehicles dynamic networks is analyzed for three cases: directed networks with fixed topology, directed networks with switching topology, and network topology and communication with fixed delay. The characteristic that distinguishes those works is the consensus approach based on networks with directed information flow. The work in [11] addresses the problem of multi-vehicles cooperation with the application of algorithms based on graph theory that relates the network communication topology to the vehicles formation stability.

The consensus problem with multi-vehicles group in the presence of limited and uncertain information flow due to time-varying topologies is considered in [12]. In that work the important concept of directed spanning tree is used to evaluate the consensus algorithm convergence and graph connectivity.

A typical application of the consensus algorithms in the cooperative control context is the *rendezvous problem*, which is characterized by a group of vehicles that negotiate among them to determine a meeting location or time. Thus, the rendezvous problem for multi-vehicle groups can be defined as the design of local control strategies without active communication between the vehicles to determine the meeting point.

The rendezvous problem is also present in air surveillance tasks, where the use of UAVs has grown considerably. For example, the work in [13] shows an approach to the cooperation problem that uses a consensus algorithm to replace a centralized strategy with a decentralized algorithm. The validation was carried out using numerical simulations.

All the works related to cooperative control algorithms and consensus mentioned above addressed the possibility of adjustments in the algorithms and strategies to improve performance of a particular requirement, such as the convergence rate to reach consensus, minimization of the tracking error, or even the vehicles control effort.

The field of optimal control algorithms has been extensively studied, especially in this last decade. The development of optimal coordination strategies based on consensus algorithms is seriously compromised by the presence of corrupted data and uncertainties in measurements. Therefore, some techniques were developed to ensure cooperation among vehicles with limited access to information [9]. In [14] the convergence rate for a classes of consensus is studied using semi-defined programming. In [15], necessary and sufficient conditions of convexity related to system topology are developed.

In [16] the strategy of receding horizon control is used to formation stabilization with quadratic cost and no coupling constraints. The consensus problem in a group of vehicles with time-varying topologies is studied in [17], in which the vehicles only need knowledge about the neighboring states to reach consensus. A similar approach is found in [18], where the objective is to develop controllers for a group of vehicles based on means of consensus. An optimal control semi-decentralized strategy is developed based on the minimization of individual cost functions with finite horizon and local information.

A formation control law based on artificial potential fields and consensus algorithms for a group of unicycles is proposed in [19], which considers connected and balanced graphs to prove stability of the controller by applying the LaSalle-Krasovskii invariance principle. The work in [20] addresses the design of optimal control to ensure consensus in a multi-vehicle network adopting a global cost function to ensuring consensus with optimal control effort. It is shown that the solution of Riccati equation does not guarantee consensus under certain conditions, and therefore, it proposes a formulation based on LMI to resolve the optimization problem with constraints.

Finally, [21] addresses the parameters weight in a problem of convex optimization for random topologies, in which the convergence rate square error is used as optimization criterion. The work in [22] studies optimal consensus algorithms from the perspective of LQR theory, in which the Laplacian matrices have direct influence on the choice of optimal parameters of the system, while [23] shows a distributed cooperation algorithm for problems with coupling hard state constraints (non-convex and external disturbances).

## 2 Consensus Algorithms

In the context of cooperative control, *consensus* can be defined by a commitment among the group members to a common goal (group decision value). A variable defined as *information state* is used to model the collective view of the common objective and it can be used to represent some abstractions of the coordination variable, such as rendezvous location.

Let a directed graph of order $n$ be represented by $G_n = (\nu_n, \varepsilon_n)$ with the set of nodes $\nu = \{v_1, ..., v_n\}$, set of edges $\varepsilon_n \subseteq \nu_n \times \nu_n$ and $n$ being the number of vehicles. The nodes belong to a finite index set $\Gamma = 1, ..., n$. An edge of $G_n$ is denoted by $e_{ij} = (v_i, v_j)$. The adjacency elements $(a_{ij})$ associated with the edges of the graph are positive, i.e., $e_{ij} \in \varepsilon \Leftrightarrow a_{ij} > 0$. It is assumed that $a_{ii} = 0$ for all $i \in \Gamma$. Finally, the set of neighbors of node $\nu_i$ is denoted by $N_i = v_j \in \nu : (v_i, v_j) \in \varepsilon$.

**Definition 1** *(Directed Tree)* A directed tree is a digraph $G_n$, where exists a vehicle named root, such that all of other vehicles of the digraph can be reached through one path only starting at the root. Consequently, $\tau_G = \{\nu_\tau, \varepsilon_\tau\}$ is a spanning tree of $G_n$, if both $\tau_G$ is a directed tree and $\nu_\tau = \nu$.

Let $\xi_i \in \Re^n$ denote the decision group value of node $v_i$, then, $G_\xi = (G_n, \xi)$ with $\xi = (\xi_1, ..., \xi_n)^T$ representing a network with communication topology (or information flow) $G_n$. Suppose each node $v_i$ of the digraph $G_\xi$ has the following dynamics where $u_i \in \Re^n$ is the input control signal of the $i$th vehicle:

$$\dot{\xi}_i = f(\xi_i, u_i), \qquad i \in \Gamma \tag{1}$$

then, we can define a digraph as a dynamic system represented by $G_\xi = (G_n, \xi)$, in which the evolution $\xi_i$ is governed by the network dynamics $\dot{\xi}_i = f(\xi_i, u_i)$. Let the information state with single integrator dynamic be given by:

$$\dot{\xi}_i(t) = u_i(t), \qquad i = 1, ..., n. \tag{2}$$

The basic consensus protocol can be defined by:

$$u_i(t) = -\sum_{j=1}^{n} a_{ij}(t)\left(\xi_i(t) - \xi_j(t)\right), \qquad i = 1, ..., n \tag{3}$$

where $a_{ij}(t)$ is the input of adjacency matrix $A_n \in \Re^{n\times n}$ associated to $G_\xi(t)$ and related to the vehicle $i$ and its neighbor $j$. Note that $a_{ij}(t) > 0$ when $(i, j) \in \varepsilon_n$, otherwise, $a_{ij}(t) = 0$.

**Definition 2** *(Average Consensus)* The consensus for the multi-vehicle network (2) is achieved when for every initial state $\xi_i(0)$,

$$lim_{t\to\infty}\left|\xi_i(t) - \xi_j(t)\right| = 0,$$

for $i = 1, ..., n$ and $j = 1, ..., n$.

The dynamic of the *information states* (2) can be implemented using the discrete model given by:

$$\xi_i[k+1] = \xi_i[k] + \Delta_k u_i[k], \qquad i = 1, ..., n \tag{4}$$

where $\Delta_k$ is the step size, and $\xi_i[k]$ and $u_i[k]$ are the information state and control input of the $i$th vehicle at discrete time $k$.

## 2.1 Consensus Tracking Protocol

The consensus tracking brings the information states of all vehicles to a reference state. Note that from Eq. (3), the consensus equilibrium is a weighted average of all vehicles initial states and hence constant. The consensus value is related to the interaction topology and weights $a_{ij}$ of the adjacency matrix and it is unknown *a priori*.

However, in some applications it is desirable that the consensus information states converge to a predefined value. In these cases, the convergence issues include both convergence to a common value, as well as convergence of the common state to its reference value. Therefore, let us consider a group with $n$ vehicles plus an additional and virtual leader $n+1$. The state $\xi_{n+1} = \xi_r \in \Re^n$ contains the information about reference consensus.

**Definition 3** *(Tracking Consensus)* The tracking consensus in the multi-vehicle network, (2), is achieved when for every initial state $\xi_i(0)$,

$$lim_{t\to\infty} \left|\xi_i(t) - \xi_j(t)\right| = 0$$

and

$$lim_{t\to\infty} |\xi_i(t) - \xi_r(t)| = 0$$

for $i = 1, ..., n$ and $j = 1, ..., n$.

The digraph $G_{n+1} = (\nu_{n+1}, \varepsilon_{n+1})$ is used to model the interaction among the $n+1$ vehicles (with a virtual leader). Let $A_{n+1} = [a_{ij}] \in \Re^{n+1\times n+1}$ the adjacency matrix associated to $G_{n+1}$, where $a_{ij} > 0$ if $(j, i) \in \varepsilon_{n+1}$ and $a_{i(n+1)} > 0$ if $\xi_r$ is available to vehicle $i$ for $i = 1, ..., n$. Finally, $a_{(n+1)j} = 0$ for all $j = 1, ..., n+1$ and $a_{ii} = 0$ for all $i = 1, ..., n$.

From [4], we have the following theorem for consensus tracking with a constant consensus reference state.

**Theorem 1** *Suppose that $A_{n+1}$ is constant. The consensus tracking problem with a constant consensus reference state is solved with according to Definition 3 if and only if the directed graph $G_{n+1}$ has a directed spanning tree.*

Note that vehicle $n+1$ has the information about the reference and the condition that $G_{n+1}$ has a directed spanning tree is equivalent to the condition that, at least, the virtual vehicle $n+1$ has a directed path to all other vehicles in the team, which is a guarantee to the tracking consensus of the *information states*.

The assumption that the matrix $A_{n+1}$ is constant is guaranteed by the connectivity constraint added to the ACvO in this work. We leave the case of varying topologies (non-constant $A_{n+1}$) for future work in which we will explore the constraints on the variability of $A_{n+1}$ that still allow reaching consensus.

**Assumption 1** *We assume that the digraph $G_{n+1}(t|t=0)$ has a directed spanning tree.*

The *Assumption 1* guarantees that the initial oriented communication graph, $G_{n+1}(t|t = 0)$, has, at least, one path connecting all the vehicles, including the virtual leader, $n + 1$, which, contains the reference information.

The main goal of the next section is to develop control laws that guarantee that each vehicle of the group achieves trajectory consensus, which is only known by a subset of the group.

## 3 Synthesis of Control Laws via Optimization for Consensus Trajectories

This section formulates a methodology for control laws synthesis based on consensus theory as an optimization problem, and therefore, it was named by ACvO (Algorithm Consensus via Optimization). As previously mentioned, the main studies found in the literature about vehicle consensus approaches were focused on the algorithm analysis, and consequently the synthesis of control laws is ignored. Therefore, requirements such as control effort, control signal saturation, convergence rate and tracking error were not considered in such previous work. The first challenge is to define an objective function, $J_i$, with the commitment between performance indexes of requirements response and cooperation terms, especially regarding the information exchange among vehicles.

Let the preliminary $J_i$ function be as follows:

$$
\begin{aligned}
J_i[k] = &\sum_{j=1}^{n} \sum_{k=1}^{N_p} \left(\hat{\xi}_i[k] - \hat{\xi}_j[k]\right)' \delta_\xi \left(\hat{\xi}_i[k] - \hat{\xi}_j[k]\right) \\
&+ \sum_{k=1}^{N_p} \left(\hat{\xi}_i[k] - \xi_r[k]\right)' \delta_e \left(\hat{\xi}_i[k] - \xi_r[k]\right),
\end{aligned}
\tag{5}
$$

where $n$ is the number of vehicles, $N_p$ is the horizon of prediction and $\hat{\xi}_i$ is prediction of state $\xi_i$, for $i = 1, ..., n$ with $J_i$ corresponding to the $i$th vehicle.

The matrices $\delta_\xi$ and $\delta_e$ are composed according to the values of the adjacency matrix. When there is no channel of communication between $i$ and $j$ vehicles, the input parameters of the matrix $\delta_\xi$ are zero. In similar way, when the vehicle $i$ has no information about the reference, the input parameter of the matrix $\delta_e$ is zero.

The objective function presented in Eq. (5) is familiar to control laws widely utilized in the literature [1, 4, 24, 25]. Both control laws and objective function represent the trade-off between states energy and tracking reference. The advantage of the proposed methodology is to precisely include requirements that are not yet included in the optimization problem. The inclusion of control effort is straightforward, just adding:

$$J_i^{aux}[k] = \sum_{k=1}^{N_u} (\Delta u_i[k])' \lambda_u[k](\Delta u_i[k]), \tag{6}$$

where, $N_u$ is the control horizon, $\Delta u_i[k]$ is the control increment and $\lambda_u[k]$ is a math function that represents the future behavior of the system. An objective function composed by Eqs. (5) and (6) includes implicitly a trade-off among control effort, states energy and tracking error.

The objective functions (5) and (6) in the matrix form are given by:

$$\begin{aligned} J_i[k] = &\sum_{j=1}^{n} \left(\hat{E}_{\xi_i}[k] - \hat{E}_{\xi_j}[k]\right)' \Delta_{\xi_i} \left(\hat{E}_{\xi_i}[k] - \hat{E}_{\xi_j}[k]\right) \\ &+ \left(\hat{E}_{\xi_i}[k] - E_{\xi_r}[k]\right)' \Delta_{e_i} \left(\hat{E}_{\xi_i}[k] - E_{\xi_r}[k]\right) \\ &+ \left(\Delta u_{\xi_i}[k]\right)' \lambda_{u_i} \left(\Delta u_{\xi_i}[k]\right), \quad \text{for} \quad i = 1, ..., n, \end{aligned} \tag{7}$$

where,

$$\begin{aligned} \hat{E}_{\xi_i}[k] &= \left[\xi_i[k+1|k]' \;\; \xi_i[k+2|k]' \;\; \ldots \xi_i[k+Np|k]'\right]' \\ \hat{E}_{\xi_j}[k] &= \left[\xi_j[k+1|k]' \;\; \xi_j[k+2|k]' \;\; \ldots \xi_j[k+Np|k]'\right]' \\ E_{\xi_r}[k] &= \left[\xi_r[k+1|k]' \;\; \xi_r[k+2|k]' \;\; \ldots \xi_r[k+Np|k]'\right]' \\ \Delta u_{\xi_i}[k] &= \left[\Delta u_i[k|k]' \;\; \Delta u_i[k+1|k]' \;\; \ldots \Delta u_i[k+Np-1|k]'\right]'. \end{aligned}$$

The prediction of states to horizon $N_p$, according to Eq. (4), and the increment of the control inputs can be presented under the following matrix form:

$$\hat{E}_{\xi_i} = E^0_{\xi_i} + T \; U_i \tag{8}$$

$$\Delta u_{\xi_i} = U^0_{\xi_i} + U_{aux} U_i, \quad \text{for} \quad i = 1, ..., n \tag{9}$$

where, $\hat{E}_{\xi_i} \in \Re^{N_p \times 2}$ is a states prediction matrix. $E^0_{\xi_i} \in \Re^{N_p \times 2}$ is a matrix with state $\xi_i$ at begin of the horizon prediction $k$, $T \in \Re^{N_p \times N_p}$ is a matrix composed by $\Delta_k$ and $U_i \in \Re^{N_p \times 2}$ is a vector with the future control inputs. In second line, $\Delta u_{\xi_i} \in \Re^{N_u \times 2}$ is a matrix of control increments, $U^0_{\xi_i} \in \Re^{N_u \times 2}$ is a matrix with control $U_{\xi_i}$ at time $k$, $U_{aux} \in \Re^{N_u \times N_u}$ is an auxiliary matrix to process the difference between increments of control and $U_i \in \Re^{N_u \times 2}$ is a vector with future control inputs.

*Remark 1* Note that Eq. (7) contains both control inputs of vehicles $i$ and $j$ ($U_i$ and $U_j$, respectively). Remember that since the control law is decentralized, for each vehicle $i$, the *information state* update can not include the future control inputs of the neighbor vehicle. Note that, to solve the problem for $U_i$, the optimal future values of $U_j$ are unknown yet, at time $k$.

A possible solution to the problem stated in *Remark 1* is to rewrite the decision vector as $U = [U_1, ..., U_n]$. However, implementing this arrangement, the problem becomes centralized, and more, it neglects the directed graph characteristic, i.e., input $a_{ij}$ has the same meaning that $a_{ji}$. Instead we assume that the neighboring prediction states are unknown, since it is not possible to use the optimal control sequence of the neighboring state. The objective function $J_i$ only implements the current state $k$ of the neighbor $j$. The new objective function is given by:

$$\begin{aligned} J_i^{new} = &\sum_{j=1}^{n} \left( E_{\xi_i}^0 + T\ U_i - E_{\xi_j}^0)' \Delta_{\xi_i} (E_{\xi_i}^0 + T\ U_i - E_{\xi_j}^0 \right) \\ &+ \left( E_{\xi_i}^0 + T\ U_i - E_{ref})' \Delta_{e_i} (E_{\xi_i}^0 + T\ U_i - E_{ref} \right) \\ &+ \left( U_{\xi_i}^0 + U_{aux} U_i)' \lambda_{u_i} (U_{\xi_i}^0 + U_{aux} U_i \right) \end{aligned} \tag{10}$$

with $i, j = 1, ..., n$. Separating the terms with $U_i$ in Eq. (10) leads to:

$$\begin{aligned} J_i^{new} = &\sum_{j=1}^{n} U_i' \left( T' \Delta_\xi T + T' \Delta_e T + U_{aux}' \lambda_{u_i} U_{aux} \right) U_i \\ &+ 2 \left( E_{\xi_i}^{0\,\prime} \Delta_\xi T - \sum_{j=1}^{n} \left( E_{\xi_j}^{0\,\prime} \Delta_\xi T \right) + E_{\xi_i}^{0\,\prime} \Delta_e T - E_{ref}' \Delta_\xi T + U_{\xi_i}^{0\,\prime} \lambda_{u_i} U_{aux} \right) U_i \\ &+ \left( E_{\xi_j}^{0\,\prime} \Delta_\xi E_{\xi_j}^0 + E_{\xi_i}^{0\,\prime} \Delta_\xi E_{\xi_i}^0 - 2(E_{\xi_j}^{0\,\prime} \Delta_\xi E_{\xi_i}^0 + U_{\xi_i}^{0\,\prime} \lambda_{u_i} U_{\xi_i}^0) \right). \end{aligned} \tag{11}$$

## 3.1 *Quadratic Formulation Problem*

Our goal is to minimize the cost function $J_i^{new}$, therefore, the constant terms in Eq. (11) can be eliminated. Moreover, defining the auxiliary matrices $H_i$ and $f_i$ as follows:

$$\begin{aligned} H_i &= T' \Delta_\xi T + T' \Delta_e T + U_{aux}' \lambda_{u_i} U_{aux} \\ f_i &= E_{\xi_i}^{0\,\prime} \Delta_\xi T - \sum_{j=1}^{n} \left( E_{\xi_j}^{0\,\prime} \Delta_\xi T \right) + E_{\xi_i}^{0\,\prime} \Delta_e T - E_{ref}' \Delta_\xi T + U_{\xi_i}^{0\,\prime} \lambda_{u_i} U_{aux}. \end{aligned}$$

allows expressing the minimization of $J_i^{new}$ in Eq. (11) as a quadratic formulation problem given by:

$$\begin{aligned} \min_{U_i} \quad & \frac{1}{2} U_i^T H_i U_i + f_i^T U_i \\ s.t. \quad & c_{\min} \leq U_i \leq c_{\max} \end{aligned} \tag{12}$$

This methodology allows the addition of constraints into the optimization problem in straightforward way. For example, the control signal saturation is implemented defining minimum and maximum values, $\underline{U} = c_{\min}$ and $\bar{U} = c_{\max}$, respectively, as shown in Eq. (12).

### 3.2 On the Consensus Trajectories and Vehicles Sensing

The ACvO protocol computes the consensus trajectory associated with the *information state*, only, which is the understanding of each vehicle about the meeting point. We assume the use of local controllers to ensure that the vehicle targets the desired position at every sampling time. It does not consider possible mechanical constraints in the motion of the vehicle, e.g., the orientation constraints of non-holonomic mobile robots.

Figure 1a shows an illustrative example, where a vehicle with local controller, even following the trajectory can fail in meeting non-holonomic constraints (*lower trajectory*). The vehicle orientation, at time $k$, is opposed to the direction of the next point, and thus, because of non-holonomic constraint, the vehicle makes a turn (*upper trajectory*), delaying its route to the consensus point, needing one more iteration to reach the trajectory.

Moreover, assuming that the vehicle has a fixed camera, in which the sensing range can be associated to the vehicle orientation, we added an optimization routine on the sensing range motivated by the knowledge of the future control sequence (optimization of $J_i$) and hence, a prediction of all points of the trajectory. The opti-

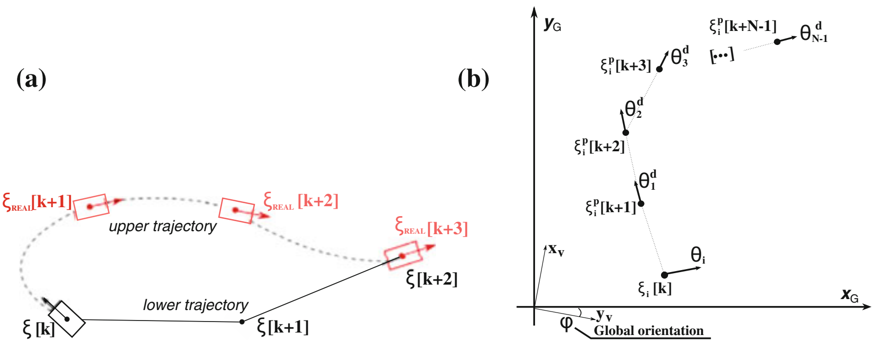


**Fig. 1** **a** Illustrative example with non-holonomic constraint. **b** Optimization problem to sensing orientation

mization of vehicle orientation can be performed by minimizing the squared error, according to Fig. 1b.

Let the cost function:

$$J_i^\theta = \sum_{k=1}^{N_p-1} \left\| \theta_i^d[k+1] - \theta_i[k+1] \right\|^2, \qquad i = 1, ..., n \tag{13}$$

where, $\theta_i^d$ is the desirable orientation (for sensing purposes) and $\theta_i$ is the actual orientation of the $i$th vehicle. The goal is to find the value of $\theta_i$ that minimizes $J_i^\theta$. Some constraints can be imposed into the problem, such as the maximum individual rotation $\delta_d$ and curvature radius $r_c$ of the vehicle. Thus, the new *information states* associated to coordinates $x$ and $y$ are:

$$\begin{aligned} \xi_i^x[k+1] &= \xi_i^x[k] \cos(\theta_i[k+1] + \varphi) \\ \xi_i^y[k+1] &= \xi_i^y[k] \sin(\theta_i[k+1] + \varphi) \end{aligned} \tag{14}$$

where $\varphi$ is the rotation related to global reference, since $\delta_d$ and $\theta_i$ are local variables, Fig. 1b. Note that each iteration of the ACvO, the maximum rotation is $3\delta_d$ (due to mechanical constraints and saturation of the control signal) and the final value of $\theta_i$ is defined by optimizing $J_i^\theta$.

It is important to note that in Eq. (14), the control values do not appear, these values have already been calculated in the previous section. This procedure is complementary, and the goal is to correct the orientation of the vehicle, which may or may not have implication on the computed consensus trajectory.

## *3.3 On the Connectivity Constraint*

As previously mentioned, one of the advantages of the proposed method is the possibility of adding constraints to the problem in straightforward way. In the optimization problem in Eq. (12), it is possible to add inequality constraints ($AU_i \leq B$) on the *information states* according to the application.

For example, conditions may be formulated to avoid collision between the vehicles, but also related to obstacles in the environment, and for UAVs, imposing a minimum height of flight. In this paper we are concerned with one situation in particular, where it is formulated a constraint related to the connectivity of the topology, and it aims at keeping any two neighbor vehicles within a *connectivity radius*.

In the connectivity constraint the main idea is that once defined that $i$ is a neighbor of $j$, then they will be neighbors throughout the complete task. Thus, if there is a temporary failure in the communication channel, the restriction aims at maintaining the neighborhood relationship between the vehicles. Let $\xi_i[k+1]^{(x,y)} = \xi_i[k] + \Delta_k U_i[k]$ and $\xi_j[k+1]^{(x,y)} = \xi_j[k] + \Delta_k U_j[k]$, the Euclidean distance between the next two positions is:

$$
\begin{aligned}
d_{ij} &= \sqrt{\left(\xi_i^x[k+1] - \xi_j^x[k+1]\right)^2 + \left(\xi_i^y[k+1] - \xi_j^y[k+1]\right)^2} \\
&= \sqrt{\left(\xi_i^x[k] + \Delta_k U_i^x[k] - \xi_j^x[k] + \Delta_k U_j^x[k]\right)^2 + \left(\xi_i^y[k] + \Delta_k U_i^y[k] - \xi_j^y[k] + \Delta_k U_j^y[k]\right)^2}
\end{aligned}
\tag{15}
$$

Again making reference to Remark 1, while the ACvO is calculating the future controls $U_i$, the values of $U_j$ are still unknown. Then, the new value of the next point of the neighbor $j$ is estimated with a simple derivative. By using the Euclidean distance, the decision variable $U_i$ assumes a quadratic form. To avoid this, we use as distance metric, the *Manhattan form*, which will provide a more conservative result. The following constraint is considered:

$$
\begin{aligned}
d_{ij}^{Mah} &= \left|\xi_i^x[k] + \Delta_k U_i^x[k] - \tilde{\xi}_j^x[k+1]\right| + \left|\xi_i^y[k] + \Delta_k U_i^y[k] - \tilde{\xi}_j^y[k+1]\right| \\
&\leq r_{com},
\end{aligned}
\tag{16}
$$

where, $\tilde{\xi}_j^{(x,y)}[k+1]$ is the estimate state of the neighbor using the derivative term and $r_{com}$ is the radius of communication. This constraint is added in a straightforward way in the optimization problem in the form $AU_i \leq B$.

## 4 Implementation of Consensus Protocol Strategy

The consensus strategy is implemented based on the blocks diagram shown in Fig. 2a. At each ACvO iteration, the optimization of $J_i$ generates the trajectory to $N_p$ points horizon based on the information exchanged, and more, with the knowledge of all trajectory prediction, the optimal orientation of vehicle $i$th is also calculated (i.e., optimization of $J_i^\theta$). (Note that the calculus of $J_i^\theta$ is only performed to the vehicles that have non-holonomic constraints). However, only the first point is implemented and we assume that the vehicle has a controller and local sensing to achieve this local target point. As a result, we have a feasible consensus trajectory for all vehicles.

The simulations were performed in *Matlab*, where we considered some communication errors (it is admitted temporary fail in some communication channels between vehicles). Note that these losses do not change the communication topology permanently. Since, the failures are temporary, the effect on the network topology just lasts a few algorithm iterations, and then, the channel is reestablished.

The topologies shown in Fig. 2b and c are used in order to evaluate the performance of the proposed algorithm. The topologies were chosen arbitrarily and the arrows indicate the direction of information flow. In Fig. 2, nodes 1 to 6 represent the vehicles while their respective *information states*, are expressed in $\xi_n$, with $n = 1, ..., 6$. We define that vehicles 1 and 2 are WMRs with non-holonomic constraint, 3 and 4 have no motion restriction and 5 and 6 are UAVs. The state $\xi_r$ is used to define the

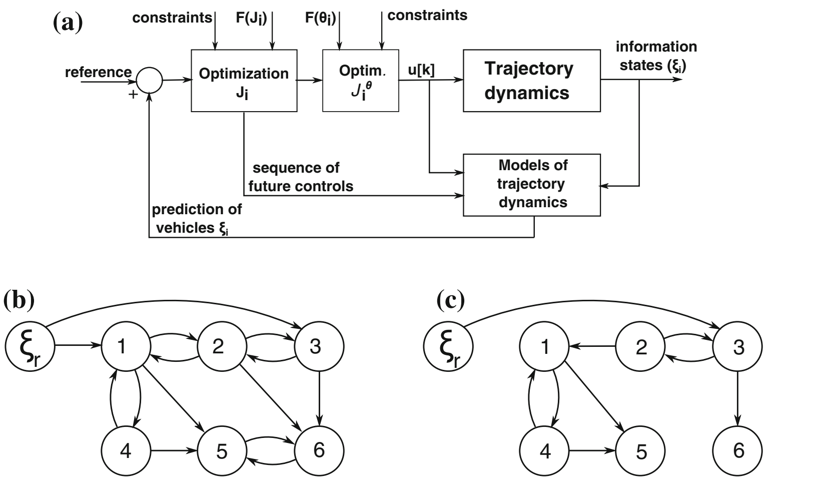


**Fig. 2** **a** Blocks diagram to implement consensus strategy via optimization; **b** Topology A; **c** Topology B. Both topologies are used in illustrative examples applying the ACvO algorithm

reference information ($\xi_r^A = [100, 100]$ and $\xi_r^B = [190, 100]$ to topologies A and B, respectively) and its information is available to a few vehicles in the group, only.

In few words, the simulation scenario is characterized by oriented information flow, limited knowledge and communication, and group with heterogeneous vehicles.

Figure 3 shows two different arrangement to the initial positions of all vehicles. The initial positions were defined as follows: in topology A, $\xi_1 = [50, 120]$, $\xi_2 = [120, 180]$, $\xi_3 = [170, 180]$, $\xi_4 = [70, 30]$, $\xi_5 = [40, 50]$ and $\xi_6 = [130, 150]$; and in topology B, $\xi_1 = [35, 100]$, $\xi_2 = [50, 150]$, $\xi_3 = [80, 190]$, $\xi_4 = [45, 30]$, $\xi_5 = [20, 30]$ and $\xi_6 = [40, 195]$.

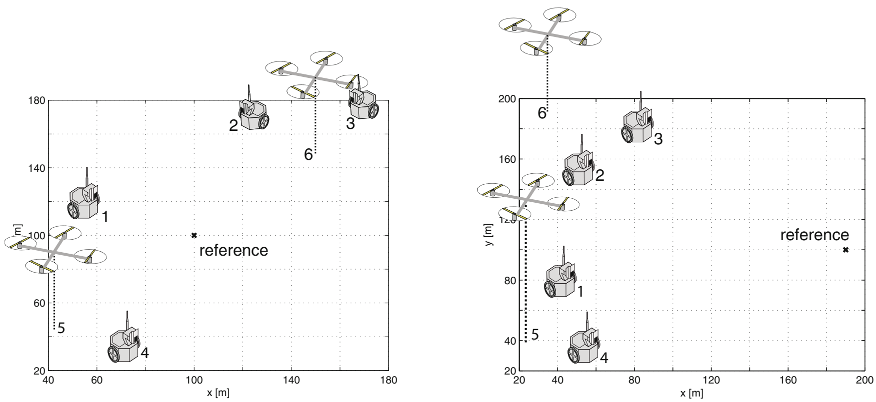


**Fig. 3** Initial positions of vehicles for topology A (*left*) and topology B (*right*)

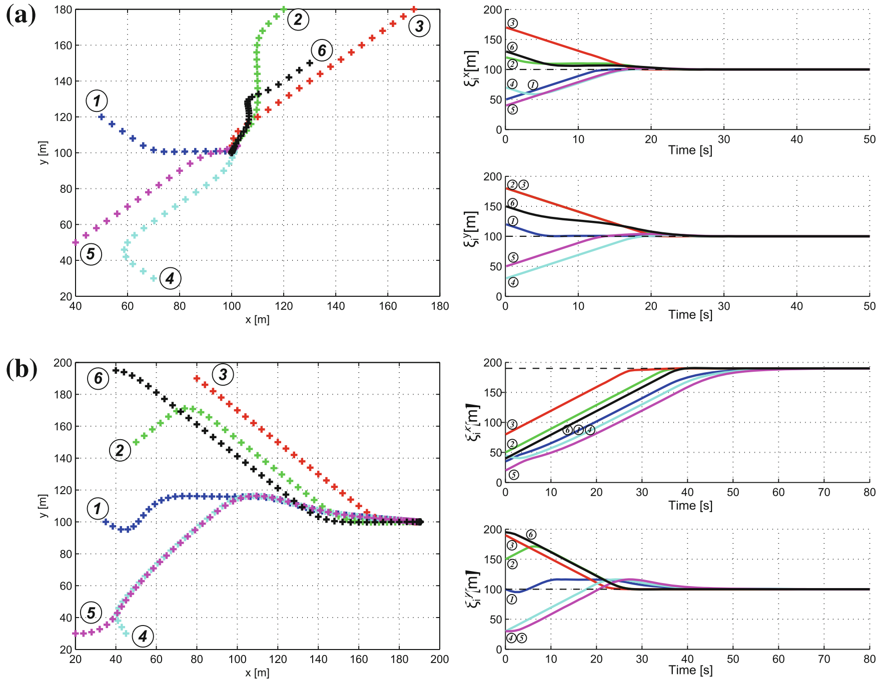


**Fig. 4** Dynamics of the *information states* $\xi_i$ applying ACvO protocol without integration of mobile robots sensing. **a** Topology A and **b** topology B

## *4.1 Numerical Simulation and Analysis of Results*

Figure 4 shows the results obtained with the application of the ACvO protocol without the optimization of sensing angle of the vehicles. In other words, only the consensus tracking and control effort are considered in the objective function $J_i^{new}$, Eq. (11).

The first goal using only the optimization of $J_i^{new}$ is to evaluate the impact of the simplification made in the objective function, in which we only implemented the current state of the neighbors (*Remark 1*). Figure 5 shows the number of iterations required for tracking consensus using a comparison of the decentralized and centralized (knowledge of predicted states in the neighbors) approaches. As a complement, we added a comparison with an algorithm with fixed parameters presented in [4] because it is very similar to other algorithms found in literature [1, 10, 11, 24].

As expected, the results show that the centralized approach has a faster convergence to the tracking consensus, but the difference between the performances is not significant (only ACvO vs. ACvO centralized). In comparison with the algorithm with fixed parameter the results presented reaffirm that the application of the ACvO allows a quick convergence of the vehicles *information state* to the reference state.

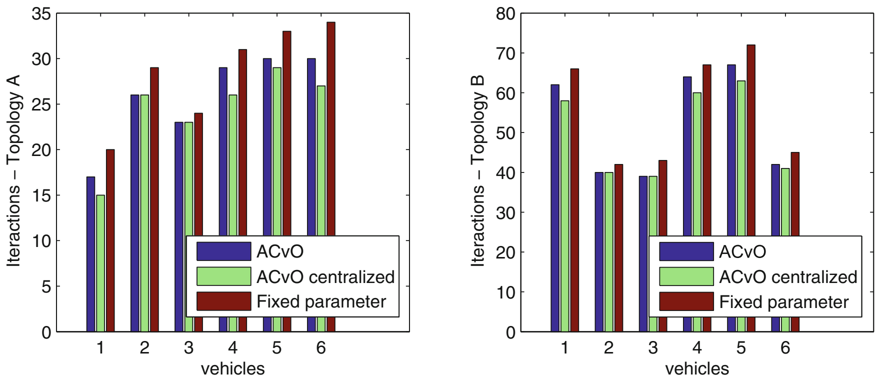


**Fig. 5** Tracking consensus convergence iterations for each vehicle of the group

Therefore, we can use our proposed ACvO protocol assuming limitations in the knowledge of the neighbors state without compromising the performance of tracking consensus. (Note that the convergence of the *information states* can be associated to the tracking error).

## *4.2 On the Computational Time*

Since the ACvO protocol is to be used on line, the computational time used to solve the optimization problem should be analyzed. In the simulations performed the sampling time was $\Delta_k = 0.5$ s. Therefore, time consumed to compute the control law should then be less than $\Delta_k$, for all $k$.

For each iteration $k$, the algorithm computes the optimization routines sequentially considering synchronized communication. Figure 6 shows the average computational

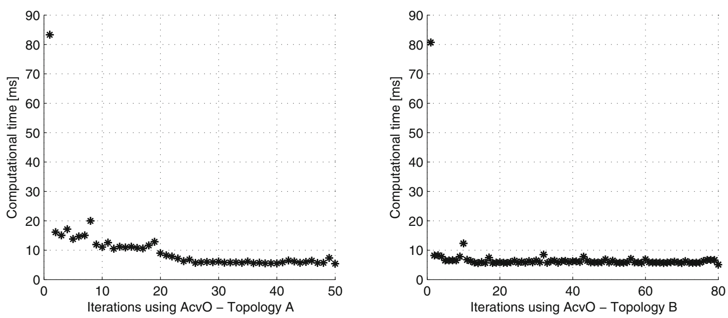


**Fig. 6** Average computational time for each iteration of the ACvO

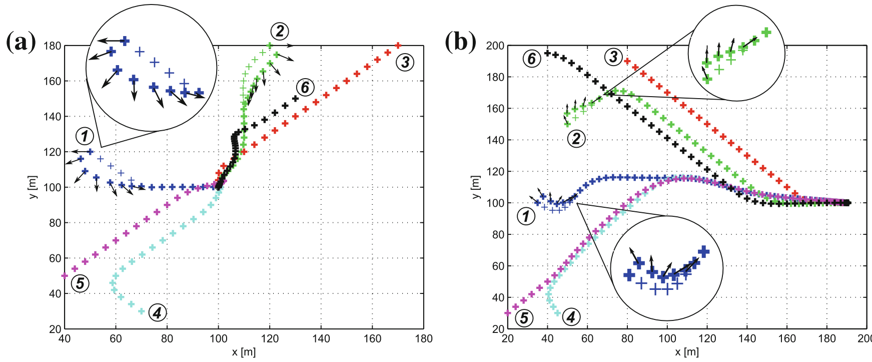


**Fig. 7** Dynamics of *information states* $\xi_i$ applying ACvO with integration of mobile robots sensing. **a** Topology A and **b** topology B

time consumed by the ACvO to compute all the 6 control laws shown in the Fig. 4. The sampling time constraint was respected in the previous experiment, using computers with *Intel Centrino Duo T2250* processor clocked at 1.73 GHz and with 2 GB of memory.

It is important to note that the computational time depends on the neighbors cardinality. In Eq. (11) is shown that the control law is directly proportional to the number of $j$th neighbors vehicles (remember that the control laws are fully decentralized). Another important aspect of the computational time is that the control laws are formulated so that $\Delta_{\xi_i} \geq 0$ and $\Delta_{e_i} \geq 0$ ensuring the convexity of quadratic formulation (12). Therefore, the solution of (12) is comparable to the resolution of a linear programming problem and the optimal global can be found [26].

**Optimization of the vehicles sensing angle**: Figure 7 shows the results obtained with the application of the ACvO protocol considering the add of the vehicles sensing angle optimization. (the algorithm development was shown in Sect. 3.2).

In the examples, we can see the difference between the behavior of the *information states* when considering the non-holonomic constraint in vehicles 1 and 2. Note that, at the initial iterations, the main objective is to guide vehicle 1, and 2, to the rendezvous point according to the calculated consensus trajectory.

The results presented confirm that the ACvO protocol allows that the *information states* reach tracking consensus. Adding another goal, the vehicle orientation, the result of the group consensus was not compromised and all vehicles were able to perform a trajectory according to their motion constraints.

**Optimization with connectivity constraint**: Figure 8 shows the results obtained with the application of the ACvO algorithm using the connectivity constraint. This example used topology B and the development of strategy was shown in Sect. 3.3.

This illustrative example shows the influence that the connectivity constraint has on the algorithm implementation. The communication channel between vehicles 1

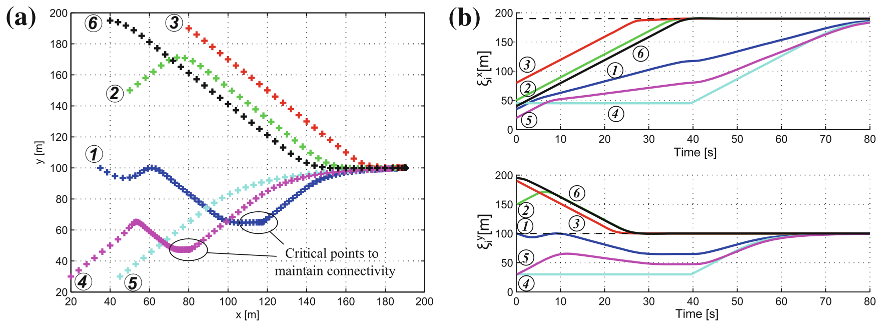


**Fig. 8** Dynamics of *information states* $\xi_i$ applying ACvO protocol with integration of connectivity constraint using topology B

and 4 was purposely disabled during the first seconds of simulation. Notice that the vehicles 1 and 5, when reaching the limit distance bound of the connectivity constraint, stop moving and wait for the reestablishment of the $a_{14}$ channel. After recovering communication, the vehicles 1, 4 and 5 continue computing their trajectories to the tracking consensus.

Note that the connectivity constraint has another positive impact. Since the $a_{14}$ communication channel was disabled, vehicle 4 behaves like a false reference to vehicles 1 and 5. However, with the connectivity constraint and communication between vehicles 1 and 2 (link to the other part of the group), vehicles 1 and 5 do not see 4 as a reference.

## 5 Conclusions

It was presented a methodology for the synthesis of decentralized control laws in order to track trajectories based on consensus. With this approach, using predictive control theory, the cost function was defined by the trade-off between the group cooperation and response requirements. According to this approach, where the addition of terms in the cost function is straightforward allowing more flexibility in design, the methodology presents the following contributions: the first one, the optimization of the sensing range with respect to the rendezvous point, and the second one, the addition of a connectivity constraint to achieve consensus.

The implementation of the ACvO protocol is decentralized and can be used in applications with heterogeneous vehicles in a group with limited knowledge and oriented information flow. The simulations were performed with various scenarios and an evaluation of the results obtained considering the convergence rate performance and computational time confirms that the ACvO protocol allows that the *information states* reach tracking consensus.

In future work we will address time varying topologies and the impact of packet losses inherent to wireless communication in the formulation of the AcvO protocol. Moreover, it is important to establish minimum conditions and correlations between packet losses and the variation of network topology in order to achieve consensus. Another issue that can be explored is the use of the presented formulation based on predictive control applied to other problems, such as formation control, coverage and flocking.

## References


1. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proc. IEEE **95**, 215–234 (2007)
2. Schurr, N., Okamoto, S., Maheswaran, R., Scerri, P., Tambe, M.: Evolution of a Teamwork Model, Cognitive Modeling and Multi-Agent Interactions. Cambridge University Press, Cambridge (2005)
3. Murray, R.M.: Recent research in cooperative control of multivehicle systems. J. Dyn. Syst. Meas. Control **129**, 571–584 (2007)
4. Ren, W., Beard, R.W.: Distributed Consensus in Multi-Vehicle Cooperative Control—Theory and Applications. Communications and Control Engineering. Springer, London (2008)
5. Ren, W.: Consensus tracking under directed interaction topologies: algorithms and experiments. IEEE Trans. Control Syst. Technol. **18**, 230–237 (2010)
6. Ordonez, B., Moreno, U.F., Cerqueira, J., Almeida, L.: Generation of trajectories using predictive control for tracking consensus with sensing. Procedia Comput. Sci. **10**, 1094–1099 (2012)
7. Chatterjee, S., Seneta, E.: Towards consensus: some convergence theorems on repeated averaging. J. Appl. Probab. **14**, 89–97 (1977)
8. DeGroot, M.H.: Reaching a consensus. J. Am. Stat. Assoc. **69**, 118–121 (1974)
9. Olfati-Saber, R., Murray, R.M.: Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans. Autom. Control **49**, 1520–1533 (2004)
10. Saber, R.O., Richard, S., Murray, R.M.: Consensus protocols for networks of dynamic agents. In: American Control Conference, Denver, 2003
11. Lin, Z., Broucke, M., Francis, B.: Information flow and cooperative control of vehicle formations. IEEE Trans. Autom. Control **49**, 1465–1476 (2004)
12. Ren, W., Beard, R.W.: Consensus seeking in multiagent systems under dynamically changing interaction topologies. IEEE Trans. Autom. Control **50**, 655–661 (2005)
13. Beard, R.W., McLain, T.W., Nelson, D., Kingston, D., Johanson, D.: Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. In: Proc. IEEE **94**, 1306–1324 (2006)
14. Xiao, L., Boyd, S.: Fast linear iterations for distributed averaging. In: IEEE Conference on Decision and Control, Atlantis, 2004
15. Decastro, G.A., Paganini, F.: Convex synthesis of controllers for consensus. In: American Control Conference, Boston, 2004
16. Dunbar, W., Murray, R.: Distributed receding horizon control for multi-vehicle formation stabilization. Automatica **42**, 549–558 (2006)
17. Bauso, D., Giarre, L., Pesenti, R.: Mechanism design for optimal consensus problems. In: IEEE Conference on Decision and Control, San Diego, 2006
18. Semsar-Kazerooni, E., Khorasan, K.: Optimal consensus algorithms for cooperative team of agents subject to partial information. Automatica **44**, 2766–2777 (2008)
19. Listmann, K.D., Masalawala, M.V., Adamy, J.: Consensus for formation control of nonholonomic mobile robots. In: IEEE International Conference on Robotics and Automation, New Jersey, 2009

20. Semsar-Kazerooni, E., Khorasani, K.: An LMI approach to optimal consensus seeking in multi-agent systems. In: American Control Conference, St. Louis, 2009
21. Jakovetic, D., Xavier, J., Moura, J.: Weight optimization for consensus algorithms with correlated switching topology. IEEE Trans. Signal Process. **58**, 3788–3801 (2010)
22. Cao, Y., Ren, W.: Optimal linear-consensus algorithms: an LQR perspective. IEEE Trans. Syst. Man Cybern. Part B Cybern. **40**, 819–830 (2010)
23. Kuwata, Y., How, J.P.: Cooperative distributed robust trajectory optimization using receding horizon MILP. IEEE Trans. Control Syst. Technol. **19**, 423–431 (2010)
24. Fax, J.A., Murray, R.M.: Information flow and cooperative control of vehicle formations. IEEE Trans. Autom. Control **49**, 1465–1476 (2004)
25. Ren, W., Beard, R.W., Atkins, E.M.: Information consensus in multivehicle cooperative control. IEEE Control Syst. Mag. **27**, 71–82 (2007)
26. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)

Springer

http://www.springer.com/978-3-642-39300-6

Cooperative Robots and Sensor Networks
Koubaa, A.; Khelil, A. (Eds.)
2014, X, 98 p. 51 illus., Hardcover
ISBN: 978-3-642-39300-6