Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Distributed Parameterized Predictive Control for Multi-robot Curve Tracking

Gabriel V. Pacheco * **Luciano C. A. Pimenta** ** **Guilherme V. Raffo** **

* *Graduate Program in Electrical Engineering - Universidade Federal de Minas Gerais - Belo Horizonte, MG, Brazil (e-mail: gabriel.pacheco@ufmg.br).*
** *Department of Electronic Engineering - Universidade Federal de Minas Gerais - Belo Horizonte, MG, Brazil (e-mail: lucpim@ufmg.br; raffo@ufmg.br).*

Abstract: This work proposes a guidance strategy of multiple robots to converge and circulate a curve while avoiding collisions by using a distributed model predictive control. To build the model predictive control framework, systems guided by control laws with parameters are considered, which laws are embedded in the optimization problem. After that, the same problem is distributed using the Alternating Direction Method of Multipliers and nonlinear optimization. To solve the task of convergence and circulation of a closed path, a vector field based control law is embedded in the predictive control scheme. The control law results from the sum of two components, a convergence term and a circulation term, whereas each term has one proportional parameter associated. Numerical results present an application example, and the strategy effectiveness is discussed.

Keywords: Vector fields, distributed model predictive control, alternating direction method of multipliers, multi-robot systems, coordination.

1. INTRODUCTION

The control of multi-robot systems has been widely studied in the past decades, due to the increased ability to execute certain tasks that could not be accomplished by single agents. Some interesting tasks require convergence and circulation of a desired curve, such as perimeter surveillance (Pimenta et al., 2013), tracking a moving target (Briñón-Arranz et al., 2019), and environmental monitoring (DeVries and Paley, 2012). To converge to limit cycles, vector field-based control strategies have been used (Jung et al., 2016; Frew and Lawrence, 2017; Gonçalves et al., 2010b). The main advantages of that type of strategy are the stability guarantees and the intrinsic robustness of vector field approaches.

Regarding multi-robot systems, the ability to optimize the system's trajectories distributively attracts the attention of researchers to distributed model predictive control (DMPC) approaches. The major advantage of optimal control strategies is the capability of minimizing (or maximizing) a cost functional, taking into account the constraints. By considering a receding horizon, MPC provides feedback to the system, which increases the robustness in front of external disturbances and parametric uncertainties (Rawlings and Mayne, 2009). In complex and large systems, to employ a centralized controller might be infeasible, as high computational burden is involved. Consequently, it is generally beneficial to implement decentralized or distributed strategies (Maestre and Negenborn, 2014). Furthermore, computing the control signals in a distributed manner increases robustness in front of system failure, since the optimization does not depend on a unique central processing unit.

To alleviate the computation of optimal control problems, parametrization of trajectories and control laws has been proposed. In Droge and Egerstedt (2013), e.g., a DMPC scheme is set considering parameterized control laws that are embedded in the optimal control problem, in which the parameters are chosen optimally. The global cost is optimized using the dual-decomposition method implemented via gradients. A parametrization of trajectories considering splines is developed in Van Parys and Pipeleers (2017). In that work, the agents' dynamics are considered to be diferentially flat, which allows enforcing formation and input constraints. To distribute the optimization, the authors apply one iteration of the Alternating Direction Method of Multipliers per MPC cycle, which lowers the computational cost.

In this work, we proposed a multi-robot control system based on a distributed predictive control strategy considering parameterized control laws, i.e. a control law with tunable parameters. As the distributed predictive control needs to predict and agree on the trajectories of the neighboring agents, the use of parameterized control laws is convenient assuming that the trajectory may be predicted with the initial state, the control law parameters, and the agent dynamics. Furthermore, the strategy allows to employ well-established control laws for certain a system.

To solve the optimization problem, we adopt the Alternating Direction Method of Multipliers (ADMM). In this technique, each agent has a version of the variables of the neighboring agents. To aim equality between these variables, the technique uses Lagrange multipliers and a penalty term that provides more robustness to the distributed optimization. As described previously, this technique has been implemented in multi-robot distributed MPC in Van Parys and Pipeleers (2017), but instead of spline-based trajectories, we use parameterized control laws as in Droge and Egerstedt (2013). The ADMM has also been implemented in Rey et al. (2018) and Ferranti et al. (2018) to solve optimal control problems regarding unmanned vehicle navigation with collision avoidance constraints.

The control laws considered in this work are based on the vector fields provided by Gonçalves et al. (2010a); Gonçalves et al. (2010b), with slight modifications. These control laws provide convergence and circulation to closed curves described by implicit functions in 2-dimensional spaces. In the parameterized MPC context, two parameters are considered: one that provides attraction to the aimed curve and one that impels the circulation of the curve. A similar parametrization without any optimal control policy has been implemented in Pimenta et al. (2013), in which priority-based modulation functions (Hsieh et al., 2008) were used to tune the convergence and circulation terms to avoid collisions.

This work presents an extension of the vector field navigation framework by regarding the problem of modulating the convergence and circulation fields as an optimization problem, which allows us to pursue collision avoidance through distributed optimal control. Besides that, a parameterized distributed optimization scheme using the alternating direction method of multipliers is proposed.

2. PROBLEM STATEMENT

In this work we address the problem of convergence and circulation of a curve by a group of robots while avoiding collisions among them. This problem is approached with the vector field navigation framework provided by Gonçalves et al. (2010a) and Gonçalves et al. (2010b), the parameterized DMPC (Droge and Egerstedt, 2013), and ADMM (Boyd et al., 2011; Wang et al., 2019).

Consider a set Ω that contains N robots ω_i, in which $i = 1, ..., N$. The robots are considered to move in the 2-dimensional space, and its states are described as $\boldsymbol{x}_i(t) = [x_1\, x_2]$ at the time instant t. From the vector field framework, the robots are assumed to be single integrators

$$\dot{\boldsymbol{x}}_i = \boldsymbol{u}_i. \tag{1}$$

Each robot ω_i has a communication range δ. The set of agents within the perceived sphere of ω_i is given by

$$\Xi_i \triangleq \{\omega_j \in \Omega \mid ||\boldsymbol{x}_i - \boldsymbol{x}_j|| \leq \delta,\ j \neq i\}, \tag{2}$$

and consequently, if ω_i is in the perception range of ω_j, ω_j is in the perception range of ω_i, that is

$$\omega_i \in \Xi_j \Leftrightarrow \omega_j \in \Xi_i. \tag{3}$$

To accomplish the task, the group of robots must converge and circulate a curve implicitly defined by the function $\phi(x_1, x_2) = 0$. Further assumptions on the choice of $\phi(x_1, x_2)$ are made in Section 5. The problem to be solved may be summarized as following.

Problem Statement *1.* Given a group of robots $\omega_i \in \Omega$ for $i = 1, ..., N$ with dynamics (1) and communication range δ, design a control strategy capable of providing convergence and circulation of a curve $\phi(x_1, x_2) = 0$ while avoiding collisions among robots.

3. OPTIMAL CONTROL PROBLEM

In order to solve the problem described in Section 2, we propose a parameterized MPC scheme distributed by the ADMM. The MPC strategy consists basically in computing the optimal control sequence that minimizes a cost functional in a prediction horizon Δ, then applying the first control signal and restarting the cycle (Rawlings and Mayne, 2009). In parameterized MPC (Droge and Egerstedt, 2013), parameterized control laws are embedded in the optimal control problem such that the parameters are decision variables.

Based on the vector field framework and assuming the agents as simple integrators, the closed loop dynamics, considering parameterized control laws $\boldsymbol{u}_i = \boldsymbol{\kappa}_i(\boldsymbol{x}_i, \boldsymbol{\theta}_i)$, are given by

$$\dot{\boldsymbol{x}}_i = \boldsymbol{\kappa}_i(\boldsymbol{x}_i, \boldsymbol{\theta}_i), \tag{4}$$

in which $\boldsymbol{x}_i \in \mathbb{R}^2$ is the state vector, and $\boldsymbol{\theta}_i \in \mathbb{R}^p$ is the parameter vector with p parameters.

By implementing parameterized control laws, the number of decision variables in the optimal control problem is reduced from a control sequence to a small set of parameters. Besides that, since the problem is solved with a lesser degree of freedom, less ability to minimize the cost is expected. Regarding distributed multi-robot systems, the parameterized predictive control strategy is convenient, since it allows the computation of agents' trajectories by the neighboring agents with limited amount of data: the initial state, the control law parameters and the dynamics of the agents. In this case, it is not necessary to compute and to communicate the entire control sequence (Droge and Egerstedt, 2013).

With regard to the problem previously stated, an optimal control problem is set. Considering the entire set Ω, the global optimization problem is given by

$$\begin{aligned} &\min_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{\theta}}} \int_{t_0}^{t_f} L_g(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\theta}}) dt \\ &\text{subject to} \quad \dot{\bar{\boldsymbol{x}}} = \bar{\boldsymbol{\kappa}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\theta}}), \end{aligned} \tag{5}$$

in which $\bar{\boldsymbol{x}}$ is the generalized state vector containing all agents states, $\bar{\boldsymbol{\theta}}$ is the generalized parameters vector and $\bar{\boldsymbol{\kappa}}$ is the generalized control law. The term $L_g(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\theta}})$ refers to the global stage cost, that must be designed regarding the accomplishment of the task, that is, driving the solution of the optimization problem to obtain convergence, circulation and collision avoidance by the agents. Also, note that parameterized MPC is not limited to the task proposed in this work, but suits for every multi-robot task that can be modeled as a global optimal control problem. Concerning predictive control, the final time is set to be $t_f = t_0 + \Delta$.

To solve the problem (5) in a distributed manner, the global cost must be split among agents. Considering the split of (5) regarding the communication ranges, one can obtain

$$
\begin{aligned}
\min_{\bar{\boldsymbol{x}},\bar{\boldsymbol{\theta}}} \quad & \sum_{i=1}^{N} \int_{t_0}^{t_f} L_i(\bar{\boldsymbol{x}}_i,\bar{\boldsymbol{\theta}}_i)dt \\
\text{subject to} \quad & \dot{\bar{\boldsymbol{x}}}_i = \bar{\boldsymbol{\kappa}}_i(\bar{\boldsymbol{x}}_i,\bar{\boldsymbol{\theta}}_i), \quad \forall i \in \{1,...,N\}, \\
& \bar{\boldsymbol{\theta}}_{ij} = \bar{\boldsymbol{\theta}}_{jj}, \quad \forall \omega_j \in \Xi_i,
\end{aligned} \tag{6}
$$

in which the generalized vectors $\bar{\boldsymbol{x}}_i$ and $\bar{\boldsymbol{\theta}}_i$ contain the positions and parameters of the j-th agents such that $j \in \Xi_i \cup \{i\}$ as computed by the i-th agent. Furthermore, $\bar{\boldsymbol{\theta}}_{ij}$ is the parameter vector of the j-th agent as computed by the i-th agent. We emphasize that the i-th agent computes its own variables and the variables of the agents in the perception set Ξ_i.

To split the global cost considering the communication range, the global cost, and, consequently, the individual costs must be varying. At each MPC cycle, the sets of agents inside the communication range Ξ_i $\forall i$ may change. Indeed, if the i-th agent does not communicate with an agent j, there must be no coupling between these agents in the optimal control problem.

The following sections describe the distributed optimization scheme used to solve the problem (6) and the vector field control laws.

4. ADMM BASED PARAMETERIZED DISTRIBUTED PREDICTIVE CONTROL

In multi-agent systems, a distributed optimal control scheme renders more robustness, since the decision is no longer associated with a central processing unit, but with a set of local negotiations between agents. Among the classical distributed optimization methods, the ADMM is the one that provides better robustness and generally converges faster (Boyd et al., 2011).

In distributed optimization, a global cost $f(x)$ is split among N processing units. If all the processing units decide about the same decision variable, that is, the same decision variable is present at all the local cost functionals, a case of global consensus is set. Therefore, to ensure that the solution of each local optimization problem reaches the same global solution the equality constraint must be satisfied, as follows

$$
\begin{aligned}
\min_{\forall x_i} \quad & \textstyle\sum_{i=1}^{N} f_i(x_i) \\
\text{subject to} \quad & x_i - z = 0, \quad i = 1, \dots, N,
\end{aligned} \tag{7}
$$

in which x_i is the local decision variable and z is the global decision variable.

In ADMM the coupling constraints are relaxed by adding Lagrange multipliers and an additional penalty term that provides more robustness to the distributed optimization (Boyd et al., 2011). The augmented cost functional is given by

$$
L_\rho(x,z,y) = \sum_{i=1}^{N} \left(f_i(x_i) + y_i^T(x_i - z) + \frac{\rho}{2}\|x_i - z\|_2^2 \right), \tag{8}
$$

in which y_i is the Lagrange multiplier associated to the i-th processing unit, and $\rho > 0$ is a penalty constant to be chosen.

The set of iterations for the global consensus problem using ADMM is defined as following (Boyd et al., 2011)

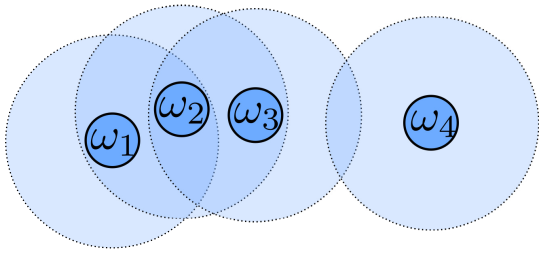

Fig. 1. Communication range example.

$$
x_i^{k+1} = \underset{x_i}{\operatorname{argmin}} \left(f_i(x_i) + (y_i^k)^T(x_i - z^k) + \frac{\rho}{2}\|x_i - z^k\|_2^2 \right) \tag{9}
$$

$$
z^{k+1} = \frac{1}{N} \sum_{i=1}^{N} \left(x_i^{k+1} + (1/\rho) y_i^k \right), \tag{10}
$$

$$
y_i^{k+1} = y_i^k + \rho\left(x_i^{k+1} - z^{k+1} \right). \tag{11}
$$

From the iteration set (9 − 11), one can observe that ADMM requires two communication phases at each cycle. First, each agent computes the x-minimization step, obtaining x_i^{k+1}. After that, the computed value is transmitted to a central collector (also named fusion unit) that gathers all the information and computes z^{k+1}. Thereafter, the z^{k+1} value is transmitted by the centralizer unit to all the agents, which compute the Lagrange multipliers y_i^{k+1}.

In this work, an agent can only communicate with agents in a certain range. Therefore, there is no global consensus, instead a set of local consensus problems in which the i-th agent is the fusion center of the $\boldsymbol{\theta}_i$ variable. This problem is named as general consensus (Boyd et al., 2011) and the solution is similar to the one previously presented. Indeed, the decision variables of the i-th agent are determined by the agent network defined by the communication range. In these conditions, each local consensus problem is equivalent to a global one, considering only the shared decision variables and the agents in the communication range. Fig. 1 illustrates the optimization scheme with a four agents example, in which the agents and their communication ranges are drawn. Agent ω_1 connects only with ω_2, therefore, only ω_1 and ω_2 influence the choice of $\boldsymbol{\theta}_1$. As ω_2 connects to ω_1 and ω_3, the three agents negotiate to achieve consensus on $\boldsymbol{\theta}_2$ value. Similarly, as ω_4 does not connect to any agent, there is no influence of other agents.

As shown in Section 3, the optimal control problem is given by (6). One may note that this global problem can be split by the ADMM technique previously described. Therefore, the cost from the problem (6) is augmented as in the ADMM distributed optimization, yielding

$$
\begin{aligned}
L_{\rho,i}(\bar{\boldsymbol{x}}_i(t),\bar{\boldsymbol{\theta}}_i,\bar{\boldsymbol{y}}_i,\bar{\boldsymbol{z}}_i) = & \int_{t_0}^{t_f} L_i(\bar{\boldsymbol{x}}_i(t),\bar{\boldsymbol{\theta}}_i)dt \\
& + \sum_{j \in \Xi_i \cup \{i\}} \left(\bar{\boldsymbol{y}}_{ij}^T(\bar{\boldsymbol{\theta}}_{ij} - \bar{\boldsymbol{z}}_{ij}) + \frac{\rho}{2}\left\|\bar{\boldsymbol{\theta}}_{ij} - \bar{\boldsymbol{z}}_{ij}\right\|_2^2 \right),
\end{aligned} \tag{12}
$$

in which $\bar{\boldsymbol{z}}_i$ is the generalized vector that contains $\bar{\boldsymbol{z}}_{ij} = \boldsymbol{z}_j$ $\forall j \in \Xi_i \cup \{i\}$. Remember that if an agent knows the initial position, the dynamics, and the parameters of a neighbor agent, the state trajectory can be computed, and therefore the consensus on the trajectories of the states is achieved via parameters.

Algorithm 1 aims to compute the control laws' parameters. In that algorithm, an external loop is responsible for incrementing the MPC horizon, while an internal loop is responsible for finding the optimal problem solution in a distributed manner. One can note that in the internal loop the sets of agents in range Ξ_i are taken into account. As in Van Parys and Pipeleers (2017) and Ferranti et al. (2018), the number of ADMM iterations per MPC cycle n_{ite} is limited, aiming to reduce the communication exchange and the computational burden. In addition, after the first ADMM iteration in each MPC cycle, the optimization problems are warm started with the previous computed solutions.

Algorithm 1 Distributed Parameterized Model Predictive Control

repeat
  $t := t + dt$;
  $t_0 := t$;
  $t_f := t_0 + \Delta$;
  $n_{ite} := 0$;
  Each agent communicates $\bar{\boldsymbol{\theta}}_{ii}$ and $\boldsymbol{x}_{ii}(t_0)$ to the agents in range.
  repeat
    In parallel, each agent solves an optimization problem
    $$\bar{\boldsymbol{\theta}}_i^{k+1} = \underset{\bar{\boldsymbol{\theta}}_i}{\operatorname{argmin}} \quad L_{\rho,i}(\bar{\boldsymbol{x}}_i(t), \bar{\boldsymbol{\theta}}_i, \bar{\boldsymbol{y}}_i^k, \bar{\boldsymbol{z}}_i^k)$$
    $$\text{subject to} \quad \dot{\bar{\boldsymbol{x}}}_i(t) = \bar{\boldsymbol{\kappa}}_i(\bar{\boldsymbol{x}}_i(t), \bar{\boldsymbol{\theta}}_i), \quad \begin{bmatrix} -\theta_{max} \\ \vdots \\ -\theta_{max} \end{bmatrix} \leq \bar{\boldsymbol{\theta}}_i \leq \begin{bmatrix} \theta_{max} \\ \vdots \\ \theta_{max} \end{bmatrix}.$$
    Each agent communicates $\bar{\boldsymbol{\theta}}_{ii}$, $\bar{\boldsymbol{\theta}}_{ij}$ and $\bar{\boldsymbol{y}}_{ij}$.
    In parallel, the i-th agent computes $\boldsymbol{z}_i^{k+1}$
    $$\boldsymbol{z}_i^{k+1} := \frac{1}{|\Xi_i|+1} \sum_{j \in \Xi_i \cup \{i\}} \left(\bar{\boldsymbol{\theta}}_{ji}^{k+1} + (1/\rho) \bar{\boldsymbol{y}}_{ji}^k \right).$$
    Each i-th agent transmits the $\boldsymbol{z}_i^{k+1}$ value.
    In parallel, each agent computes the Lagrange multipliers
    $$\bar{\boldsymbol{y}}_i^{k+1} := \bar{\boldsymbol{y}}_i^k + \rho \left(\bar{\boldsymbol{\theta}}_i^{k+1} - \bar{\boldsymbol{z}}_i^{k+1} \right).$$
    $n_{ite} = n_{ite} + 1$
  until $n_{ite} == n_{max}$
  Apply the first control signal computed.
until reaches the final time

Convergence of the ADMM regarding convex problems is a well established result (Boyd et al., 2011).

5. VECTOR FIELD NAVIGATION

The strategy used to guide the agents to converge and circulate a curve is based on those presented in Gonçalves et al. (2010a) and Gonçalves et al. (2010b), with some slight modifications. In this work, we focus on closed planar curves that can be described by a function $\phi(x_1, x_2) : \mathbb{R}^2 \rightarrow \mathbb{R}$ as the constraint $\phi(x_1, x_2) = 0$, in which x_1 and x_2 are coordinates in a Cartesian system $\boldsymbol{x} \in \mathbb{R}^2$ such that $\boldsymbol{x} = [x_1 \ x_2]^T$. To produce the desired behavior, the function $\phi(x_1, x_2)$ must be positive when evaluated outside of the desired curve, and negative inside.

The task of convergence and circulation can be solved separately. After that, the solutions are joined, generating a vector field in the form

$$\boldsymbol{v}(x_1, x_2) = \boldsymbol{N}(x_1, x_2) + \boldsymbol{T}(x_1, x_2), \tag{13}$$

in which $\boldsymbol{N}(x_1, x_2)$ is a field component normal to the contour curve at (x_1, x_2), and $\boldsymbol{T}(x_1, x_2)$ is a field component tangent to the same contour curve. The first term is responsible for providing convergence to the aimed curve, while the second is responsible for driving the circulation of the curve.

As proposed in Gonçalves et al. (2010a), a vector field-based control law $\boldsymbol{\kappa}_i(\boldsymbol{x}_i) := \boldsymbol{v}(x_1, x_2)$ in the form of (13) can be defined as

$$\boldsymbol{\kappa}_i(\boldsymbol{x}_i) = G(\phi)\nabla\phi + H(\phi)\nabla_H \phi, \tag{14}$$

for an agent i, in which ϕ is evaluated in $\boldsymbol{x}_i$. The function $G(\phi)$ is defined such that $G(\phi) > 0$ for $\phi < 0$, $G(\phi) < 0$ for $\phi > 0$, and $G(0) = 0$, $\nabla\phi$ is the gradient of ϕ, $H(\phi)$ is a function that defines the circulation sense, and $\nabla_{H\phi} = \left[-\frac{\partial\phi}{\partial x_2} \ \frac{\partial\phi}{\partial x_1} \right]^T$ is the vector orthogonal to the gradient, denominated Hamiltonian vector field. Note that the first term guarantees convergence and the second term circulation.

Similarly to Pimenta et al. (2013), we desire to avoid collisions among robots by modulating the parameters of the vector field. In this work, a parameter vector $\theta_i \in \mathbb{R}^2$ is considered, in which the first is multiplied by the convergence term and the second is multiplied by the circulation term. With the introduced parameters, the control law is given by

$$\boldsymbol{\kappa}_i(\boldsymbol{x}_i) = \theta_1 G(\phi)\nabla\phi + \theta_2 H(\phi)\nabla_H \phi. \tag{15}$$

As one can observe in (15), the vector field magnitude in each point (x_1, x_2) depends on the choice of the function ϕ. This characteristic can generate huge control signals through optimal control solved discretely, and therefore cause large displacements between two samples when assuming unbounded velocities, making collision detection difficult. In front of this, we limit the parameters between $\pm\theta_{max}$, and the vector field components are normalized

$$\boldsymbol{\kappa}_i(\boldsymbol{x}_i, \boldsymbol{\theta}_i) = \theta_1 G(\phi)\frac{\nabla\phi}{||\nabla\phi||} + \theta_2 H(\phi)\frac{\nabla_H\phi}{||\nabla_H\phi||}. \tag{16}$$

As the gradient is zero only on the center of the curve, there are no divisions by zero if the agents' initial position is considered to be different from that point, since the center of the curve is repulsive. To illustrate the influence of tuning θ_1 and θ_2 on the vector field, Fig. 2 shows three tuning situations for a given vector field. As expected, when the convergence parameter dominates the vector field, the vectors point almost directly to the curve. Conversely, when the circulation parameter dominates, the vectors are disposed almost tangentially to the curve.

To aim at convergence to the curve while avoiding collisions, the i-th agent stage cost might be chosen as

$$\begin{aligned} L_i(\bar{\boldsymbol{x}}_i(t), \bar{\boldsymbol{\theta}}_i) &= \eta_1(\bar{\boldsymbol{\theta}}_{ii,1} - \theta_{d1})^2 + \eta_2(\bar{\boldsymbol{\theta}}_{ii,2} - \theta_{d2})^2 \\ &+ \eta_3 \sum_{j \in \Xi_i} \exp\left(-\zeta \left((\bar{\boldsymbol{x}}_{ii}(t) - \bar{\boldsymbol{x}}_{ij}(t))^T (\bar{\boldsymbol{x}}_{ii}(t) - \bar{\boldsymbol{x}}_{ij}(t)) \right.\right. \\ &\left.\left. -\epsilon_e \right) \right), \end{aligned} \tag{17}$$

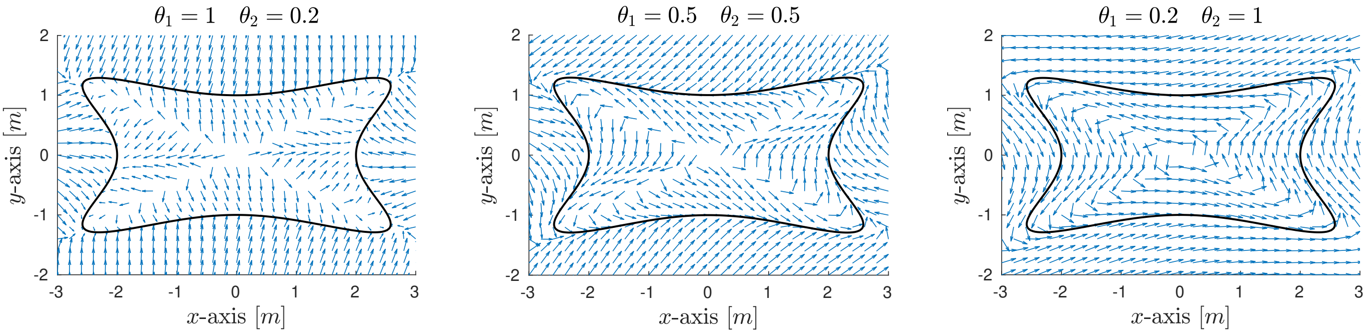

Fig. 2. Example of vector field modulation by the control law parameters.

in which η_1, η_2 and η_3 are weighting parameters, $\zeta > 0$ is a constant, ϵ_e is a safety distance, and $\bar{\boldsymbol{\theta}}_{ii} = [\bar{\boldsymbol{\theta}}_{ii,1}\ \bar{\boldsymbol{\theta}}_{ii,2}]^T$ is the parameter vector of the i-th agent computed by itself. The terms $(\bar{\boldsymbol{\theta}}_{ii,1} - \theta_{d1})^2$ and $(\bar{\boldsymbol{\theta}}_{ii,2} - \theta_{d2})^2$ penalize the cost based on the desired convergence and circulation parameters, θ_{d1} and θ_{d2}, respectively. The sum of exponential functions provides repulsion between agents.

As the elements of the parameter vectors are limited between $\pm\theta_{max}$, the three terms in (17) are bounded. Therefore, collision avoidance can be aimed by choosing η_1, η_2, η_3, ζ and θ_{max} properly.

6. RESULTS

In order to observe the complete system operation, a simulation with eight agents is proposed. As target, consider the quartic plane curve described implicitly by the function

$$\phi_1(x_1, x_2) = a x_1^4 + b x_1^2 x_2^2 + c x_2^4 - 1 = 0, \tag{18}$$

in which $a = 1/16$, $b = -2/5$ and $c = 1$. This curve is the same presented previously in Fig. 2.

The agents are guided by the control law presented in (16) in the form

$$\boldsymbol{\kappa}_i(\boldsymbol{x}_i, \boldsymbol{\theta}_i) = \theta_1 \tanh(-\phi) \frac{\nabla\phi}{||\nabla\phi||} + \theta_2 \frac{\nabla_H \phi}{||\nabla_H \phi||}, \tag{19}$$

in which the parameters θ_1 and θ_2 are chosen optimally according to the cost functional described in (17). The control laws' parameters are bounded by $-1 \leq \theta_1 \leq 1$ and $-1 \leq \theta_2 \leq 1$, and the optimization weights and variables are chosen as $\rho = 4$, $\eta_1 = 1$, $\eta_2 = 5$, $\eta_3 = 5$, $\zeta = 10$, and $\theta_{d1} = \theta_{d2} = 0.5$. Also, the robot has radius $r = 0.3m$ and $\epsilon = 4r^2$. As discussed in Gonçalves et al. (2010a), convergence and circulation of the curve happen if θ_{d1} and θ_{d2} are greater than zero. From Equation (19), one can note that the maximum velocity occurs when $|\theta_1| = |\theta_2| = \theta_{max}$, and therefore, θ_{max} has the role of limiting the maximum velocity, and θ_{d1} and θ_{d2} are defined regarding desired velocities.

The total simulation time is set $60\,s$ with step size $dt = 0.1\,s$ and prediction horizon $\Delta = 1\,s$. Besides that, each agent has a perception range $\delta = 1\,m$, and the maximum number of ADMM iterations is set $n_{max} = 5$. The optimization problems are implemented in CasADi (Andersson et al., 2019) and solved with the Ipopt package (Wächter and Biegler, 2006).

Initially, the agents are disposed in line, as presented in Fig. 3, in which each agent is represented by a colored square surrounded by a safety distance circle. In the same figure, one can observe the initial position of agents and the convergence of trajectories to the desired curve. In the final time, the agents are spread over the curve, in which the distance between agents depends on the communication range and the weight parameters.

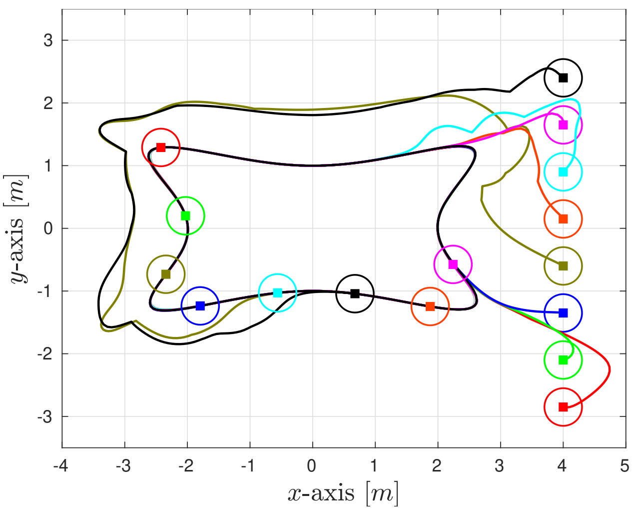

Fig. 3. Agents' trajectories.

The computed control laws' parameters are presented in Fig. 4, in which the convergence parameters are the continuous lines and circulation parameters are the dashed lines. One can observe that most of the variations occur in the first $30\,s$. Indeed, at the beginning of the simulation, the agents are next to each other, requiring modulation of the control laws to increase distance among agents. Once in the curve, small fluctuation occurs when different agents appear in the perception range of an agent. In average, each local optimization problem took $0.018\,s$ to be solved and the worst case was solved in $0.1175\,s$.

Variations of the cost functional weights are also tested. In one simulation, the weights are set as $\eta_1 = 1$, $\eta_2 = 1$, $\eta_3 = 10$ and $\eta_4 = 10$, in which collision avoidance is prioritized over convergence and circulation. Numerical results show that in this case, the set of robots stuck over the curve. In another simulation, the weights are set as $\eta_1 = 1$, $\eta_2 = 5$, $\eta_3 = 1$ and $\eta_4 = 5$, in which circulation is prioritized. As a very low weight is attributed to the collision avoidance cost, a collision happens. All the simulation videos can be watched in `https://youtu.be/1_TQ6XRnJj8`.

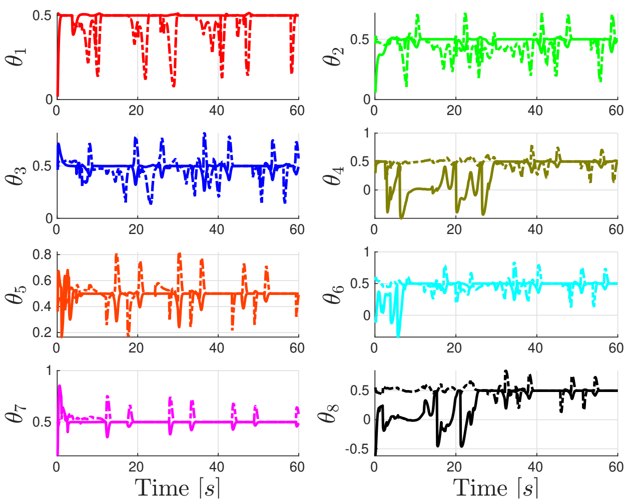

Fig. 4. Control laws parameters.

7. CONCLUSION

This work proposed a strategy to drive a convoy of agents to converge and circulate a curve while avoiding collisions. The strategy is based on embedding a control law with parameters in a predictive optimal control problem. Hence, the agent dynamics are restricted to the control law behavior, in which the control law parameters are the decision variables. The optimal control problem is solved in a distributed way using the ADMM method and considering a communication structure that varies at each control cycle.

Numerical results show that collision avoidance is obtained by including repulsive terms in the cost functional with a careful selection of weighting parameters. Future works aim to demonstrate stability of the proposed DMPC, besides experimental results.

ACKNOWLEDGEMENTS

This work was in part supported by the project INCT (National Institute of Science and Technology) under the grant CNPq (Brazilian National Research Council) 465755/2014-3, FAPESP, Brazil 2014/50851-0. This work was also partially supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil (Finance Code 88887.136349/2017-00), CNPq, Brazil (grant numbers 426392/2016-7, 313568/2017-0, 311063/2017-9), and FAPEMIG, Brazil (grant number APQ-03090-17).

Luciano C. A. Pimenta and Guilherme V. Raffo are also members of the National Institute of Science and Technology (INCT) for Cooperative Autonomous Systems Applied to Security and Environment.

REFERENCES

Andersson, J.A., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1), 1–122.

Briñón-Arranz, L., Seuret, A., and Pascoal, A. (2019). Circular formation control for cooperative target tracking with limited information. *Journal of the Franklin Institute*, 356(4), 1771 – 1788.

DeVries, L. and Paley, D.A. (2012). Multivehicle control in a strong flowfield with application to hurricane sampling. *Journal of Guidance, Control, and Dynamics*, 35(3), 794–806.

Droge, G. and Egerstedt, M. (2013). Distributed parameterized model predictive control of networked multiagent systems. In *2013 American Control Conference*, 1332–1337. IEEE.

Ferranti, L., Negenborn, R.R., Keviczky, T., and Alonso-Mora, J. (2018). Coordination of multiple vessels via distributed nonlinear model predictive control. In *2018 European Control Conference (ECC)*, 2523–2528. IEEE.

Frew, E.W. and Lawrence, D. (2017). Tracking dynamic star curves using guidance vector fields. *Journal of Guidance, Control, and Dynamics*, 40(6), 1488–1495.

Gonçalves, V.M., Pimenta, L.C., Maia, C.A., Dutra, B.C., and Pereira, G.A. (2010b). Vector fields for robot navigation along time-varying curves in n-dimensions. *IEEE Transactions on Robotics*, 26(4), 647–659.

Gonçalves, V.M., Maia, C.A., Pereira, G.A.S., and Pimenta, L.C.d.A. (2010a). Navegação de robôs utilizando curvas implícitas. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 21, 43 – 57.

Hsieh, M.A., Kumar, V., and Chaimowicz, L. (2008). Decentralized controllers for shape generation with robotic swarms. *Robotica*, 26(5), 691–701.

Jung, W., Lim, S., Lee, D., and Bang, H. (2016). Unmanned aircraft vector field path following with arrival angle control. *Journal of Intelligent & Robotic Systems*, 84(1-4), 311–325.

Maestre, J.M. and Negenborn, R.R. (2014). *Distributed model predictive control made easy*, volume 69. Springer.

Pimenta, L.C., Pereira, G.A., Gonçalves, M.M., Michael, N., Turpin, M., and Kumar, V. (2013). Decentralized controllers for perimeter surveillance with teams of aerial robots. *Advanced Robotics*, 27(9), 697–709.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model predictive control: Theory and design*. Nob Hill Pub.

Rey, F., Pan, Z., Hauswirth, A., and Lygeros, J. (2018). Fully decentralized admm for coordination and collision avoidance. In *2018 European Control Conference (ECC)*, 825–830. doi:10.23919/ECC.2018.8550245.

Van Parys, R. and Pipeleers, G. (2017). Distributed mpc for multi-vehicle systems moving in formation. *Robotics and Autonomous Systems*, 97, 144–152.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25–57.

Wang, Y., Yin, W., and Zeng, J. (2019). Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1), 29–63.