Control Engineering Practice 94 (2020) 104206

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Multi-vehicle coordination based on hierarchical quadratic programming

Héctor M. Pérez-Villeda [a,b], Gustavo Arechavaleta [a,*], América Morales-Díaz [a]

[a] Robotics and Advanced Manufacturing Group, Centro de Investigación y de Estudios Avanzados del IPN, Saltillo, Coah, Mexico
[b] Introid Inc. S.A. de C.V., Ramos Arizpe, Coah, Mexico

ARTICLE INFO

Keywords:
Multi-vehicle systems
Optimization-based control
Hierarchical quadratic programming

ABSTRACT

This paper presents an optimization-based control scheme for generating online multi-vehicle coordination behaviors to accomplish missions in indoor environments. The proposed control scheme relies on the use of hierarchical task functions in terms of the multi-vehicle configuration variables. The task functions are related to individual and group obstacle avoidance, reaching fixed targets, group trajectory tracking, maintaining formations, enclosing the group within a geometric area, among others. The stack of hierarchical tasks automatically handles possible conflicts between them. Quadratic programs are formulated for explicitly solving inequality and equality task constraints at any hierarchy. In addition, a finite state machine is employed to build complex group behaviors for successfully fulfilling group missions. The proposed control scheme is demonstrated on two experiments with static and moving obstacles where a group composed by six vehicles tracks a predefined trajectory for the center of mass of the group. In the second experiment, the group is asked to clean the workspace by pushing movable objects.

1. Introduction

In recent years, the success of multi-vehicle systems (MVS) in industrial facilities has attracted more attention in robotics and control communities (Wurman, D'Andrea, & Mountz, 2008). MVS represents an important category of networked systems with potential applications in industrial warehouse environments for surveillance, inspection, and transportation. The vehicles need to navigate in formations to fulfill missions such as carrying packages, or even cleaning areas by removing objects. In addition, MVS should be able to avoid collisions with both static and moving obstacles while simultaneously achieving their task objectives. Among the important challenges associated to this class of control systems are the coordination (Fanti, Mangini, Pedroncelli, & Ukovich, 2018; Kallem, Komoroski, & Kumar, 2013), communication (Gutiérrez, Morales, & Nijmeijer, 2017; Oh, Park, & Ahn, 2015), scheduling (Reveliotis & Roszkowska, 2011) and obstacle avoidance (Olmi, Secchi, & Fantuzzi, 2011; Trujillo, Becerra, Gómez-Gutiérrez, Ruiz-León, & Ramírez-Treviño, 2018) problems. Due to the underlying complexity of these systems, available control architectures commonly apply a two-stage approach, in which the path planning problem is treated first. Then, a collision-free motion coordination is executed (Alonso-Mora, Baker, & Rus, 2017; Draganjac, Miklić, Kovačić, Vasiljević, & Bogdan, 2016; Krnjak et al., 2015).

Centralized (Li, Liu, Xiao, Yu, & Zhang, 2017; Olmi et al., 2011) and decentralized (Draganjac et al., 2016; Krnjak et al., 2015) path planning algorithms have been suggested for automated guided vehicles (AGV). One of the main purposes has been to overcome collisions when a picking or transportation task is assigned to each vehicle. Another important concern has been to prevent conflicts between two or more vehicles, in particular, deadlock situations (Reveliotis & Roszkowska, 2011). The use of finite state machines has been proposed to overcome collisions as well as deadlocks during execution (Draganjac et al., 2016).

Behavior-based control schemes for MVS have been also suggested (Balch & Arkin, 1998). The behavioral null-space approach is able to control multi-vehicle formation tasks while avoiding self-collisions (Antonelli, Arrichiello, & Chiaverini, 2009). Also, it has been extended to accomplish task objectives in a predefined time regardless of the initial state of the MVS (Arechavaleta, Morales-Díaz, Pérez-Villeda, & Castelán, 2017). In Gracia et al. (2018), the null-space approach has been applied to handle conflicts between task objectives while performing cooperative object transportation maneuvers guided by a human operator.

The proposed optimization-based control scheme solves behavioral task objectives for one or multiple vehicles. However, the conflict resolution method differs from both zone occupancy strategies (Fanti et al., 2018; Krnjak et al., 2015) and the null-space approach. In particular, we adopt the hierarchical quadratic programming (HQP) framework to integrate the path planning and motion coordination problems, as it has been demonstrated in Escande, Mansard, and Wieber (2014), Herzog et al. (2016) and Kanoun, Lamiraux, and Wieber (2011), for humanoid robots. The HQP handles strict hierarchical tasks composed by linear

* Corresponding author.
E-mail addresses: hector@introid.com (H.M. Pérez-Villeda), garechav@cinvestav.mx (G. Arechavaleta), america.morales@cinvestav.mx (A. Morales-Díaz).

https://doi.org/10.1016/j.conengprac.2019.104206
Received 18 January 2019; Received in revised form 17 October 2019; Accepted 19 October 2019
Available online 1 November 2019
0967-0661/

equality and inequality constraints (Kanoun et al., 2011). For designing complex missions, such as cooperative object transportation, a finite state machine is employed where each state represents a behavior of the MVS. Each behavior or state is composed by a stack of task objectives. The precise definition of tasks is given in Section 2. They are mainly designed to avoid collisions, maintain formations, track moving references and reach targets. The communication problem is treated here in a centralized manner, however, hierarchical tasks can also be defined in the context of decentralized schemes through consensus as it is explored in Trujillo et al. (2018).

1.1. Problem formulation

The aim of this work is to solve an online constrained optimization problem for driving the team of vehicles to destinations, or goal regions in the environment populated with moving and static obstacles, while keeping proximity constraints and avoiding collisions. The coordination tasks assigned to the group account for trajectory tracking as well as surrounding movable objects to transport them by multiple vehicles.

Let us consider a group of n vehicles as:

$$\mathcal{G} = \{\boldsymbol{q}_i \mid i = 1, \dots, n\} \tag{1}$$

where $\boldsymbol{q}_i = [x_i, y_i]^T \in \mathbb{R}^2$ is the configuration of i-vehicle with its first-order dynamics:

$$\dot{\boldsymbol{q}}_i = \boldsymbol{u}_i, \qquad \boldsymbol{u}_i \in \mathcal{U} \subseteq \mathbb{R}^2, \qquad i = 1, \dots, n \tag{2}$$

In general, the problem consists of minimizing an objective function $f(\boldsymbol{u}) : \mathbb{R}^{2n} \to \mathbb{R}$ subject to equality and inequality constraints, $\boldsymbol{g}_j(\boldsymbol{q}, \boldsymbol{u}) : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \to \mathbb{R}$, $j \in \{1, \dots, r\}$, and $\boldsymbol{h}_k(\boldsymbol{q}, \boldsymbol{u}) : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \to \mathbb{R}$, $k \in \{1, \dots, s\}$, respectively. The functions f, $\boldsymbol{g}_j$, and $\boldsymbol{h}_k$ are continuously differentiable, and they depend on the state, $\boldsymbol{q} = [\boldsymbol{q}_1^T \ \dots \ \boldsymbol{q}_n^T]^T \in \mathbb{R}^{2n}$, and decision variables $\boldsymbol{u} = [\boldsymbol{u}_1^T \ \dots \ \boldsymbol{u}_n^T]^T \in \mathbb{R}^{2n}$.

The constrained optimization problem is formulated as:

$$\begin{aligned} &\underset{\boldsymbol{u}}{\text{minimize}} && f(\boldsymbol{u}) \\ &\text{subject to} && \begin{cases} \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{u}) = 0 \\ \boldsymbol{h}(\boldsymbol{q}, \boldsymbol{u}) \geq 0 \\ \boldsymbol{u}_{min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{max} \end{cases} \end{aligned} \tag{3}$$

The resulting behavior of the group, induced by the optimal decision variable $\boldsymbol{u}^*$, should guarantee convergence of task errors while handling conflicts among task constraints.

1.2. Related work

Among the vast research efforts for increasing the degree of autonomy of MVS, optimization-based control techniques have been successfully applied to steer MVS in complex situations. In Ayanian and Kumar (2010), decentralized feedback controllers are synthesizes for multi-agent teams subject to obstacle avoidance constraints where a linear program is solved.

Tractable and scalable multi-agent control methods have been explored for large-scale networked agents (Derenick & Spletzer, 2007; Rudd, Foderaro, Zhu, & Ferrari, 2017). In particular, numerical optimal control and convex optimization techniques have been satisfactory applied. In Rudd et al. (2017), a distributed optimal control problem is studied to fulfill multiple cooperative objectives for many agents subject to obstacles and wind currents. The optimality conditions are used to derive an indirect method known as generalized reduced gradient. It turns to be computationally more efficient than direct methods based on sequential quadratic programming. On the other hand, a variety of convex optimization methods have been explored (Alonso-Mora et al., 2017; Derenick & Spletzer, 2007; Mousavi, Moshiri, & Heshmati, 2015). In Derenick and Spletzer (2007) the optimal formation problem for MVS is formulated with shape analysis techniques and second-order cone programming. In order to be implemented in real-time, the method relies on the solution of convex quadratic programs. More recently, Alonso-Mora et al. (2017) proposed to approximate convex regions with semi-definite programming to capture the collision-free workspace. Within each convex region, the group formation remains fixed, and a sequential convex program solves the multi-robot formation control in environments populated with moving obstacles. It is known that model-predictive control is able to greatly reduce the computation of trajectory planning by means of prediction phases with a time horizon (Mousavi et al., 2015). In that manner, a less computationally demanding convex optimization problem is solved in real-time for performing MVS tasks.

This work adopts HQP method, which is a rather different convex optimization method to compute the instantaneous optimal control for driving the MVS toward the group targets. It has demonstrated good performance in the context of humanoid robotics where the feasibility of task objectives and inequality constraints follows a strict hierarchy (Escande et al., 2014; Herzog et al., 2016; Kanoun et al., 2011). In MVS, many coordination constraints are naturally defined as inequalities, such as obstacle avoidance and occupancy regions. Different from the null-space approach (Antonelli & Chiaverini, 2006), it is shown how HQP permits to introduce inequality constraints at any hierarchical level. For extending the HQP to perform complex MVS missions, a finite state machine is defined where each state solves an instance of HQP with a given hierarchical structure, which is also known as stack of tasks.

The remaining sections of the paper are organized as follows. In Section 2, the formulation and taxonomy of MVS tasks is detailed. In Section 3 a set of experiments with real MVS is carefully detailed to verify the effectiveness of the proposed control scheme. Finally, some concluding remarks and future directions are given in Section 4.

2. Formulation of multi-vehicle tasks

The definition of multi-vehicle coordination tasks follows the task function approach (Samson, Borgne, & Espiau, 1991), which has been introduced in the context of motion control for kinematically redundant robotic manipulators. A multi-vehicle task function induces a motion behavior for the group of vehicles. Each task involves the motion coordinates of the whole group, or it could act on at least one vehicle belonging to the group. It is referred to *global* or *local* task whether it depends on more than one vehicle or just one of them, respectively. From the constrained optimization perspective, a given task could be naturally defined in terms of either equalities or inequalities. Thus, the definition of task functions is sufficiently general to deal with one or more agents within the group as well as to assign feasible regions with mixed constraints, i.e. *equality* and *inequality tasks*, as it is illustrated in Fig. 1.

For clarity purposes, Fig. 2 shows the properties of every single coordination task that is described in the next sections. As it is observed, obstacle avoidance tasks are composed by inequalities, and they could be local or global. A reaching task could also be global or local, but it is described as the convergence of an error function involving the current and target values of the task.

2.1. Geometric formation

The task function to maintain a circular formation with the vehicles is an example of a geometric formation task. In this particular case, the task limits the feasible displacements of each agent over the circumference. This is a local task function because each vehicle reaches the perimeter of a given circumference without the need to know the location of other team members. The error function of the task is defined as

$$e_{c_i} = \frac{1}{2}(\boldsymbol{q}_i - \boldsymbol{p}_c)^T(\boldsymbol{q}_i - \boldsymbol{p}_c) - \frac{r^2}{2} \in \mathbb{R} \tag{4}$$

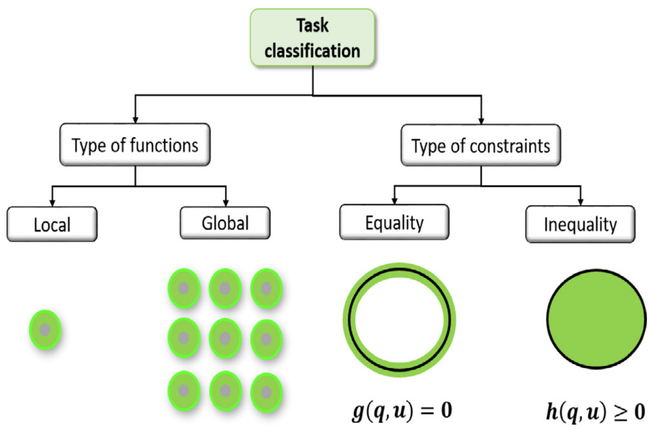

Fig. 1. Task classification: the scope of coordination tasks could be global or local. The first type of task functions involves more than one vehicle while the second type deals with only one. In addition, a task is composed by a set of either equality or inequality constraints depending on the nature of such task.

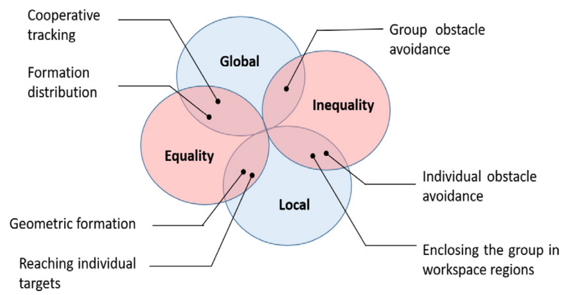

Fig. 2. Task properties: obstacle avoidance tasks are naturally defined as inequalities since there is a forbidden area where the vehicles cannot penetrate. In contrast, the equality tasks can be defined to reach a target by one or more vehicles. The constraints related to maintain geometric formations are also designed as equality tasks.

where r is the radius and $\boldsymbol{p}_c = [x_c \ y_c]^T \in \mathbb{R}^2$ the center of the circle. The stack of circular tasks of the form (4) is $\boldsymbol{e}_c = [e_{c_1} \ e_{c_2} \ \cdots \ e_{c_n}]^T \in \mathbb{R}^n$, and its time-derivative is $\dot{\boldsymbol{e}}_c = \boldsymbol{J}_c \boldsymbol{u}$ where the task Jacobian $\boldsymbol{J}_c = \frac{\partial e_c}{\partial q} \in \mathbb{R}^{n \times 2n}$ is of the form:

$$\boldsymbol{J}_c = \text{block diag}\left[(\boldsymbol{q}_1 - \boldsymbol{p}_c)^T \cdots (\boldsymbol{q}_n - \boldsymbol{p}_c)^T\right] \tag{5}$$

The task is formulated as a set of equality constraints since any vehicle's location different from the circular boundary is forbidden. The linear system to be satisfied at each instant of time is:

$$\boldsymbol{J}_c \boldsymbol{u} = -\alpha_c \boldsymbol{e}_c \tag{6}$$

Note that an exponential error decrease is imposed (i.e. $\dot{\boldsymbol{e}}_c = -\alpha_c \boldsymbol{e}_c$) where $\alpha_c \in \mathbb{R}^+$ is a small positive constant.

2.2. Enclosing the group in workspace regions

Unlike the task that makes the vehicles to reach the boundary of a geometric shape, as described in Section 2.1, the enclosing task function limits the navigation of the agents inside the area defined by a given geometric shape. Any location outside that region is not allowed. The error function associated to this task is defined as $\boldsymbol{e}_e \in \mathbb{R}^n$ with its task Jacobian $\boldsymbol{J}_e \in \mathbb{R}^{n \times 2n}$. If the allowable navigation area represents the interior of a circle, then the structure of the task error and its Jacobian are the same as in (4) and (5), respectively. The only difference relies on the type of constraints that compose the enclosing task:

$$\boldsymbol{J}_e \boldsymbol{u} \leq -\alpha_e \boldsymbol{e}_e \tag{7}$$

where $\alpha_e \in \mathbb{R}^+$ is a small positive constant

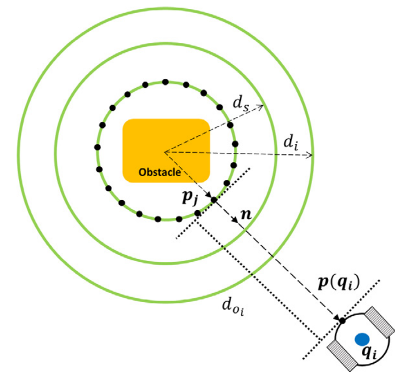

Fig. 3. Individual obstacle avoidance task: the yellow area represents the obstacle. The inner circle encapsulates the geometry of the obstacle. The circle at the middle represents the security distance with radius d_s. The outer circle is defined by the radius of influence d_i with respect to the vehicles location q_i. The normal vector $\boldsymbol{n}$ points to the nearest distance between the obstacle and the vehicle.

2.3. Individual obstacle avoidance task

A critical task function required for safety vehicle navigation is obstacle avoidance. This task is formulated as local to overcome potential collisions between a given vehicle with the environment. Note that the other agents as well as moving and static obstacles belong to the same workspace. The obstacle avoidance task only depends on the vehicle and the nearest obstacle. Thus, there is no need for the remaining agents to share their locations. Fig. 3 illustrates the *velocity damper* introduced in Faverjon and Tournassoud (1987), which defines a set of inequality task constraints for the vehicle to avoid near obstacles.

Security and influence distances around the obstacle are defined as d_s and d_i, respectively. The inner circle is represented by a sequence of points $\boldsymbol{p}_j$ uniformly distributed over the circumference. The point $\boldsymbol{p}(\boldsymbol{q}_i)$ is over the perimeter of i-vehicle such that the nearest distance between the obstacle and i-vehicle is $d_{o_i} = ||\boldsymbol{p}(\boldsymbol{q}_i) - \boldsymbol{p}_j||$. The error function for the obstacle avoidance task of i-vehicle is formulated as

$$e_{o_i} = d_{o_i} - d_s \in \mathbb{R} \tag{8}$$

and its time-derivative is

$$\dot{e}_{o_i} = \boldsymbol{n}^T \dot{\boldsymbol{p}}(\boldsymbol{q}_i) \tag{9}$$

where $\boldsymbol{n}$ is a unit vector defined as

$$\boldsymbol{n} = \frac{\boldsymbol{p}(\boldsymbol{q}_i) - \boldsymbol{p}_j}{d_{o_i}} \quad \text{and} \quad \dot{\boldsymbol{p}}(\boldsymbol{q}_i) = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}_i} \dot{\boldsymbol{q}}_i$$

Therefore, the task Jacobian becomes:

$$\boldsymbol{J}_{o_i} = \boldsymbol{n}^T \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}_i} \in \mathbb{R}^{1 \times 2} \tag{10}$$

From (8) and (10), the inequality constraint defining the individual obstacle avoidance task is:

$$\boldsymbol{J}_{o_i} \boldsymbol{u}_i \geq -\alpha_{o_i} e_{o_i} \tag{11}$$

where $\alpha_{o_i} = \frac{\xi_{o_i}}{d_i - d_s} \in \mathbb{R}$, $d_i > d_s$ and ξ_{o_i} is a small positive constant that regulates the convergence speed. For continuously differentiable d_{o_i}, it is proved in Kanehiro, Lamiraux, Kanoun, Yoshida, and Laumond (2008) that the minimum distance between the i-vehicle and the obstacle constrained by the *velocity damper* never be smaller than d_s.

Notice that when this task is combined with others, such as reaching a target location for instance, the vehicle keeps its motion along the tangent direction with respect to the obstacle's inner circle until the potential collision is avoided.

It is worth highlighting when two vehicles are facing each other. If this occurs, the agents consider each other as moving obstacles,

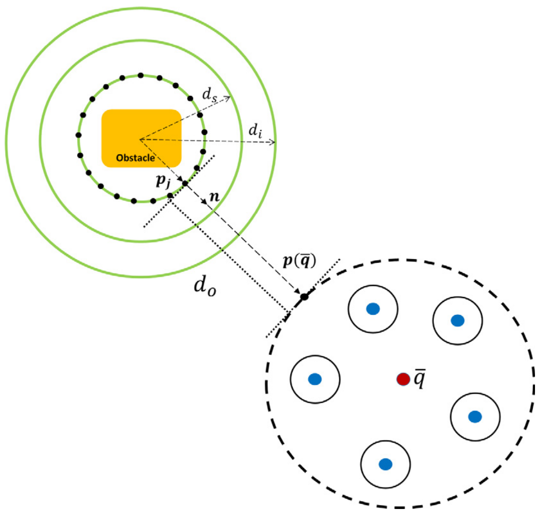

Fig. 4. Group obstacle avoidance: the centroid of the team formation is used as the control point to measure the distance to the nearest obstacle in yellow. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which means that each of them performs its own individual obstacle avoidance task. Since there is not an error function dependency, each task remains local.

2.4. Group obstacle avoidance

Here, the individual obstacle avoidance task is extended to deal with a group of vehicles. This task turns to be useful when the corresponding set of agents is transporting a movable object without colliding with the environment, i.e. the team formation to enclose the movable object is maintained. The scope of the task is global since the position coordinates of the group of vehicles are involved. The difference with respect to the individual obstacle avoidance task is regarding the control point to measure the distance to the nearest obstacle. Instead of using the centroid of the vehicle as the control point, it is used the centroid of the formation, as it is shown in Fig. 4.

Since the group obstacle avoidance task needs all the robots involved in the formation, the task error in this case has the form

$$e_o = d_o - d_s \in \mathbb{R} \tag{12}$$

where $d_o = ||\boldsymbol{p}(\overline{\boldsymbol{q}}) - \boldsymbol{p}_j||$, $\boldsymbol{p}(\overline{\boldsymbol{q}})$ is a point over the perimeter of a virtual circle containing the group of vehicles as it is illustrated in Fig. 4. The centroid of the formation is $\overline{\boldsymbol{q}} = [\overline{x} \quad \overline{y}]^T \in \mathbb{R}^2$ where

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and} \quad \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

and the corresponding task Jacobian becomes

$$\boldsymbol{J}_o = \frac{1}{n}\left[\boldsymbol{n}^T \frac{\partial p}{\partial \overline{q}} \frac{\partial \overline{q}}{\partial q_1} \quad \cdots \quad \boldsymbol{n}^T \frac{\partial p}{\partial \overline{q}} \frac{\partial \overline{q}}{\partial q_n}\right] \in \mathbb{R}^{1\times 2n} \tag{13}$$

Notice that $\frac{1}{n}$ represents the individual contribution of each robot to accomplish the task. Therefore, the inequality constraints are defined as

$$\boldsymbol{J}_o \boldsymbol{u} \geq -\alpha_o e_o \tag{14}$$

where $\alpha_o = \frac{\xi_o}{d_i - d_s} \in \mathbb{R}$, $d_i > d_s$ and ξ_o is a small positive constant that regulates the convergence speed.

2.5. Formation distribution

The vehicles can follow a desired variance $[\sigma_x, \sigma_y]^T$ to achieve a formation with the following task function

$$\boldsymbol{e}_\sigma = \frac{1}{n}\sum_{i=1}^{n}\begin{bmatrix}(x_i - \overline{x})^2 \\ (y_i - \overline{y})^2\end{bmatrix} - \begin{bmatrix}\sigma_x \\ \sigma_y\end{bmatrix} \in \mathbb{R}^2 \tag{15}$$

The task is global due to the average position $\overline{x}$ and $\overline{y}$ of the formation. The Jacobian is given by

$$\boldsymbol{J}_\sigma = \frac{2(n-1)}{n^2}\left[\boldsymbol{J}_{\sigma_1} \quad \cdots \quad \boldsymbol{J}_{\sigma_n}\right] \in \mathbb{R}^{2\times 2n} \tag{16}$$

where

$$\boldsymbol{J}_{\sigma_i} = \begin{bmatrix} x_i - \overline{x} & 0 \\ 0 & y_i - \overline{y}\end{bmatrix} \in \mathbb{R}^{2\times 2} \tag{17}$$

The team of vehicles are then spatially distributed by means of the following equality constraint

$$\boldsymbol{J}_\sigma \boldsymbol{u} = -\alpha_\sigma \boldsymbol{e}_\sigma \tag{18}$$

where $\alpha_\sigma \in \mathbb{R}^+$ is a small positive constant.

2.6. Reaching a target

This is a local task for which the individual error function is

$$\boldsymbol{e}_{g_i} = \boldsymbol{q}_i - \boldsymbol{q}_{d_i} \in \mathbb{R}^2 \tag{19}$$

where the target $\boldsymbol{q}_{d_i}$ in regulation regime is constant, otherwise it is a time-varying function. The corresponding equality task for each vehicle becomes

$$\boldsymbol{J}_{g_i} \boldsymbol{u}_i = -\alpha_{g_i} \boldsymbol{e}_{g_i} + \dot{\boldsymbol{q}}_{d_i} \tag{20}$$

where $\alpha_{g_i} \in \mathbb{R}^+$ is a small positive constant.

2.7. Cooperative tracking

This is a global task designed to reach a target by the group of vehicles in a coordinated way while keeping the desired formation. It consists of regulating the centroid of the formation toward the target

$$\boldsymbol{e}_\mu = \overline{\boldsymbol{q}} - \boldsymbol{q}_d \in \mathbb{R}^2 \tag{21}$$

It is straightforward to adapt the task error for tracking purposes when the target is a time-varying function:

$$\dot{\boldsymbol{e}}_\mu = \boldsymbol{J}_\mu \boldsymbol{u} - \dot{\boldsymbol{q}}_d \tag{22}$$

where the task Jacobian is given by

$$\boldsymbol{J}_\mu = \frac{1}{n}\left[\boldsymbol{J}_{\mu_1} \quad \cdots \quad \boldsymbol{J}_{\mu_n}\right] \in \mathbb{R}^{2\times 2n} \tag{23}$$

with

$$\boldsymbol{J}_{\mu_i} = \begin{bmatrix} 1 & 0 \\ 0 & 1\end{bmatrix} \in \mathbb{R}^{2\times 2} \tag{24}$$

An exponential convergence of the task error is imposed

$$\boldsymbol{J}_\mu \boldsymbol{u} = -\alpha_\mu \boldsymbol{e}_\mu + \dot{\boldsymbol{q}}_d \tag{25}$$

where $\alpha_\mu \in \mathbb{R}^+$ is a small positive constant.

3. Experimental results

This section is devoted to evaluate the proposed MVS control scheme. In particular, the validation has been performed by means of two experiments with six vehicles (see Fig. 5). The first one illustrates the trajectory tracking behavior of the group while maintaining the formation and avoiding static obstacles in the environment. The second experiment demonstrates a more complex group behavior in which the vehicles are asked to cooperate for cleaning the workspace by pushing different objects. The attached multimedia material contains the recorded execution of both experiments.

The experimental setup is conformed by six differential drive mobile platforms in an indoor environment. An optical motion capture system composed by twelve fixed cameras is used to instantaneously obtain, at 120 fps, the vehicle's position as well as the placement of movable objects, static and mobile obstacles in the workspace. The control law is

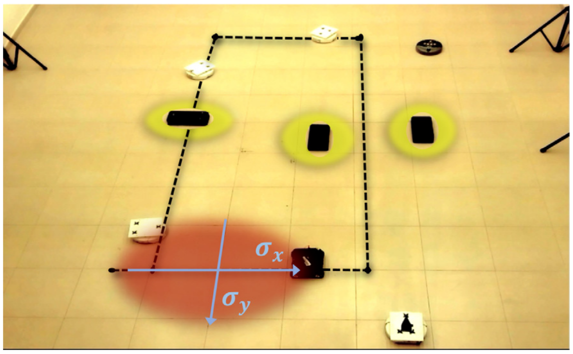

Fig. 5. The scenario for experiment 1. The centroid of the group tracks a trajectory (dotted black) while maintaining the formation (red circle) and avoiding obstacles (yellow regions). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

computed in an external laptop with conventional resources. The Robot Operating System (ROS) Indigo (Quigley et al., 2009), and Bluetooth protocol serve to communicate with the networked system. All the routines have been written in C++. For solving quadratic programs it is used qpOASES (Ferreau, Kirches, Potschka, Bock, & Diehl, 2014) with a damping factor of 0.5. The optimal control signals obtained by the HQP serve as reference velocities to be tracked by the PID controller of vehicle's wheels. The control cycle of the MVS was 50 ms. It considers the computation of task Jacobians, the HQP solver and the communication through bluetooth protocol.

3.1. Computing the angular velocity reference

Since the vehicles are subject to nonholonomic constraints, it is applied the classic input–output linearization to get the reference angular velocity of each vehicle (Oriolo, De Luca, & Venditteli, 2002). The differential kinematics of each vehicle is of the form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \nu \tag{26}$$

where $\nu = [v \ \omega]^T \in \mathbb{R}^2$ contains the speed and angular velocity of the vehicle, respectively. The pair $[x \ y]^T \in \mathbb{R}^2$ stands for its position that corresponds to the midpoint of the wheel's main axis, and $\theta \in \mathbb{S}^1$ represents its orientation with respect to the horizontal axis of the world's reference frame. In particular, the coordinates of a reference point located in front of the vehicle are selected:

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x + L\cos\theta \\ y + L\sin\theta \end{bmatrix} \tag{27}$$

where L is the length. Using (26), the time-derivative of the new reference point is expressed as

$$\dot{z} = B(\theta)\nu \tag{28}$$

where

$$B(\theta) = \begin{bmatrix} \cos\theta & -L\sin\theta \\ \sin\theta & L\cos\theta \end{bmatrix}$$

and ν is the control input for the input–output linearized model. The heading controller is obtained from (28):

$$\nu = B^{-1}(\theta) \begin{bmatrix} -k_1 (x + L\cos\theta) \\ -k_2 (y + L\sin\theta) \end{bmatrix} \tag{29}$$

where k_1 and k_2 are positive constant gains. For the experiments, $k_1 = k_2 = 0.1$. The reference angular velocity ω is then extracted from ν in (29) for each vehicle.

Table 1
Hierarchical tasks of experiment 1.

Task	Hierarchy	Type of function	Type of constraint
e_{o_i}	1	Local	Inequality
$\boldsymbol{e}_e$	2	Local	Inequality
$\boldsymbol{e}_\sigma$	3	Global	Equality
$\boldsymbol{e}_\mu$	4	Global	Equality

Table 2
Task parameters of experiment 1.

Task	Parameters
e_{o_i}	$d_i = 0.5$ m, $d_s = 0.3$ m, $\alpha_{o_i} = 0.08$
$\boldsymbol{e}_e$	$r = 0.8$ m, $\alpha_e = 2.5$
$\boldsymbol{e}_\sigma$	$\sigma_x = 0.3$ m, $\sigma_y = 0.5$ m $\alpha_\sigma = 2.5$
$\boldsymbol{e}_\mu$	$\alpha_\mu = 4$

3.2. Trajectory tracking behavior of the multi-vehicle system

In this experiment, six vehicles are asked to track a reference trajectory while maintaining the formation and avoiding static obstacles. In addition, the vehicles are constrained to move inside a circular area where its center is a time-varying point. The scenario is shown in Fig. 5. The hierarchical task functions follow the order $e_{o_i} \succ \boldsymbol{e}_e \succ \boldsymbol{e}_\sigma \succ \boldsymbol{e}_\mu$. Table 1 summarizes the taxonomy of task functions involved for this experiment.

As it is depicted in Fig. 5, the black boxes inside the yellow area represent fixed obstacles. Each vehicle is considered itself as a mobile obstacle. Thus, the first hierarchical task is e_{o_i}, i.e., the individual obstacle avoidance task. The geometric formation task $\boldsymbol{e}_e$ belongs to the second hierarchical level. It is illustrated by the red circle in Fig. 5. This implies that the agents are not allowed to move outside the red circle. The desired distribution of vehicles inside the red circle corresponds to the third hierarchical level. It is performed by the variance task function $\boldsymbol{e}_\sigma$. In Fig. 5, the desired distribution is expressed in terms of x and y axes with blue arrows. Note that the variance depends on the position of every agent. At the lowest hierarchical level the group trajectory tracking $\boldsymbol{e}_\mu$ is solved. The reference trajectory is represented by a black dotted rectangular path parametrized with respect to time. The tracking error is measured with the centroid of the group formation $\bar{\boldsymbol{q}}$.

Fig. 6 illustrates the desired effect of the group behavior when each task comes to play. The individual obstacle avoidance task e_{o_i} makes the agents spread out each other.

At the second hierarchical level, the task $\boldsymbol{e}_e$ attracts and keeps the vehicles inside the geometric formation (see Fig. 6(b)).

At the third hierarchical level, the task $\boldsymbol{e}_\sigma$ distributes the agents within the allowable area by imposing a desired variance (see Fig. 6(c)). Finally, the task with the lowest hierarchy $\boldsymbol{e}_\mu$ is in charge of tracking the desired trajectory by controlling the center of mass of the group (see Fig. 6(d)).

It is worth to mention that the formation could be broken within a time interval if any vehicle needs to perform an obstacle avoidance. This could happen because e_{o_i} has the highest hierarchy. The parameters to execute the task functions for this experiment are given in Table 2.

Figs. 7 and 8 show the evolution of task errors and control profiles, respectively.

During the MVS execution, the first obstacle avoidance occurred at $t = 53$ s as depicted in Fig. 9. Although the task errors in Fig. 7 increased due to the obstacle avoidance, the formation is recovered. Also, within the time interval (91.2 s, 93.2 s), vehicle 6 suffered an external disturbance represented by $* \longrightarrow \times$ in Fig. 9. The vehicles 1, 2 and 4 reacted by trying to compensate the desired distribution. Regardless of the undesired effect of obstructions and perturbations, the task errors converged to zero as it is observed in Fig. 7.

Note that the first obstacle avoidance occurred at $t = 53$ s as depicted in Fig. 9. Although the task errors in Fig. 7 increased due

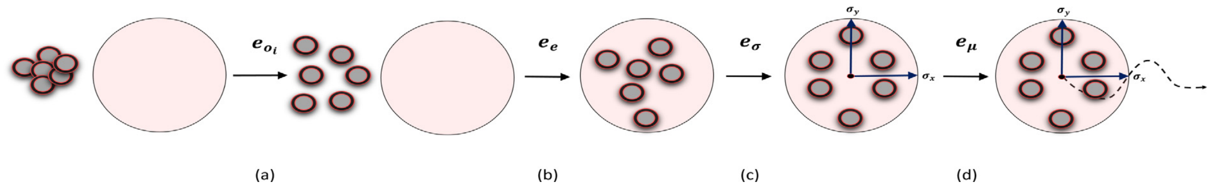

Fig. 6. Group behavior with hierarchical task functions. Each task induces specific group behaviors. It is illustrated, from left to right, the individual obstacle avoidance, enclosing the group, formation distribution, and cooperative tracking tasks, respectively.

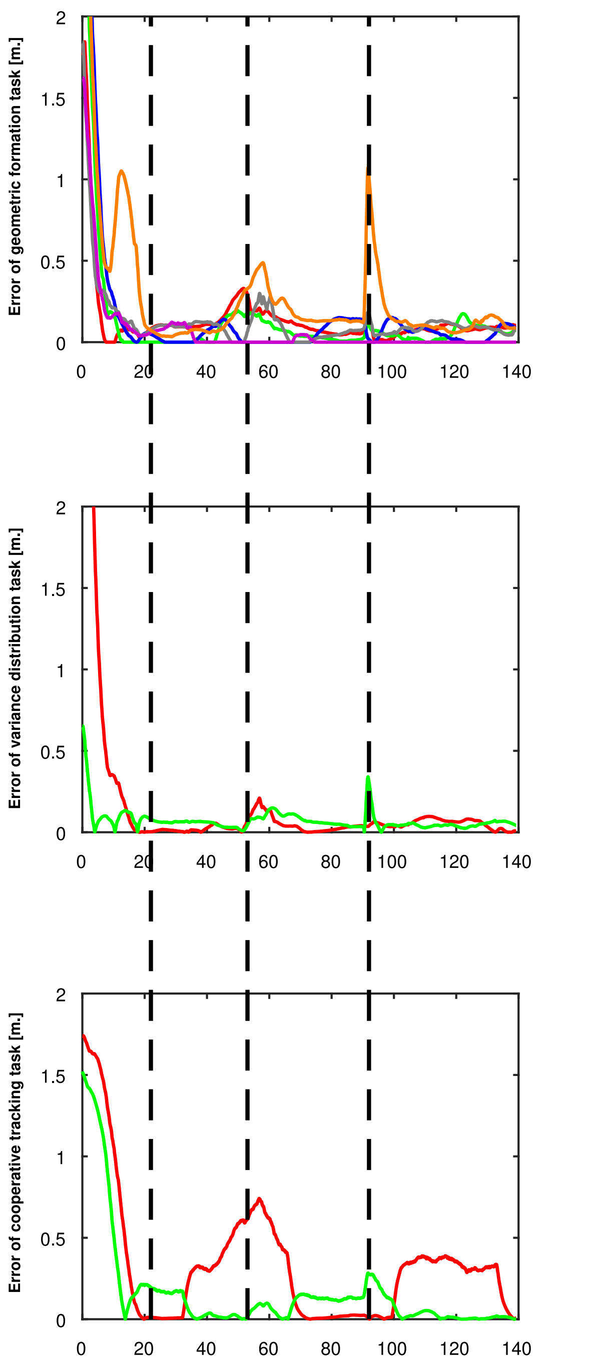

Fig. 7. The task error profiles for experiment 1. Task errors are consistent with the imposed hierarchical structure. Thus, only when obstacles and the external perturbation appeared, the errors increased.

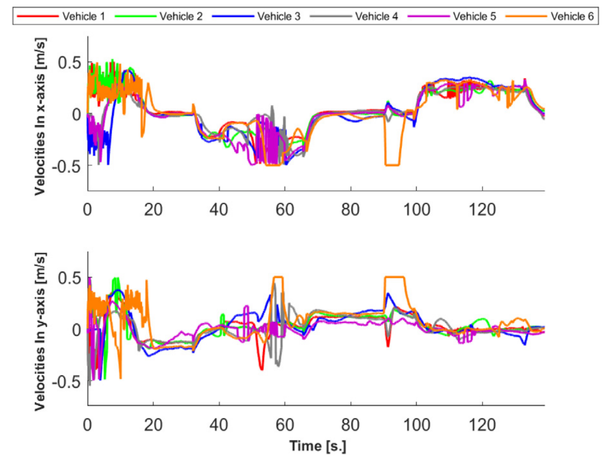

Fig. 8. Control profiles for experiment 1. The top-row shows the control profiles corresponding to six vehicles along the x-axis. Bottom-row depicts the control profiles along y-axis. The velocity limits are ± 0.5 m/s.

to the obstacle avoidance, the formation is recovered. Also, within the time interval $(91.2\ \mathrm{s}, 93.2\ \mathrm{s})$, vehicle 6 suffered an external disturbance represented by $* \longrightarrow \times$ in Fig. 9. The vehicles 1, 2 and 4 reacted by trying to compensate the desired distribution. Regardless of the undesired effect of obstructions and perturbations, the task errors converged to zero as it is observed in Fig. 7 and the control profiles remain below the boundaries defined by the velocity limits of the vehicles at ± 0.5 m/s as depicted in Fig. 8.

3.3. Cleaning the workspace by pushing

The multi-vehicle behaviors related to the second experiment highlight the flexibility and scalability of the proposed optimization-based control scheme. In this case, the group composed by five vehicles was asked to transport three movable objects located within the workspace. The sixth vehicle played the roll of an moving obstacle. Fig. 10 shows the scenario. The mission for the group was to clean the workspace by pushing movable objects. The mission needed individual and group obstacle avoidance. Also, once the group finished to relocate the objects, each vehicle reached a target location represented by red circle in Fig. 10. A finite state machine was employed to represent the mission. Each state was composed by a stack of hierarchical tasks while discrete events triggered state transitions. The finite state machine is illustrated in Fig. 11. In particular, the hierarchical structures $e_{o_i} \succ e_c \succ e_{g_i}$, $e_o \succ e_c \succ e_{g_i} \succ e_\mu$ and $e_{o_i} \succ e_{g_i}$ correspond to *States* 1, 2 and 3, respectively. It is important to note that the task function e_g is global since it regulates the centroid of the group $\overline{q}$ toward a desired position. Note that the task parameters to perform this experiment are given in Table 3.

The scenario of experiment 2 is shown in Fig. 12(a). The first 76 s of the mission execution correspond to Fig. 12(b). It is observed how the vehicles reached the desired geometric formation, and the diameter of the circumference d_1 shrank gradually until the first movable object was wrapped by the group. The shrinking process was triggered when

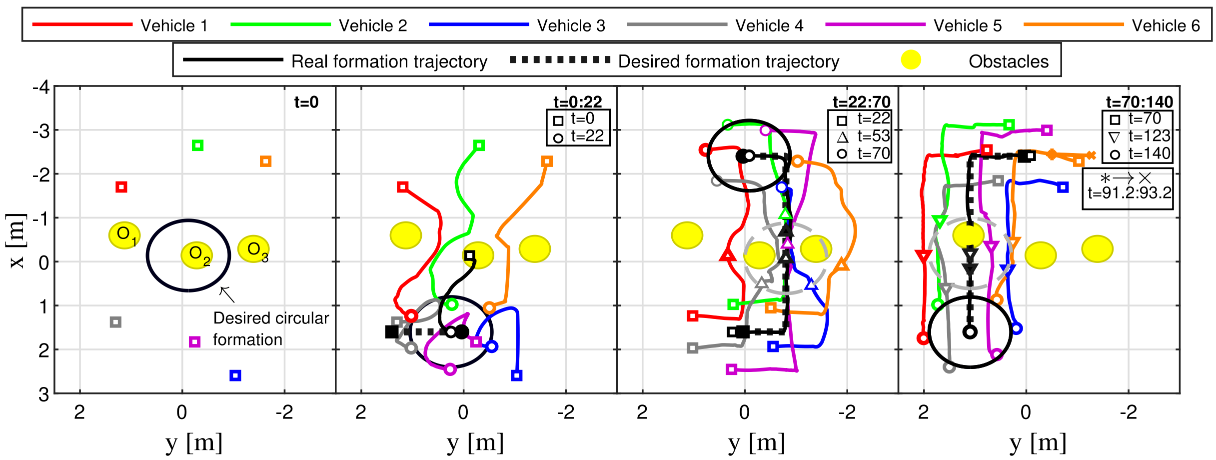

Fig. 9. Execution of group behaviors for experiment 1. The squares and circles represent the initial and final position of each vehicle. Dotted gray circles represent the desired formation. Yellow circles O_1 O_2 and O_3 represent three static obstacles. The task functions are performed with the following hierarchical levels $e_{o_i} \succ e_c \succ e_\sigma \succ e_\mu$. Individual obstacle avoidance happened at 53 and 123 s. Between 91 and 93 s an external perturbation is induced. In particular, a manual displacement of vehicle 6 was performed.

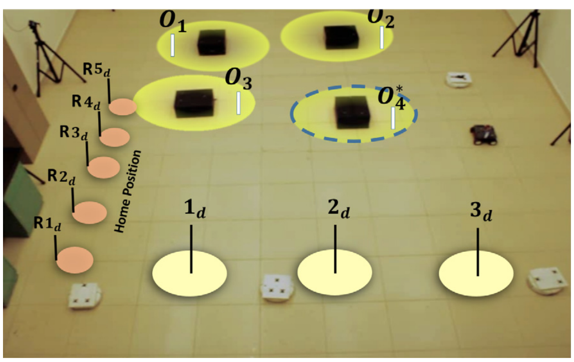

Fig. 10. The scenario for experiment 2. The MVS is asked to clean the workspace by pushing three movable objects represented by yellow circles (1, 2, 3). The fourth object enclosed by blue dotted contour represents a static obstacle. The light yellow circles represent the desired placements where the movable objects have to be relocated. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

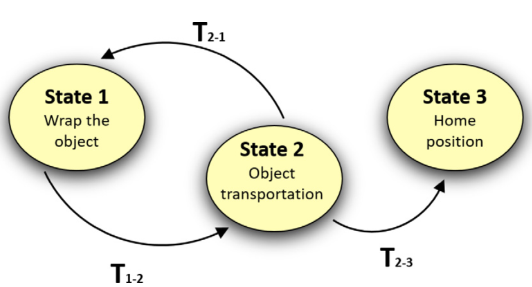

Fig. 11. Finite state machine of experiment 2. Once the group wrapped the object in *State 1*, it switched to *State 2* by T_{1-2} to transport the object toward the goal location. Then, the group switched to *State 1* by T_{2-1} to repeat the task with the second movable object. Once the workspace is cleaned, the vehicles switched to *State 3* by T_{2-3}.

the error between the centroid of group $\bar{q}$ and the object position O_1 was small enough, i.e., $\|\bar{q} - \mathrm{O}_1\| < \epsilon$, where $\epsilon = 0.04$ m in this experiment. The diameter d_1 shrank until it reached a desired value d_2. The ×-markers represent the position of the vehicles when d_1 reached d_2. Then, the state transition was activated T_{1-2}.

Fig. 12(c) shows the execution of experiment 2 from 76 to 130 s. A critical group behavior within this time interval was to avoid obstacles while transporting the movable object. For this, the group obstacle avoidance was performed while the centroid of the formation

Table 3
Parameters of experiment 2.

Tasks	Parameters	
e_{o_i}	$d_i = 0.5\,\mathrm{m}$, $d_s = 0.7\,\mathrm{m}$, $\alpha_{o_i} = 0.003$	
e_o	$d_i = 0.5\,\mathrm{m}$, $d_s = 0.7\,\mathrm{m}$, $\alpha_o = 0.05$	
e_c	$\triangle \longrightarrow \square$	$r = 1.2$ m; $\alpha_c = 0.1$
	$\square \longrightarrow +$	$r = 1.2 : 0.52$ m; $\alpha_c = 1.0$
e_{g_i}	$\triangle \longrightarrow \square$	$\alpha_{g_i} = 0.1$
	$\square \longrightarrow +$	$\alpha_{g_i} = 1.0$
	$\triangle \longrightarrow \circ$	$\alpha_{g_i} = 0.7$
e_μ	$\alpha_\mu = 1.2$	

was regulated toward the desired location, i.e. the movable object O_1 was relocated to O_{1_d}. Then, the group switched to *State 1* for wrapping the second movable object O_2 in a cooperative manner. The event that activated the state transition T_{2-1} was the error condition $\left\|\mathrm{O}_1 - \mathrm{O}_{1_d}\right\| < \epsilon$. In Fig. 12(d) is depicted the next time interval of the execution that goes from 130 to 229 s. It is observer that arbitrary displacements of O_3 and O_4 were performed by a human to illustrate that the proposed optimization-based control is able to deal with such modifications without altering the purpose of the mission, i.e., wrapping O_2 to be transported to O_{2_d}.

In Fig. 12(e) the group had to avoid a moving obstacle without breaking the formation at 261 s. The trajectory of the moving obstacle is represented by $* \longrightarrow \Diamond$, which caused a decrease of the group velocity to avoid collisions. Then, the group switched again to *State 1* for wrapping O_3. The time window from 291 to 396 s is depicted in Fig. 12(f). In particular, it can be observed how vehicle 3 had to avoid potential collisions against vehicles 1 and 2. For this, vehicle 3 momentary violated the geometric formation by passing thought the interior of it. Such behavior was possible due to the hierarchical structure together with the slack variables, i.e. individual obstacle avoidance was at the highest hierarchy.

Fig. 12(g) depicts the execution from 396 to 454 s. The important feature here was an arbitrary displacement of O_3 while the group was transporting it. That event activated T_{2-1} for wrapping again O_3 at its new location. In Fig. 12(h) is shown the last part of the experiment that covered the time window from 454 to 613 s. It is observed that

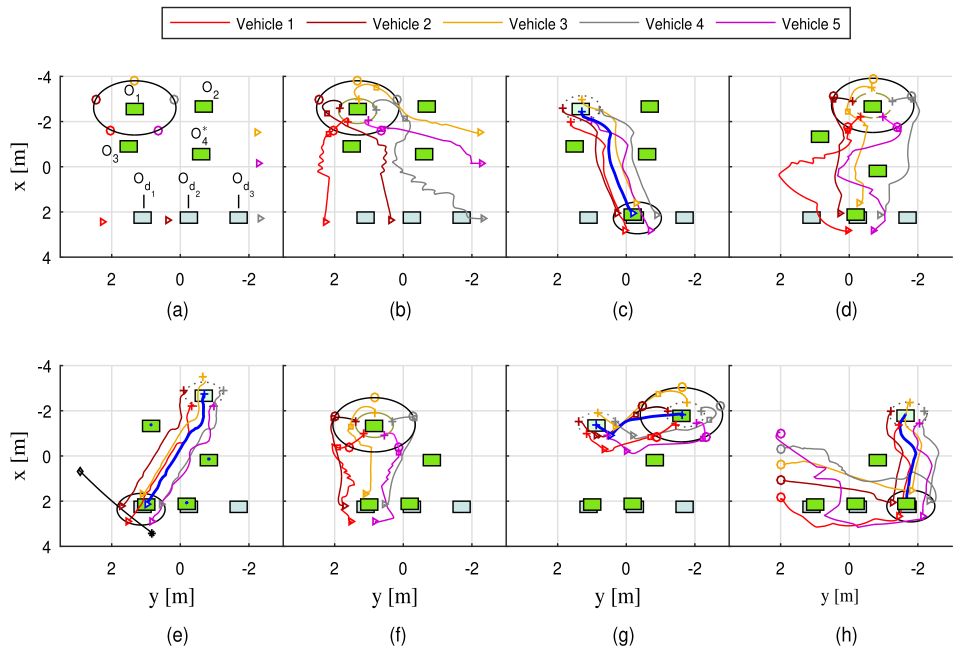

Fig. 12. Execution of group behaviors for experiment 2. Solid green boxes represent the movable objects, and blue boxes refer to the desired locations. Triangle marks represent the initial configuration of the MVS. The black circle corresponds to the desired geometric formation, whereas the small circles represent the vehicle target positions over the geometric formation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the vehicles switched to *State 2* for transporting O_3 to its desired location $\times \longrightarrow \triangle$ while avoiding O_4. The MVS accomplished the mission at 510 seconds. Thus, it switched to *State 3* to go toward the home configuration without collisions $\triangle \longrightarrow \circ$. Finally, Fig. 13 depicts the control profiles for this experiment. As it can be observed, control signals are below the velocity limits of ± 0.5 m/s.

4. Conclusion

In this paper, it is proposed an optimization-based control scheme for coordinating MVS in indoor environments. In particular, hierarchical quadratic programs are formulated to solve complex multi-vehicle tasks, such as object transportation, obstacle avoidance, tracking moving targets, reaching goal positions, keeping a given formation, among the most important. The underlying optimization solves a cascade of convex quadratic programs where equality and inequality constraints are satisfying following a strict hierarchy. In addition, complex missions assigned to the group can be designed with finite state machines that embed several stack of tasks.

It has verified the performance of the proposed controller with two experiments with six vehicles to track a moving target while keeping a formation and avoiding obstacles. A more complex mission has been successfully executed for cleaning the workspace with cooperative multi-vehicle tasks.

As future work, it is planned to extend the optimization-based controller to cope with consensus tasks within a decentralized setting as it has been suggested with the null-space approach in Trujillo et al. (2018).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

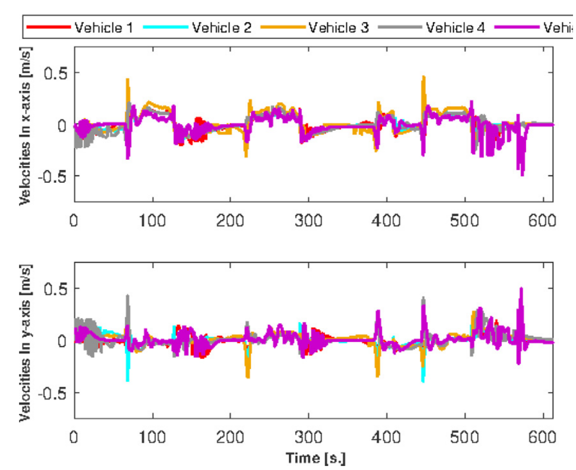

Fig. 13. Control profiles for experiment 2. The top-row shows the control profiles corresponding to five vehicles along the x-axis. Bottom-row depicts the control profiles along y-axis. The velocity limits are ± 0.5 m/s.

Acknowledgment

The third author acknowledge the financial support of the National Council of Science and Technology (CONACyT), Mexico within the project No. 285599, which is called *Toma de decisiones multiobjetivo para sistemas altamente complejos*.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.conengprac.2019.104206.

References

Alonso-Mora, J., Baker, S., & Rus, D. (2017). Multi-robot formation control and object transport in dynamic environments via constrained optimization. *International Journal of Robotics Research*, *36*(9), 1000–1021.

Antonelli, G., Arrichiello, F., & Chiaverini, S. (2009). Experiments of formation control with multirobot systems using the null-space-based behavioral control. *IEEE Transactions on Control Systems Technology*, *17*(5), 1173–1182.

Antonelli, G., & Chiaverini, S. (2006). Kinematic control of platoons of autonomous vehicles. *IEEE Transactions on Robotics*, *22*(6), 1285–1292.

Arechavaleta, G., Morales-Díaz, A., Pérez-Villeda, H. M., & Castelán, M. (2017). Hierarchical task-based control of multirobot systems with terminal attractors. *IEEE Transactions on Control Systems Technology*, *25*(1), 334–341.

Ayanian, N., & Kumar, V. (2010). Decentralized feedback controllers for multiagent teams in environments with obstacles. *IEEE Transactions on Robotics*, *26*(5), 878–887.

Balch, T., & Arkin, R. (1998). Behaviour-based formation control for multi-robot systems. *IEEE Transactions on Robotics and Automation*, *14*(6), 926–939.

Derenick, J. C., & Spletzer, J. R. (2007). Convex optimization strategies for coordinating large-scale robot formations. *IEEE Transactions on Robotics*, *23*(6), 1252–1259.

Draganjac, I., Miklić, D., Kovačić, Z., Vasiljević, G., & Bogdan, S. (2016). Decentralized control of multi-agv systems in autonomous warehousing applications. *IEEE Transactions on Automation Science and Engineering*, *13*(4), 1433–1447.

Escande, A., Mansard, N., & Wieber, P.-B. (2014). Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *International Journal of Robotics Research*, *33*(7), 1006–1028.

Fanti, M. P., Mangini, A. M., Pedroncelli, G., & Ukovich, W. (2018). A decentralized control strategy for the coordination of agv systems. *Control Engineering Practice*, *70*, 86–97.

Faverjorn, B., & Tournassoud, P. (1987). A local based approach for path planning of manipulators with a high number of degrees of freedom. In *IEEE international conference on robotics and automation* (pp. 1152–1159).

Ferreau, H., Kirches, C., Potschka, A., Bock, H., & Diehl, M. (2014). QpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, *6*(4), 327–363.

Gracia, L., Solanes, J. E., Munoz-Benavent, P., Esparza, A., Miro, J. V., & Tornero, J. (2018). Cooperative transport tasks with robots using adaptive non-conventional sliding mode control. *Control Engineering Practice*, *78*, 35–55.

Gutiérrez, H., Morales, A., & Nijmeijer, H. (2017). Synchronization control for a swarm of unicycle robots: Analysis of different controller topologies. *Asian Journal of Control*, *19*(19), 1–12.

Herzog, A., Rotella, N., Mason, S., Grimminger, F., Schaal, S., & Righetti, L. (2016). Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots*, *40*(3), 473–491.

Kallem, V., Komoroski, A., & Kumar, V. (2013). An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Robotics*, *9*(6), 1152–1159.

Kanehiro, F., Lamiraux, F., Kanoun, O., Yoshida, E., & Laumond, J.-P. (2008). A local collision avoidance method for non-strictly convex polyhedra. *Proc. Robot.: Sci. Syst. IV*.

Kanoun, O., Lamiraux, F., & Wieber, P.-B. (2011). Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, *27*(4), 785–792.

Krnjak, A., Draganjac, I., Bogdan, S., Petrović, T., Miklić, D., & Kovačić, Z. (2015). Decentralized control of free ranging agvs in warehouse environments. In *Robotics and automation (ICRA), 2015 IEEE international conference on* (pp. 2034–2041). IEEE.

Li, B., Liu, H., Xiao, D., Yu, G., & Zhang, Y. (2017). Centralized and optimal motion planning for large-scale agv systems: A generic approach. *Advances in Engineering Software*, *106*, 33–46.

Mousavi, M. A., Moshiri, B., & Heshmati, Z. (2015). Cooperative control of networked autonomous vehicles using convex optimization. In *Robotics and mechatronics (ICROM), 2015 3rd RSI international conference on* (pp. 681–687). IEEE.

Oh, K.-K., Park, M.-C., & Ahn, H.-S. (2015). A survey of multi-agent formation control. *Automatica*, *53*, 424–440.

Olmi, R., Secchi, C., & Fantuzzi, C. (2011). An efficient control strategy for the traffic coordination of agvs. In *Intelligent robots and systems (IROS), 2011 IEEE/RSJ international conference on* (pp. 4615–4620). IEEE.

Oriolo, G., De Luca, a., & Venditteli, M. (2002). Wmr control via dynamic feedback linearization: Design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, *10*(6), 835–852.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, Kobe, Japan, p. 5.

Reveliotis, S. A., & Roszkowska, E. (2011). Conflict resolution in free-ranging multivehicle systems: A resource allocation paradigm. *IEEE Transactions on Robotics*, *27*(2), 283–296.

Rudd, K., Foderaro, G., Zhu, P., & Ferrari, S. (2017). A generalized reduced gradient method for the optimal control of very-large-scale robotic systems. *IEEE Transactions on Robotics*, *33*(5), 1226–1232.

Samson, C., Borgne, M. L., & Espiau, B. (1991). *Oxford engineering science series*: *vol. 22*, *Robot control: The task function approach* (1st ed.). New York, USA: Oxford University Press.

Trujillo, M. A., Becerra, H. M., Gómez-Gutiérrez, D., Ruiz-León, J., & Ramírez-Treviño, A. (2018). Priority task-based formation control and obstacle avoidance of holonomic agents with continuous control inputs. *IFAC-PapersOnLine*, *51*(13), 216–222.

Wurman, P. R., D'Andrea, R., & Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, *29*(1), 9.