A Switching Model Predictive Control for Collision-Free Path-Tracking of Mobile Robots

Chrystian Pool Edmundo Yuca Huanca, Gian Paolo Incremona and Patrizio Colaneri

***Abstract*— This paper addresses the problem of collision-free path-tracking of a team of mobile robots, which move in the environment in the presence of obstacles. A model predictive control approach is presented, relying on the switched-system formalism to capture each robot model. Specifically, a minimal description of their dynamics, consisting of two modes (roto-traslation around a fixed pivot and rotation on spot) capable of efficiently capturing most of the possible motions on the plane, is considered to reduce the curse of dimensionality. Moreover, in order to enable robots to roam around while avoiding collisions and maintaining low computational complexity, obstacle avoidance constraints are relaxed to linear form and included in the optimization problem. The proposal is finally assessed both in simulation and experimentally on the Robotarium remote arena, in comparison with an intrusion-based obstacle avoidance strategy.**

***Index Terms*— Switched systems, tracking, collision avoidance, model predictive control, wheeled mobile robots.**

I. Introduction

In the last decades, unmanned ground vehicles (UGVs) have been investigated in many application fields ranging from logistics and surveillance [1] to search and rescue operations [2]. Among the UGV classes, wheeled mobile robots (WMRs) have shown easier manufacturing and control capabilities due to their simple and energy-efficient structure. Aware of the WMR benefits, when considering a multi-robot cooperating setting, properties such as robustness and flexibility naturally arise, improving the system performance even for complex tasks [3]. Despite the diverse challenges observed in various autonomous and cooperative control tasks, path tracking is still one of the most important task-attainment preconditions. Tracking controllers are designed to minimize the position error between the robot and the reference path or trajectory while accounting for disturbances and model uncertainties. In a complex environment, they are also required to fulfil secondary objectives in a hierarchical structure, such as collision avoidance, speed constraints, or connectivity maintenance.

Many results have been proposed to cope with the tracking objective, most of them integrating smart collision avoidance algorithms. For instance, among many others, a backstepping controller in cascade with a fuzzy sliding mode controller (SMC) is proposed in [4] for trajectory tracking in the presence of model mismatches and exogenous disturbances. In [5], a linear receding horizon (RH) controller for trajectory tracking is proposed to tackle the state-dependent input constraints originated when the robot kinematic model is feedback linearized. The zeroing control barrier function (CBF) notion, together with an emergency braking manoeuvre, is exploited in [6] to guarantee that the admissible control space of the double integrator model of the multi-robot system is big enough to prioritize nominal control actions while limiting invasive collision avoidance reactions. A multi-objective Lagrangian-based controller is designed in [7] to solve target tracking and collision avoidance in a shared environment (collision among robots and obstacles). Global asymptotic stability is proven in [8], where a hybrid controller is proposed under a differential game formulation of the tracking objective with collision-free and deadlock-free properties.

Furthermore, multiple agents require robust, flexible and scalable control algorithms for coordination and cooperative tasks. Indeed, limitations, such as motion congestions and communications conflicts, can affect the scalability of the control algorithms, even drastically in the case of centralized solutions [9]. Adopting simple rules is, therefore, instrumental in facing these issues. Among the solutions, those proposed in [10], [11] rely on switching systems theory to perform an aggregation objective for WMRs and unmanned aerial vehicles (UAVs), respectively, by reducing the complexity of the robot model. Although straightforward reasoning would lead one to think of a direct extension of the path-tracking objective, such a modification requires a preprocessing of the intended path to follow in order to ensure its practical feasibility.

Spurred by these motivations, we address the problem of navigation of WMRs, specifically focusing on the path-tracking sub-problem. In fact, under the assumption that the path planning and path generation sub-problems are preliminarily solved, an original easy-to-implement control solution to track the reference points while avoiding obstacles is proposed by exploiting switching systems theory. Inspired by [12], the differential wheeled robot (DWR) dynamics is constrained to two motion modes (rototraslation around a fixed pivot and rotation on spot), thus allowing to recast the robot network as a switched system. A switching model predictive control (SMPC) is then proposed to synthesize the optimal solution to the tracking control problem while

This is the final version of the accepted paper submitted for inclusion in the Proceedings of the 2024 IEEE 63rd Conference on Decision and Control (CDC), Dec., 2024, DOI: 10.1109/CDC56724.2024.10886785. This work has been supported by the Italian Ministry of Enterprises and Made in Italy in the framework of the project 4DDS (4D Drone Swarms) under grant no. F/310097/01-04/X56. The authors would like to thank the student Federico Neri for the full help provided during the simulation tests.

Chrystian Pool Edmundo Yuca Huanca, Gian Paolo Incremona and Patrizio Colaneri are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy (e-mails: *chrystianpool.yuca@polimi.it*, *gianpaolo.incremona@polimi.it*, *patrizio.colaneri@polimi.it*).

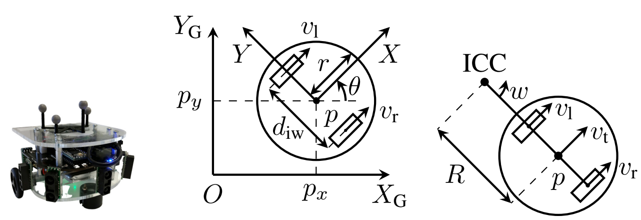
dealing with the switching nature of the considered system and a challenging environment configuration. The SMPC is embedded in a centralized supervisor responsible for gathering the network information, computing the optimal control signals, and sending them to each agent. The environment settings, including the position and dimension of the obstacles and the reference trajectory, are considered available in advance. Moreover, the SMPC formulation integrates obstacle constraints via a polytope-based approximation [13] to further enhance a computational-efficient direction accounting for large-scale implementation. The proposed approach is assessed in simulation over four environment configurations, and its implementation feasibility is validated through experiments performed in a remote swarm-robotic testbed. An alternative methodology based on the deformable virtual zone (DVZ) for obstacle avoidance [14] is also evaluated for the sake of comparison.

II. Modelling and Problem Formulation

In this section, the considered robot model is first introduced and recast in the switched-systems formalism, and then the control objective for a team of WMRs is formulated.

A. Differential wheeled robot model

A mobile robot with a differential wheeled configuration is constituted by two main wheels, individually controlled, and a third passive castor wheel, which stabilises the robot, see Fig. 1(a). This configuration allows the planar motion of the robot by setting linear velocities to the main wheels. For the sake of simplicity, the robot is assumed to have a symmetric circular body with centre of mass and centroid coinciding at point $p = [p_x \ p_y]^\top$, whereas Fig. 1(b) depicts the global ($X_\mathrm{G} - Y_\mathrm{G}$) and body ($X - Y$) frames, the inter-wheel distance d_iw, the left and right wheel linear velocities v_l and v_r, the body radius r, and the robot orientation θ.



(a) robot (b) robot planar view (c) ICC point

Fig. 1. The considered wheeled robot (a) and schematic rendering in the plane (b) with instantaneous centre of curvature (c).

Moreover, in a WMR, the rolling motion of the wheels is constrained by the existence of the instantaneous centre of curvature (ICC), a point where the wheels rotate about, commonly found by connecting the lateral axis of each wheel. The DWR case, where the castor wheel is ignored to reduce the model complexity, is illustrated in Fig.1(c). According to [15], while the robot rotates, the angular speed w and the distance between the ICC and the centroid, named R, are defined by the linear wheel velocities v_r and v_l as follows:

$$R = \frac{d_\mathrm{iw}}{2}\left(\frac{v_\mathrm{r}+v_\mathrm{l}}{v_\mathrm{r}-v_\mathrm{l}}\right), \quad w = \frac{1}{d_\mathrm{iw}}\left(v_\mathrm{r}-v_\mathrm{l}\right). \tag{1}$$

In our relaxed setup, the robot linear velocity v_t and its vertical and horizontal components projected in the global frame axes correspond to

$$v_\mathrm{t} = wR = \frac{v_\mathrm{r}+v_\mathrm{l}}{2}, \quad \dot{p}_x = v_\mathrm{t}\cos\theta, \quad \dot{p}_y = v_\mathrm{t}\sin\theta. \tag{2}$$

Finally, let us define the system state $x \coloneqq [p_x \ p_y \ \theta]^\top \in \mathbb{R}^3$ as the vector collecting robot poses, i.e., global position and orientation, and the system input $u \coloneqq [v_\mathrm{r} \ v_\mathrm{l}]^\top \in \mathbb{R}^2$ as the vector accounting for the linear velocities of the wheels, so that the DWR kinematic model can be written as

$$\dot{x} = f(x,u), \quad x(0) = x_0. \tag{3}$$

B. Switched DWR model

Having in mind a centralized control architecture, and to alleviate the curse of dimensionality in the case of large-scale systems, the DWR model is recast to constrain the robot dynamics to a finite set of motion modes, exploiting the switched systems formalism. In this regard, let m be the number of predefined motion modes such that the switching signal $\sigma \in \mathcal{M} \coloneqq \{1,\ldots,m\}$ defines the switching input $u(\sigma) \in \mathcal{U}_\mathrm{sw}$, where $\mathcal{U}_\mathrm{sw}$ is the set containing all possible input vectors. Then, the switched time-invariant form of (3) is written as

$$\dot{x} = f_\sigma(x, u(\sigma)), \quad x(0) = x_0, \tag{4}$$

where f_σ belongs to the set $\mathcal{Q} \coloneqq \{f_1,\ldots,f_m\}$.

Analogously to [10] and [12], this study considers the rotation around the robot ICC defined by two different fixed pairs of wheel velocities ($m = 2$). Thus, the corresponding switching input is given by

$$u(\sigma) = \begin{cases} [v_\mathrm{r1} \ v_\mathrm{l1}]^\top, & \sigma = 1 \\ [v_\mathrm{r2} \ v_\mathrm{l2}]^\top, & \sigma = 2 \end{cases}, \tag{5}$$

and the elements of $\mathcal{Q}$ are

$$f_1 = \begin{bmatrix} 0 \\ 0 \\ \frac{(v_\mathrm{r1}-v_\mathrm{l1})}{d_\mathrm{iw}} \end{bmatrix}, \quad f_2 = \begin{bmatrix} \frac{1}{2}\left(v_\mathrm{r2}+v_\mathrm{l2}\right)\cos\theta \\ \frac{1}{2}\left(v_\mathrm{r2}+v_\mathrm{l2}\right)\sin\theta \\ \frac{(v_\mathrm{r2}-v_\mathrm{l2})}{d_\mathrm{iw}} \end{bmatrix},$$

capturing the rotation on spot (i.e., $v_{\mathrm{r}_1} = -v_{\mathrm{l}_1}$) and around a fixed pivot (i.e., $v_{\mathrm{r}_1} \neq v_{\mathrm{l}_1} \neq 0$), respectively.

C. Multi-DWR switched model and problem statement

After describing the single-robot switched model, we are now in a position to derive the model expression for the multi-robot system. In the following, the apex $^{[i]}$ denotes the index of the ith element in a set, and we assume that a centralized supervisor communicates to all the agents the sequence of objective points (i.e., it executes the path planning and generation). Moreover, let n be the number of agents in the system, $\mathcal{N} \coloneqq \{i \in [1,\ldots,n]\}$ be the set collecting all agents indexes, $x \coloneqq [x^{[1]^\top},\ldots,x^{[n]^\top}]^\top$ and

$u := [u^{[1]^\top}, \dots, u^{[n]^\top}]^\top$ be the redefined state and input vectors, respectively, and $\Sigma := \{\sigma^{[1]}, \dots, \sigma^{[n]}\} \in \mathcal{S} := \mathcal{M}^n$ be the network switching string. Therefore, the multi-DWRs switched model is

$$\dot{x} = f_\Sigma(x, u(\Sigma)), \quad x(0) = x_0, \tag{6}$$

where $u(\Sigma) \in \mathcal{V} := \mathcal{U}_{\text{sw}}^n$, and $f_\Sigma \in \mathcal{Q}_n := \{f_1, \dots, f_{m^n}\}$. Considering the future practical implementation of a receding horizon approach, the discrete-time counterpart of (6) is derived, by selecting T as the sampling time, and $k \in \mathbb{N}$ as the discrete-time variable, so that, according to forward Euler discretization method, one has

$$x_{k+1} = x_k + T f_\Sigma(x_k, u_k(\Sigma)) = f_{\text{d}_\Sigma}(x_k, u_k(\Sigma)), \tag{7}$$

with $f_{\text{d}_\Sigma} \in \mathcal{Q}_{\text{d}_n} := \{f_{\text{d}_1}, \dots, f_{\text{d}_{m^n}}\}$.

Therefore, the control problem consists of *requiring a team of DWRs, constrained to m^n motion modes as represented in* (7)*, to track a planned path, while avoiding inter-robot and obstacle-robot collisions*. To pursue this goal, the switching nature of the multi-agent system is exploited.

III. The Proposed Switching Model Predictive Control With Collision Avoidance

This section formalizes the previously mentioned path-tracking objective assigned to the DWRs as a finite horizon optimal control problem, and introduces the proposed SMPC.

A. Path preprocessing and cost design

In order to define the performance function for the tracking control problem, let us first define the set $\mathcal{X}_{\text{ref}}$ containing the position of n_{ref} reference points with index set $\mathcal{N}_{\text{ref}} := \{i \in [1, \dots, n_{\text{ref}}]\}$ that describe the desired path provided by the supervisor. Moreover, assume that the considered environment contains n_{obs} predefined static obstacles, whose centroid coordinates are collected in the set $\mathcal{X}_{\text{obs}}$ with index set $\mathcal{N}_{\text{obs}} := \{i \in [1, \dots, n_{\text{obs}}]\}$.

Then, preprocessing is required to avoid unreachable reference points and assign the initial reference position to each robot. For the sake of simplicity, let us assume that the obstacles have a circular shape with radius $r_{\text{obs}}^{[i]}$, where $i \in \mathcal{N}_{\text{obs}}$. Thus, the set of reachable reference points for the robot network is defined as

$$\mathcal{X}_{\text{rch}} := \{p_{\text{rch}} \in \mathcal{X}_{\text{ref}} \mid \|p_{\text{rch}} - p_{\text{obs}}^{[i]}\|_2 > r + r_{\text{obs}}^{[i]}, \forall i \in \mathcal{N}_{\text{obs}}\}. \tag{8}$$

Moreover, letting $\mathcal{N}_{\text{rch}}$ be the set of indexes associated to $\mathcal{X}_{\text{rch}}$, the initial reference point, namely $p_{\text{ref}_{o^{[i]}}}^{[i]} \in \mathcal{X}_{\text{rch}}$, of the ith robot is selected such that

$$o^{[i]} = \arg\min_{j \in \mathcal{N}_{\text{rch}}} \|p_0^{[i]} - p_{\text{rch}}^{[j]}\|_2, \tag{9}$$

while the updating rule of the reference index, i.e., $p_{\text{ref}_s}^{[i]} \to p_{\text{ref}_{s+1}}^{[i]}$, with $s \in \mathcal{N}_{\text{rch}}$, is state-dependent and occurs at any time $k > 0$ such that $\|p_k^{[i]} - p_{\text{ref}_s}^{[i]}\|_2 < \epsilon$, where $\epsilon > 0$ is a predefined parameter to avoid computational errors.

Given the preprocessed path, the proposed cost function for the tracking problem, at time instant $k \geq 0$, is given by

$$J_k = \sum_{i \in \mathcal{N}} \|p_k^{[i]} - p_{\text{ref}_s}^{[i]}\|_2, \quad p_{\text{ref}_{o^{[i]}}}^{[i]} \in \mathcal{X}_{\text{rch}}, \tag{10}$$

where s is the current reference index at time k. The goal becomes, therefore, that of minimizing the distance between each robot and the feasible reference points, which belong to the set (8) of time-varying reference points with initial index given by (9). Such an index is updated to the next one once the corresponding agent reaches its current target position. All is subject to the switched dynamics (7), while avoiding any collisions. The latter objective is hereafter discussed.

B. Polytope-based collision avoidance constraints

Since the considered optimal control problem to solve requires the integration of collision avoidance constraints without affecting the computational complexity of the solution, linear terms to describe obstacle barriers are integrated into the proposed SMPC, by exploiting the strategy in [13].

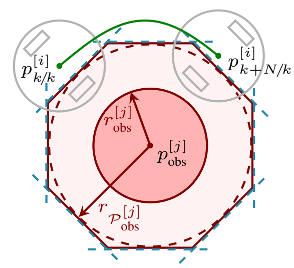


Fig. 2. Polytope $\mathcal{P}_{\text{obs}}^{[j]}$ (blue dashed lines) associated to the jth obstacle (red circle). The pink area represents the linear approximation of the dashed red circle, while the green line depicts the predicted trajectory of the ith robot (gray circle).

In fact, a natural reasoning for addressing collision avoidance involves keeping the distance between the agent and the obstacle greater than zero. In our case, this condition can be formulated such that, at time instant k, the following holds

$$\|p_k^{[i]} - p_{\text{obs}}^{[j]}\|_2 > r + r_{\text{obs}}^{[j]}, \quad \forall\, i \in \mathcal{N}, j \in \mathcal{N}_{\text{obs}}. \tag{11}$$

However, the applicability of (11) in large-scale optimization control problems is constrained by its nonlinear and non-convex properties, which discard conventional quadratic programming solvers, thus increasing the computational load up to a possibly unfeasible implementation. Considering instead linear constraints [13], as for the jth circular obstacle, the corresponding set of n_{pol} linear inequalities, with index set $\mathcal{N}_{\text{pol}} := \{\nu \in [1, \dots, n_{\text{pol}}]\}$, belongs to the polytope $\mathcal{P}_{\text{obs}}^{[j]}$, which circumscribes a concentric circle with radius $r_{\mathcal{P}_{\text{obs}}^{[j]}} = r + r_{\text{obs}}^{[j]}$, see Fig. 2. The inequalities expressions are given by

$$\boldsymbol{h}_{\mathcal{P}_{\text{obs}}^{[j]}}(p_k^{[i]} - p_{\text{obs}}^{[j]}) < \boldsymbol{r}_{\mathcal{P}_{\text{obs}}^{[j]}}, \quad \forall\, i \in \mathcal{N}, j \in \mathcal{N}_{\text{obs}}, \tag{12}$$

with vectors defined as $\boldsymbol{h}_{\mathcal{P}_{\text{obs}}^{[j]}} = [h_{\mathcal{P}_{\text{obs}}^{[j]}}^{[1]^\top}, \dots, h_{\mathcal{P}_{\text{obs}}^{[j]}}^{[n_{\text{pol}}]^\top}]^\top$, $h_{\mathcal{P}_{\text{obs}}^{[j]}}^{[\nu]} = \left[\cos\frac{2\pi(\nu-1)}{n_{\text{pol}}}\ \sin\frac{2\pi(\nu-1)}{n_{\text{pol}}}\right]$, and $\boldsymbol{r}_{\mathcal{P}_{\text{obs}}^{[j]}}$ being a column vector with all elements $r_{\mathcal{P}_{\text{obs}}^{[j]}}$.

Note that (12) is only sufficient since it can only detect if the ith robot and the jth obstacle are in collision, but not the *no-collision* condition. Indeed, such an opposite condition is given by the union of half-spaces. Nevertheless, it can be identified by finding one $\bar{\nu} \in \mathcal{N}_{\text{pol}}$ such that

$$\rho_{\text{obs}_{(\bar{\nu},i,j,k)}} = h^{[\bar{\nu}]}_{\mathcal{P}^{[j]}_{\text{obs}}}(p^{[i]}_k - p^{[j]}_{\text{obs}}) - r_{\mathcal{P}^{[j]}_{\text{obs}}} > 0, \tag{13a}$$

e.g., by solving

$$\bar{\nu} = \arg\max_{\nu \in \mathcal{N}_{\text{pol}}} \rho_{\text{obs}_{(\nu,i,j,k)}}. \tag{13b}$$

On the other hand, concerning inter-robot collision avoidance, the previous constraints can be extrapolated to consider a moving robot as an obstacle with a time-varying position. Thus, the corresponding condition expression for the ith and jth robot at time instant k is given by

$$\rho_{\text{rob}_{(\bar{\nu},i,j,k)}} = h^{[\bar{\nu}]}_{\mathcal{P}^{[j]}_{\text{rob}}}(p^{[i]}_k - p^{[j]}_k) - r_{\mathcal{P}^{[j]}_{\text{rob}}} > 0, \tag{14a}$$

similarly to (13), where one can find a valid $\bar{\nu}$ as

$$\bar{\nu} = \arg\max_{\nu \in \mathcal{N}_{\text{pol}}} \rho_{\text{rob}_{(\nu,i,j,k)}}, \; j \in \mathcal{N}, i \neq j. \tag{14b}$$

Even though a centralized structure grants the system full communication capability and allows to assess all robot-obstacle and robot-robot constraint conditions, a reduction in the complexity of the optimization problem can be achieved by evaluating the collision condition only between robots and their proximal agents (robots or obstacles). An agent is considered proximal if the distance between the robot and the agent is less than a predefined distance r_{prx}. Therefore, the new pair of index sets for the proximal agents associated with the ith robot modifies the expression for $\bar{\nu}$, and the set of the indexes of the proximal agents are

$$\begin{aligned} \mathcal{N}^{[i]}_{\text{obs}\,k} &:= \{j \in \mathcal{N}_{\text{obs}} \mid \|p^{[i]}_k - p^{[j]}_{\text{obs}}\|_2 < r_{\text{prx}}\}, \\ \mathcal{N}^{[i]}_k &:= \{j \in \mathcal{N} \mid \|p^{[i]}_k - p^{[j]}_k\|_2 < r_{\text{prx}}, j \neq i\}. \end{aligned} \tag{15}$$

C. Finite horizon optimal control problem (FHOCP)

Given model (7), cost function (10), and the constraint conditions (13) and (14), we are in a position to formulate the FHOCP. Let $\mathcal{T}_k := \{k, \dots, k+N-1\}$ be the prediction horizon time vector, where $N \geq 1$ is the horizon length, and $t \in \mathcal{T}_k$ be the spanning time variable. Thus, the proposed centralized SMPC problem to solve is

$$\begin{aligned} &\min_{\boldsymbol{\Sigma}_k} \; J_k = \sum_{t \in \mathcal{T}_k} \sum_{i \in \mathcal{N}} \|p^{[i]}_t - p^{[i]}_{\text{ref}\,s}\|_2 \\ &\text{subject to, } \forall t \in \mathcal{T}_k, \; p^{[i]}_{\text{ref}\,o^{[i]}} \in \mathcal{X}_{\text{rch}} \\ &\quad x_{t+1} = f_{d_{\Sigma_t}}(x_t, u_t(\Sigma_t)), \; x_k = \tilde{x}_k \\ &\quad \rho_{\text{obs}_{(\bar{\nu},i,j,t)}} > 0, \; \forall i \in \mathcal{N}, j \in \mathcal{N}^{[i]}_{\text{obs}\,t}, \\ &\quad \rho_{\text{rob}_{(\bar{\nu},i,j,t)}} > 0, \; \forall i \in \mathcal{N}, j \in \mathcal{N}^{[i]}_t, \end{aligned} \tag{16}$$

where $\boldsymbol{\Sigma}_k := \{\boldsymbol{\sigma}^{[1]}_k, \dots, \boldsymbol{\sigma}^{[n]}_k\}$, $\boldsymbol{\sigma}^{[i]}_k := [\sigma^{[i]}_k, \dots, \sigma^{[i]}_{k+N-1}]$, and $\tilde{x}_k$ is the state reading at time instant k.

Remark 3.1 (Feasibility): As for the feasibility of the proposed SMPC, the set of possible sequences, namely $\mathcal{W}$, is feasible by construction since a constant zero vector along the prediction horizon guarantees constraint fulfilment. Moreover, it is possible to consider motion policies to further reduce the computational burden (e.g., only one robot can change its mode at each time instant), thus taking the optimal sequences inside a feasible subset, namely $\mathcal{F} \subset \mathcal{W}$. As for the recursive feasibility, instead, in case of regulation or tracking problems, it can be guaranteed by including a suitable terminal constraint and by relying on a suitable auxiliary law such that all the constraints remain fulfilled. However, it is worth mentioning that the definition of such a term is not always common and straightforward in practical problems (see e.g., [16, §1]). $\triangledown$

IV. Case Study

This section presents the case study and the simulation and experimental outcome assessing the proposed SMPC.

A. Settings

The considered robot setup and the corresponding model parameters are those of a real robotic platform as reported in [10, §5]. Moreover, the SMPC performance is evaluated over four different reference paths (ellipse, cardioid, drop, and star) for a team of $n = 5$ mobile robots in an environment with $n_{\text{obs}} = 6$ circular obstacles with different radius. The initial distribution of the robots and the obstacles over the plane, with dimension 3.2×2 m, have been randomly generated and fixed to provide a fair comparison between collision algorithms for each reference path. Finally, the chosen simulation time has been 200 seconds with a sampling time of $T = 0.033$ s, and prediction horizon length $N = 5$.

Moreover, the proposed SMPC in (16) with polytopic obstacle avoidance constraints has been compared with the solution of an SMPC embedding a DVZ mechanism to consider possible collisions. Specifically, proximity information is exploited to determine the deformation of a virtual zone attached to an agent, produced by the interaction with the environment [14]. The formulation of the deformable zone

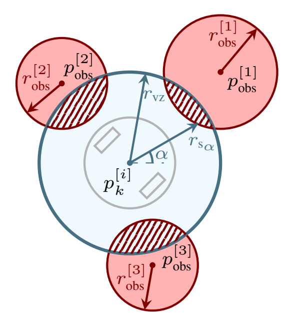


Fig. 3. Deformable virtual zone (blue area) of the ith robot (gray circle) at time k. The deformation regions produced by the obstacle intrusions (red circles) are highlighted with dashed red lines.

geometry is simplified by the choice of a fixed circular region of radius r_{vz} and concentric with the DWR, see Fig 3. Furthermore, the intrusion information variable $\mathcal{I}^{[i]}_k$ quantifies the virtual zone distortion of the ith robot at time instant k as $\mathcal{I}_k^{[i]} := \int_0^{2\pi} \frac{r_{\mathrm{vz}} - r_{\mathrm{s}_{\alpha,k}} + r}{r_{\mathrm{s}_{\alpha,k}} - r}\, d\alpha$, where $r_{\mathrm{s}_{\alpha,k}}$ is the intrusion sensor reading with angular orientation α. Hence, an increasing intrusion value is obtained as $r_{\mathrm{s}_{\alpha,k}} \to r$. This implies that, if bounded intrusion information is achieved for all time, a collision-free motion is guaranteed. In addition, the collision avoidance reactivity of the system corresponds to the inclusion in the cost function (10) of the intrusion term given by $V_{\mathrm{dvz}_k} = \sum_{i\in\mathcal{N}} \frac{1}{2}{\mathcal{I}_k^{[i]}}^2$, in place of the polytopic constraints defined in (16).

B. Simulation results

Fig. 4 shows the path-tracking of the robots obtained when the SMPC in (16) is solved, with initial and final positions represented by coloured circles and stars, respectively. Fig. 5 complementarily illustrates the time evolution of the cost function average. It is worth noticing that the peaks appearing in the cost evolution are due to the change of reference positions, according to the procedure in §III-A. As expected, the proposed SMPC successfully steers the robots to follow the different reference paths, while avoiding collisions.

The same scenario has been adopted for the DVZ-based SMPC. For the sake of comparison, average values of the cost functions over the whole simulation time are computed, namely $\tilde{J}_{\mathrm{dvz}}$ and $\tilde{J}_{\mathrm{pol}}$, together with the computational times, t_{dvz} and t_{pol}, and their relative percentage errors, $\eta_{\tilde{J}_{\mathrm{pol/dvz}}}$ and $\eta_{t_{\mathrm{pol/dvz}}}$. The outcomes are reported in Table I. Due to the linear approximation of the obstacle constraints, in most of the scenarios, adopting a polytopic-based method inside the proposed SMPC (16) leads to greater values of the average cost with respect to adopting a DVZ approach. However, the latter methodology requires a higher computational time (an order of magnitude), which could affect the tractability of the DVZ approach for larger-scale multi-robot systems. Table I evidences a clear trade-off between cost (goal performance) and computational time (scalability).

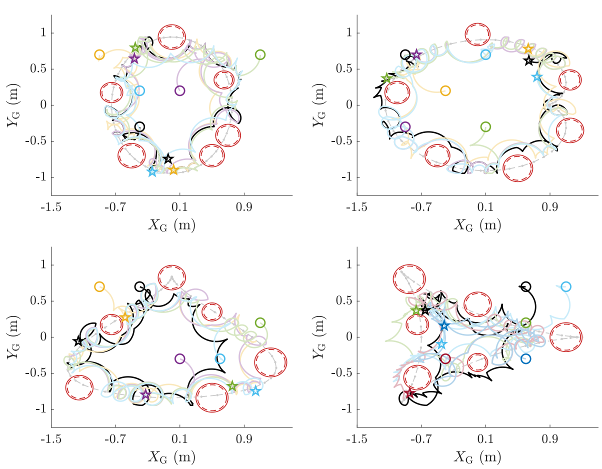


Fig. 4. Planar views of the network trajectories obtained in simulations for ellipse, cardioid, drop, and star paths.

C. Experimental results

In order to assess the practical feasibility of the proposal, the same tests illustrated in the previous subsection have also

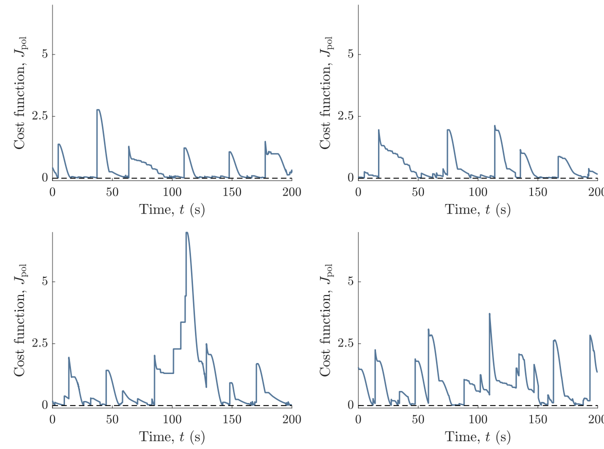
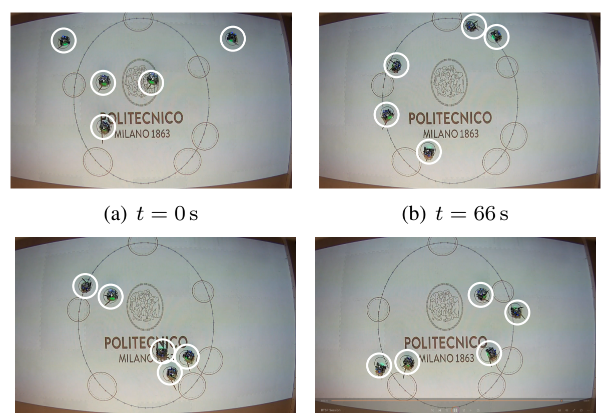


Fig. 5. Time evolutions of the average cost J_{pol} obtained in simulation for ellipse, cardioid, drop, and star paths.

TABLE I
AVERAGE INDEXES IN SIMULATION.

Path	$\tilde{J}_{\mathrm{dvz}}$	$\tilde{J}_{\mathrm{pol}}$	$\eta_{\tilde{J}_{\mathrm{pol/dvz}}}$ (%)	t_{dvz} (s)	t_{pol} (s)	$\eta_{t_{\mathrm{pol/dvz}}}$ (%)
ellipse	1.23	0.86	30.1%	0.0021	0.00079	62.38%
cardioid	0.86	0.96	-11.6%	0.0022	0.00075	65..9%
drop	1.23	1.85	-50.4%	0.0021	0.00084	60%
star	1.11	1.79	-61.26%	0.0021	0.00074	64.76%

been implemented using the Robotarium remote platform [17] (see Fig.6, where four frames of the experiment with ellipsoidal path are shown). The settings are kept unchanged



(a) $t = 0\,\mathrm{s}$ (b) $t = 66\,\mathrm{s}$ (c) $t = 132\,\mathrm{s}$ (d) $t = 200\,\mathrm{s}$

Fig. 6. Snapshots of the robot network during an experiment: white circles denote the robot space, while the projected circles in the background are the obstacles.

except for the robot radius (a value equal to the robot diameter has been considered) and the maximum linear velocity of the wheels (reduced by half), for safety reasons.

In Fig. 7, one can observe that the robots describe free-collision path-tracking, while Fig. 8 represents the corresponding time evolution of the average cost function. Unlike the simulated case, the robot motion and the time evolution of the cost are slightly affected by unavoidable unmodeled dynamics errors, communication delays, and external disturbances. However, the robust property of the SMPC manages to induce the system to track the reference even in those conditions, validating the implementation feasibility of the proposed approach. Table II reports the achieved performance indexes. Note that, in this case, the computation times provided by the platform take into account also the interval to send the input to the robots. Therefore, although consistent with simulations, the difference between the two compared methodologies looks less marked.

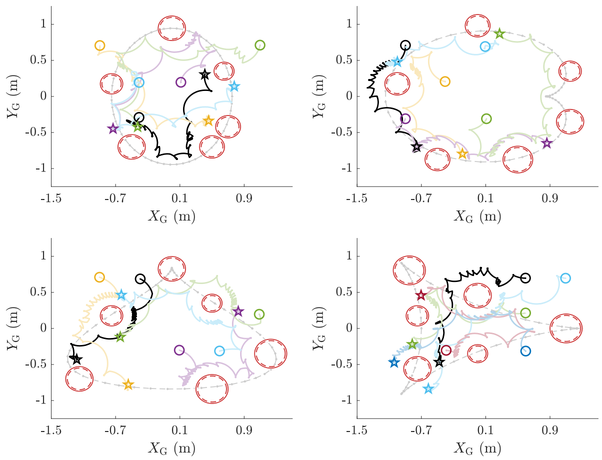


Fig. 7. Planar views of the network trajectories obtained during experiments for ellipse, cardioid, drop, and star paths.

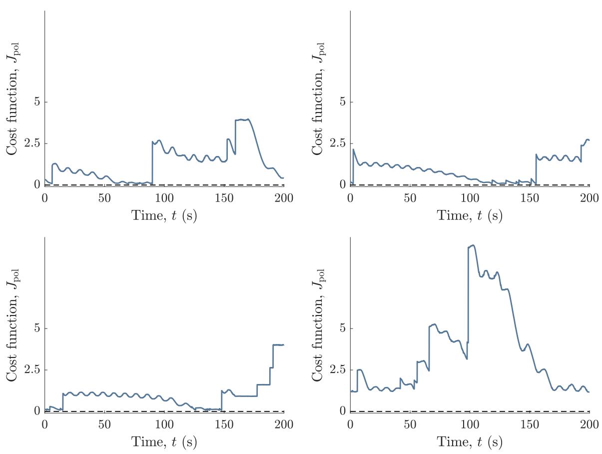


Fig. 8. Time evolutions of the average cost J_{pol} obtained during experiments for ellipse, cardioid, drop, and star paths.

TABLE II

AVERAGE INDEXES USING THE ROBOTARIUM SETUP.

Path	$\tilde{J}_{dvz}$	$\tilde{J}_{pol}$	$\eta_{\tilde{J}_{pol/dvz}}$ (%)	t_{dvz} (s)	t_{pol} (s)	$\eta_{t_{pol/dvz}}$ (%)
ellipse	1.46	2.91	-99.31%	0.0052	0.0034	34.61%
cardioid	1.12	0.96	14.28%	0.0057	0.0035	38.59%
drop	1.35	1.12	17.03%	0.0055	0.0054	1.81%
star	2.69	4.35	-61.71%	0.0049	0.0034	30.61%

V. CONCLUSIONS

This paper has introduced an original SMPC strategy to address the problem of path-tracking of mobile robots in an environment with obstacles. The latter is considered by considering a polytope-based approximation for the constraints and compared with a DVZ-based method. Ongoing research is devoted to developing a distributed control architecture to further improve the scalability of the approach, and to more complex settings with moving obstacles or the extension to the spatial case of UAVs.

REFERENCES

[1] F. Rubio, F. Valero, and C. Llopis-Albert, “A review of mobile robots: Concepts, methods, theoretical framework, and applications,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, pp. –, 2019.

[2] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, “Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision,” *IEEE Access*, vol. 8, no. -, pp. 191 617–191 643, 2020.

[3] A. Khamis, A. Hussein, and A. Elmogy, *Multi-robot Task Allocation: A Review of the State-of-the-Art*. Cham: Springer International Publishing, 2015, pp. 31–51.

[4] X. Wu, P. Jin, T. Zou, Z. Qi, H. Xiao, and P. Lou, “Backstepping trajectory tracking based on fuzzy sliding mode control for differential mobile robots,” *Journal of Intelligent & Robotic Systems*, vol. 96, no. 1, pp. 109–121, Oct. 2019.

[5] C. Tiriolo, G. Franzè, and W. Lucia, “A receding horizon trajectory tracking strategy for input-constrained differential-drive robots via feedback linearization,” *IEEE Transactions on Control Systems Technology*, vol. 31, no. 3, pp. 1460–1467, 2023.

[6] L. Wang, A. D. Ames, and M. Egerstedt, “Safety barrier certificates for collisions-free multirobot systems,” *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[7] X. Li, Z. Xu, S. Li, Z. Su, and X. Zhou, “Simultaneous obstacle avoidance and target tracking of multiple wheeled mobile robots with certified safety,” *IEEE Transactions on Cybernetics*, vol. 52, no. 11, pp. 11 859–11 873, 2022.

[8] D. Cappello, S. Garcin, Z. Mao, M. Sassano, A. Paranjape, and T. Mylvaganam, “A hybrid controller for multi-agent collision avoidance via a differential game formulation,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1750–1757, 2021.

[9] K. Soma, V. S. Vardharajan, H. Hamann, and G. Beltrame, “Congestion and scalability in robot swarms: a study on collective decision making,” 2023.

[10] C. E. P. Yuca Huanca., G. P. Incremona., R. Groß., and P. Colaneri., “Design of a switched control Lyapunov function for mobile robots aggregation,” in *Proc. of the 19th International Conference on Informatics in Control, Automation and Robotics*, Jul. 2022, pp. 492–499.

[11] C. P. E. Y. Huanca, G. P. Incremona, and P. Colaneri, “Design of a distributed switching model predictive control for quadrotor UAVs aggregation,” *IEEE Control Systems Letters*, vol. 7, no. -, pp. 2964–2969, 2023.

[12] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß, “Self-organized aggregation without computation,” *The International Journal of Robotics Research*, vol. 33, no. 8, pp. 1145–1161, 2014.

[13] M. Farina, A. Perizzato, and R. Scattolini, “Application of distributed predictive control to motion and coordination problems for unicycle autonomous robots,” *Robotics and Autonomous Systems*, vol. 72, pp. 248–260, 2015.

[14] L. Lapierre, R. Zapata, and P. Lepinay, “Simulatneous path following and obstacle avoidance control of a unicycle-type robot,” in *Proc. of the 2007 IEEE International Conference on Robotics and Automation*, vol. -, no. -, Rome, Italy, Apr. 2007, pp. 2617–2622.

[15] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Cambridge, UK: Cambridge University Press, 2010.

[16] L. Magni, G. D. Nicolao, L. Magnani, and R. Scattolini, “A stabilizing model-based predictive control algorithm for nonlinear systems,” *Automatica*, vol. 37, no. 9, pp. 1351–1362, 2001.

[17] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, “The Robotarium: A remotely accessible swarm robotics research testbed,” 2016.