POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Design of robust and stochastic model predictive control strategies for collision-free trajectory tracking of multiple mobile robots

Tesi di Laurea Magistrale in
Automation and Control Engineering

Author: **Ernesto Ceramelli**

Student ID: 976297
Advisor: Prof. Patrizio Colaneri
Co-advisors: Prof. Gian Paolo Incremona, Chrystian P. E. Yuca Huanca
Academic Year: 2022-23

Abstract

The main goal of this thesis is to develop solutions in the field of switched systems that allow a network of robots to safely navigate an environment with obstacles while guaranteeing collision avoidance. The thesis extends previous works where a network of differential wheeled robots is considered and only two motions are allowed: rotation on the spot and roto-translation describing a circular path. To manage collision avoidance, the problem is reformulated in terms of a model predictive control description, which, thanks to its flexibility, allows to tackle several tasks for the network. In order to alleviate the computational burden, a distributed approach has been developed along with a robust and stochastic formulation of the model predictive control able to cope with disturbances acting on the robots. To conclude, a novel robust model predictive control approach based on disturbances reachable set is developed, resulting in a control scheme that improves on previous solutions in terms of computational effort, flexibility, and robustness.

Keywords: mobile robotics, switched system, model predictive control, robust MPC, stochastic MPC.

Abstract in lingua italiana

L'obiettivo principale di questa tesi è sviluppare soluzioni nel campo dei sistemi a commutazione che consentano a una rete di robot di navigare in sicurezza in un ambiente con ostacoli, garantendo al contempo di evitare le collisioni. La tesi estende lavori precedenti in cui si considera una rete di robot a ruote differenziali per i quali sono ammessi solo due moti, la rotazione sul posto e la roto-traslazione che descrive un percorso circolare. Per evitare le collisioni, il problema viene descritto in modo da applicare una strategia di controllo predittivo, che grazie alla sua flessibilità permette di affrontare diversi compiti per la rete. Per alleggerire il carico computazionale, è stato sviluppato un approccio distribuito e una formulazione robusta e stocastica del controllo predittivo in grado di far fronte ai disturbi che agiscono sui robot. In conclusione, è stato sviluppato un nuovo approccio di controllo predittivo robusto basato sull'insieme raggiungibile dai disturbi, che ha portato a uno schema di controllo che migliora le formulazioni precedenti in termini di sforzo computazionale, flessibilità e robustezza.

Parole chiave: robotica mobile, sistemi switched, model predictive control, MPC robusto, MPC stocastico.

Contents

1 | Introduction

Literature overview

Mobile robotics has a large variety of applications, which range from the more classical task of navigation and mapping to more complex tasks such as patrolling, search and rescue, and many others. Different types of models can describe wheeled mobile robots as unicycle, bicycle, or tractor-trailers. However, due to the complex nature of nonholonomic constraints, highly nonlinear control strategies generally have to be employed for even the most simple robot structure and simple environment in which no obstacles are present, see for instance [1]. This problem is based on the fact that the necessary conditions for the existence of time-invariant control law are violated [2]. The situation becomes even more difficult in the case of multiple robot systems, a scenario enabling the possibility to approach several other tasks as formation control [3, 4], flocking [5], coverage [6], aggregation [7]. Of course, under this scenario, different strategies are required for motion and coordination. This is even more evident once considered that often a centralized controller, that has all the available information at its disposal, is not suitable due to high computational demand or communication delay. To overcome this limitation, the use of other control architectures can be useful, such as decentralized controller, where each robot is controlled by an independent controller, disregarding possible connections between robots, or a distributed controller, where the exchange of information between controllers is allowed. Several approaches have been used to tackle the problem, as mentioned earlier. For instance, in [3], a feedback linearization approach exploiting only local information is proposed with the scope of stabilizing the robot in a specified formation, in [8] coordinated motion is obtained exploiting Lyapunov theory, in [9] the concept of control barrier function is exploited to achieve collision-free motion, while in [10] a behaviour based approach is exploited to carry out several different tasks, other approaches use distributed model predictive control (MPC) to perform several possible tasks such as formation control and coverage [11].

The flexibility of MPC also allows to easily consider a large variety of input constraints, such as the presence of a finite control set. There is indeed a strong connection between

a system with a discrete control set and switched systems, [12, 13], which are a particular class of hybrid system, whose evolution is governed by both continuous and discrete dynamics. Switched systems, and hybrid systems in general, have attracted a lot of attention in the last period due to the peculiarity of the behaviour of their dynamics and offer a suitable framework to model and analyze complex systems as the one considered in this thesis.

1.1. Comparison with respect to the related literature

Inspired by some recent work [14, 15] in which the self-aggregation problem is considered, in this thesis, an enhanced control strategy for trajectory tracking and aggregation of planar multi robot system in the presence of disturbances is considered. Exploiting the same setting in terms of robot motions, we consider a network of differential wheeled circular robots with constrained dynamic determined by two fixed pairs of wheel velocities allowing rotation on the spot and roto-translation describing a circular path. In this framework, the problem is recast in the context of switched system. In the literature, several works cover the issue of stabilization of switched systems, exploiting the concept of control Lyapunov function [16] or the use of Lyapunov-Metzler equation [13, 17] to obtain the switching laws. Differently, in this thesis, the "parking" problem is reformulated as an optimal control problem (OCP), where tracking and collision avoidance are considered. Then, the associated cost function is used as Lyapunov function for the switching MPC strategy.
Given the discrete nature of the actuation set, the optimal control problem is subject to a considerable computational complexity. This can be alleviated when considering the whole network in a distributed implementation, as in [11], where the original optimal control problem is divided into several smaller problems that are more easily manageable. Subsequently, it is shown how the developed strategy can be extended to other tasks such as path-following and self-aggregation with the use of an outer layer managing the reference of the switching MPC. The presence of disturbances acting on the robot position is also considered and the switching MPC strategy is modified, so as to guarantee a correct navigation of the environment despite the presence of disturbances.
At first, a robust formulation is considered which guarantees collision avoidance at the cost of obtaining a very conservative solution. To avoid the inherent conservativity present in the robust formulation, also a stochastic version of switching MPC is presented which makes use of chance-constraint on collision avoidance to obtain a trade-off between per-

formance and conservativity.
An analysis of the resulting switched model for a differential wheeled robot is carried out, highlighting the limitation of the techniques available in the literature for the case under study. Moreover, a novel robust MPC formulation is developed based on the disturbances reachable set. This strategy presents major computational advantages by addressing the switched nature of the system with an auxiliary control law, which reduces the optimal control problem to a simpler quadratic programming problem.

1.2. Main contribution

Spurred by the motivations and the challenges previously highlighted, the main contributions of this thesis with respect to the state of the art are the following:

- We have performed an analysis of the nonlinear switched model adopted to capture the robots dynamics, and we have designed different stabilizing switching laws, highlighting the current limitations in the literature;
- We have developed different stabilizing switching MPCs, able to handle the presence of uncertainty and disturbances in both robust and stochastic frameworks;
- We have proposed a completely original switching MPC approach based on disturbances reachable set, highlighting the main advantages and potential extension;
- All the approaches have been deeply analyzed from a theoretical viewpoint, also assessing their efficacy in realistic simulation scenarios.

1.3. Thesis outline

This thesis is organized as follows:

- In Chapter 2, the kinematic equations of a differential wheeled robot are derived, and the corresponding switched model defined by a switching signal is obtained; then the model is extended for a multi-robot system. To conclude the chapter, different models incorporating disturbances are introduced.
- In Chapter 3, an analysis of the obtained switched system is carried out. The limitations of available techniques present in the literature are pointed out; then, switching laws are obtained for both the continuous-time and sampled data systems. To conclude, an analysis of the effect of disturbances on the sampled domain control system is carried out.

- Chapter 4 deals with the optimal control formulation, taking into account avoidance objectives. Exploiting the optimal control problem, a switching MPC formulation is presented, and a distributed implementation is formulated; then the approach is extended to the robust and stochastic cases. Simulation results are shown and commented on for each of the developed algorithms.
- In Chapter 5, the reachability analysis described in Chapter 3 is exploited to develop a novel MPC approach that presents a major computational advantage. At first, a simple formulation is presented for both the centralized and distributed implementations; then a possible extension is developed.
- Finally, some conclusions are drawn, and future works are suggested in Chapter 6.

2 | Modelling of the system

This chapter presents the modelling of a differential wheeled mobile robot based on kinematic equations.

2.1. Kinematic modelling

A differential wheeled robot (see Figure 2.1) is composed of a passive castor wheel and two controlled wheels. This configuration allows the robot to move in a plane by properly defining the velocity of the two active wheels. In order to obtain a kinematic model for

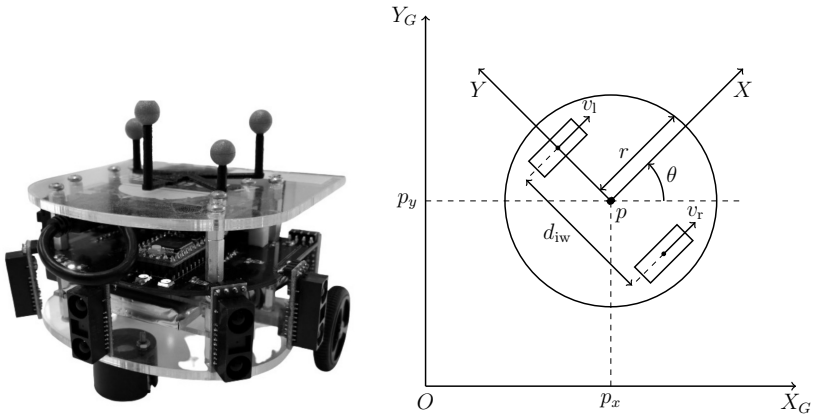


Figure 2.1: Differential wheeled mobile robot.

the considered configuration, the passive castor wheel is usually ignored. In the following, we shall consider d_{iw} as the distance between wheels, v_l and v_r as the linear velocities of the left and right wheels, respectively, θ as the robot orientation.

During the motion of the differential wheeled robot, it is usually assumed that the wheel of the robot rolls without slipping. Therefore, the motion of the wheel is constrained by the instantaneous center of rotation (ICR), which lies in the common lateral axis of the two active wheels.

Furthermore, during motion, there is a relationship between the angular speed of the

robot ω, the distance R of the ICR from the centroid p of the robot, and the linear wheel velocities v_l and v_r, i.e.,

$$\begin{aligned} \omega(R + \frac{d_{iw}}{2}) &= v_r \\ \omega(R - \frac{d_{iw}}{2}) &= v_l \end{aligned} \tag{2.1}$$

which can be rearranged as:

$$\begin{aligned} R &= \frac{d_{iw}}{w}(\frac{v_r + v_l}{v_r - v_l}) \\ \omega &= \frac{1}{d_{iw}}(v_r - v_l) \end{aligned} \tag{2.2}$$

From the previous equations, the linear velocity of the centroid p of the robot in the global frame of reference, namely v_{tx} and v_{ty}, can be computed as:

$$\begin{aligned} v = \omega R &= \frac{v_r + v_l}{2} \\ v_{tx} &= v\cos\theta \\ v_{ty} &= v\sin\theta \end{aligned} \tag{2.3}$$

With the previously computed relations, it is possible to obtain the kinematic model of the mobile robot. By considering as state variable the robot posture (position of the robot and orientation) in the global frame of reference $p = [p_x,\ p_y,\ \theta]^\top$ and a control action the linear and angular velocity of the robot $u = [v,\ \omega]^\top$ the time-invariant system model can be written as:

$$\begin{aligned} \dot{p}_x &= v\cos\theta \\ \dot{p}_y &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{2.4}$$

Systems of this type are known as non-holonomic systems, meaning that the system is subject to constraints involving both the state p and its derivative $\dot{p}$.
It has to be noted that v_t and ω are not the real inputs acting on the system, which are instead v_r and v_l. The value of v_r and v_l can be recovered by exploiting the relations:

$$\begin{aligned} v &= \frac{v_r + v_l}{2} \\ \omega &= \frac{1}{d_{iw}}(v_r - v_l) \end{aligned} \tag{2.5}$$

2.2. Switched model and network

Previously, the dynamics of a mobile robot have been defined. In this section, the particular constrained set of motion and the resulting switched dynamics are introduced.
In more detail, the robots will be constrained to a very restrictive set of possible motions as follows.

1. The first motion is defined by the relation $v_0 = 0$, $\omega = \omega_0$, which correspond in terms of linear velocities of the wheel to:

$$\begin{aligned} v_{r0} &= v_{l0} \\ \omega_0 &= \frac{1}{d_{iw}}(v_{r0} - v_{l0}) \end{aligned} \tag{2.6}$$

   Considering the relation for v_{t0} and ω_0 in the kinematic model (2.4) one has:

$$\begin{aligned} \dot{p}_x &= 0 \\ \dot{p}_y &= 0 \\ \dot{\theta} &= \omega_0 \end{aligned} \tag{2.7}$$

   Therefore, this first possible motion is characterized by only a rotation of the robot while maintaining its position unchanged.

2. The second motion is defined by $v = v_1$ and $\omega = \omega_1$, which correspond in terms of linear velocities of the wheel to

$$\begin{aligned} v_1 &= \frac{v_{r1} + v_{l1}}{2} \\ \omega_1 &= \frac{1}{d_{iw}}(v_{r1} - v_{l1}) \end{aligned} \tag{2.8}$$

   considering the relation for v_{t0} and ω_0 in the kinematic model (2.4) one has:

$$\begin{aligned} \dot{p}_x &= v_1 \cos\theta \\ \dot{p}_y &= v_1 \sin\theta \\ \dot{\theta} &= \omega_1 \end{aligned} \tag{2.9}$$

   which corresponds to a circular trajectory.

The considered set of motion composed by rotation and roto-translation could seem quite arbitrary, but it is chosen in analogy to [14, 15], where it is considered the problem of self-aggregation (also called consensus problem). In [15] an optimal controller is obtained

which exploits only the described set of motion; in [14] the switched formulation exploits the concept of control a Lyapunov function (see also [16]).
In analogy to these previous works, we define $[\ v_{l1}\ ,\ v_{r1}\ ,\ v_{l0}\ ,\ v_{r0}\ ] = [-0.7, -1, 1, -1]v_{max}$ where v_{max} is the maximum linear velocity of the wheel.
In order to recast the dynamics of the mobile robot as a nonlinear switched system, we need to define the switching signal $\sigma(t) \in \{0, 1\}$, so obtaining:

$$\dot{p}(t) = f_{\sigma(t)}(p(t)) \tag{2.10}$$

where $f_{\sigma(t)}$ belong to the set of vector field $\{f_0, f_1\}$:

$$f_0 = \begin{bmatrix} 0 \\ 0 \\ \omega_0 \end{bmatrix} \qquad f_1 = \begin{bmatrix} v_1 \cos\theta \\ v_1 \cos\theta \\ \omega_1 \end{bmatrix} \tag{2.11}$$

Furthermore, a network of N_{rob} robots can be considered, by defining the switching string $\Sigma(t) = \{\ \sigma_1(t)\ ,\ \sigma_2(t)\ \ldots,\ \sigma_{N_{rob}}(t)\}$, while the network dynamics can be modelled as:

$$\dot{p}(t) = \begin{bmatrix} \dot{p}^{[1]}(t) \\ . \\ . \\ \dot{p}^{N_{rob}}(t) \end{bmatrix} = \begin{bmatrix} f_{\sigma_1(t)}(p^{[1]}(t)) \\ . \\ . \\ f_{\sigma_{N_{rob}}(t)}(p^{[N_{rob}]}(t)) \end{bmatrix} = f_{\Sigma(t)}(p(t)) \tag{2.12}$$

It is relevant to note that the dynamics of the different robots are not coupled. In fact, the source of coupling between different robots resides in the constraint to be imposed on the evolution of the robot network.

2.3. Perturbed system

A relevant point of focus of the work is the analysis and design of a switching law in the presence of uncertainty and disturbances in the mobile robot dynamics. In particular, two different models for the uncertainty, one multiplicative and one additive, will be considered.
For the sake of simplicity, the perturbed model will be presented for a single mobile robot since the extension to a network of N_{rob} is quite trivial.
The first perturbed model, which will be considered, describes the uncertainty present in

the system as multiplicative:

$$\begin{aligned}\dot{p}_x &= ((1+d_1)\cos(\theta) - d_2\sin(\theta))v_\sigma \\ \dot{p}_y &= ((1+d_1)\sin(\theta) + d_2\cos(\theta))v_\sigma \\ \dot{\theta} &= (1+d_3)\omega_\sigma\end{aligned} \tag{2.13}$$

where d_1, d_2, d_3 represent perturbations related to the original system. In more detail, the term d_1 represents the difference between the wheels velocity and the linear velocity of the mobile robot due to effects such as the wheel slip, the term d_2 represents the lateral velocity caused by skidding of the mobile robot, the term d_3 represents the deviation from the nominal angular velocity ω_σ. For the remainder of the work, the term d_3 will be considered null or negligible since it is possible to translate its effect in the overall effect of d_1 and d_2.
A possible sensible assumption is to consider the terms d_1 and d_2 bounded and uniformly distributed: $|d_i| < \bar{d} < 1$ for $i = 1, 2$. These bounds reflect the fact that a magnitude larger than 1 means that the perturbation would have a greater effect than the input applied, thus making it impossible to modify the behavior of the mobile robot. Another possible assumption is to consider a norm-bounded perturbation: $d_1^2 + d_2^2 < \bar{d}^2 < 1$. Furthermore, it is also possible to consider the multiplicative uncertainty as a mode-dependent disturbance acting on the system, i.e.,

$$\begin{aligned}\dot{p}_x &= v_\sigma\cos(\theta) + d_x(\sigma) \\ \dot{p}_y &= v_\sigma\sin(\theta) + d_y(\sigma) \\ \dot{\theta} &= \omega_\sigma\end{aligned} \tag{2.14}$$

where:

$$\begin{aligned}d_x(\sigma) &= (d_1\cos(\theta) - d_2\sin(\theta))v_\sigma \\ d_y(\sigma) &= (d_1\sin(\theta) + d_2\cos(\theta))v_\sigma\end{aligned} \tag{2.15}$$

Under the assumption $d_1^2 + d_2^2 < \bar{d}^2$ it is possible to show that:

$$\begin{aligned}&d_x^2(\sigma) + d_y^2(\sigma) = v_\sigma^2(d_1^2\cos^2(\theta) - 2d_1d_2\cos(\theta)\sin(\theta) \\ &+ d_2^2\sin^2(\theta)d_1^2\sin^2(\theta) + 2d_1d_2\cos(\theta)\sin(\theta) + d_2^2\cos^2(\theta)) \\ &d_x^2(\sigma) + d_y^2(\sigma) = v_\sigma(d_1^2 + d_2^2) < v_\sigma\bar{d}\end{aligned} \tag{2.16}$$

It is relevant to note that, considering a multiplicative description of the uncertainty acting on the system, implies that the effect of the uncertainty in the case of $\sigma(t) = 0$ is null. This could be a stringent condition on the disturbances acting on the system since it

prohibits the presence of external influences on the system dynamics, which can be taken into account by allowing the additive disturbances as follows:

$$\begin{aligned} \dot{p}_x &= v_\sigma \cos(\theta) + d_x \\ \dot{p}_y &= v_\sigma \sin(\theta) + d_y \\ \dot{\theta} &= \omega_\sigma \end{aligned} \tag{2.17}$$

The difference between (2.14) and (2.17) resides in the fact that in (2.17) the disturbances are independent from the value of the switching signal σ. Furthermore, if instead of using p_x and p_y as state variables, we consider the deviation e_x and e_y from a time-varying reference position, the same model (2.17) is obtained, where the additive disturbances can be interpreted as the rate of change with respect to the desired reference position.
So far, only the continuous time models have been considered. Discretized models will be covered in later chapters, while discussing the receding-horizon control approach that is the main focus of the present work.

3 | Switched system analysis

In this chapter, the analysis of the dynamics of a single robot is carried out. The result will be useful to design and assess the performance of the control law, which will be proposed in the later chapter. Furthermore, some general considerations on the stabilization of nonholonomic systems are presented.

3.1. Preliminaries

We start by recalling the original model of the system:

$$\begin{aligned} \dot{p}_x &= v\cos\theta \\ \dot{p}_y &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{3.1}$$

In order to further motivate the adopted switched approach, it is relevant to recall the well-known Brockett's condition [2]

Theorem 3.1. *Consider the control system $\dot{x} = f(x,u)$ with $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{U} = \mathbb{R}^m$, and suppose that there exists a continuous feedback law $u = k(x)$ satisfying $k(0) = 0$ which makes the origin a (locally) asymptotically stable equilibrium of the closed-loop system $\dot{x} = f(x,k(x))$. Then, the image of every neighborhood of (0, 0) in $\mathbb{R}^n \times \mathbb{R}^m$ under the map*

$$(x,u) \to f(x,u)$$

contains some neighborhood of zero in $\mathbb{R}^n$.

It is easy to show that system (3.1) does not satisfy Theorem 3.1. Therefore, it is not possible to stabilize the system with a continuous control law. This result further motivates the use of a switched modelling of the system, being the switching signal σ intrinsically associated with a discontinuous control action. This discussion is more relevant for the continuous time case than for the discrete time one. For a general survey on switched systems that also covers the non-holonomic case, see [18].

3.2. Coordinate transformation

In order to simplify the analysis of the system (3.1) a change of coordinate is proposed:

$$\begin{aligned} z_1 &= p_x \sin(\theta) - p_y \cos(\theta) \\ z_2 &= p_x \cos(\theta) + p_y \sin(\theta) \\ z_3 &= \theta \end{aligned} \tag{3.2}$$

The proposed change of coordinates corresponds to a rotation of the reference of the frame to the axis oriented as the mobile robot. It is relevant to note, that being a rotation, the distance from the origin is preserved.

In the new set of coordinates, the system dynamics of the system can be easily computed as:

$$\begin{aligned} \dot{z}_1 &= \dot{p}_x \sin(\theta) + p_x \cos(\theta)\dot{\theta} - \dot{p}_y \cos(\theta) + p_y \sin(\theta)\dot{\theta} \\ &= v_\sigma(\cos(\theta)\sin(\theta) - \cos(\theta)\sin(\theta)) + \omega_\sigma(p_x \cos(\theta) + p_y \sin(\theta)) = \omega_\sigma z_2 \\ &= \dot{p}_x \cos(\theta) - p_x \sin(\theta)\dot{\theta} + \dot{p}_y \sin(\theta) + p_y \cos(\theta)\dot{\theta} \\ \dot{z}_2 &= v_\sigma(\cos^2(\theta) + \sin^2(\theta)) + \omega_\sigma(-p_x \sin(\theta) + p_y \cos(\theta)) = -\omega_\sigma z_1 + v_\sigma \\ \dot{z}_3 &= \dot{\theta} = \omega_\sigma \end{aligned} \tag{3.3}$$

Note that the state variable z_3 does not affect the dynamics of the subsystem composed by the state $[z_1,\ z_2]$, furthermore the equilibrium $\bar{p} = [0,\ 0, \theta]^\top$ is mapped into $[\bar{z}_1,\ \bar{z}_2] = [0,\ 0]$, therefore the problem of Cartesian regulation it is equivalent to the regulation of the subsystem $[z_1,\ z_2]$.
Overall, the evolution of the reduced state $z = [z_1,\ z_2]^\top$ is governed by the switched affine system (SAS):

$$\dot{z} = A_\sigma z + B_\sigma \tag{3.4}$$

where:

$$A_\sigma = \begin{bmatrix} 0 & \omega_\sigma \\ -\omega_\sigma & 0 \end{bmatrix} \quad B_\sigma = \begin{bmatrix} 0 \\ v_\sigma \end{bmatrix} \tag{3.5}$$

There is a wide literature regarding the stabilization of this class of systems, see, for instance, [19–21]. Despite all these contributions, the applicability of general methods developed for SAS is limited due to some not respected assumptions.
For instance, some methods exploit the existence of an Hurwitz[1] convex combination of the matrices A_σ, i.e. the existence of non-negative real numbers α_i summing to 1, such that $A_\alpha = \sum_i^N \alpha_i A_i$ has all eigenvalues with real part strictly less than 0. In our case

[1] A square matrix A is said to be Hurwitz if every eigenvalue of A has strictly negative real part, i.e. $Re\,[\lambda_i] < 0$ for each eigenvalue λ_i of A.

$N = 2$ and it is easy to see that for our A_i there is no Hurwitz convex combination:

$$A_\alpha = \alpha A_0 + (1-\alpha)A_1 = \begin{bmatrix} 0 & \omega_1 + \alpha(\omega_0 - \omega_1) \\ -\omega_1 - \alpha(\omega_0 - \omega_1) & 0 \end{bmatrix} \quad \alpha \in [0,1] \tag{3.6}$$

For all admissible values of α, the matrix A_α presents a couple of pure imaginary eigenvalues.
Other methods consider the linear and affine dynamics separately and revolve around the stabilization of the linear switched system $\dot{z} = A_\sigma z$ exploiting methods such as Lyapunov-Metzler function (see [12, 17] for a general survey about switched system).
Despite this limitation, it will be shown that it is still possible to find switching rules that asymptotically stabilize the system.
In later section also the design of switching laws for the discretized version of model (3.4), letting T be the sampling time and assuming that σ can change only at instants multiple of T one can write:

$$\begin{gathered} z(k+1) = A_{d\sigma} z(k) + B_{d_\sigma} \\ A_{d\sigma} = e^{(A_\sigma T)} \qquad B_{d\sigma} = \int_0^T e^{A_\sigma \tau} B_\tau d\tau \end{gathered} \tag{3.7}$$

where:

$$A_{d\sigma} = \begin{bmatrix} \cos(\omega_\sigma T) & \sin(\omega_\sigma T) \\ -\sin(\omega_\sigma T) & \cos(\omega_\sigma T) \end{bmatrix} \qquad B_{d\sigma} = \begin{bmatrix} \frac{-v_\sigma(\cos(\omega_\sigma T)-1)}{\omega_\sigma} \\ \frac{v_\sigma \sin(\omega_\sigma T)}{\omega_\sigma} \end{bmatrix} \tag{3.8}$$

A similar argument to the continuous-time case can be made, and similar limitations arise, see for instance [22, 23] for some results about SAS in discrete time, and [13] for some general results about discrete-time switched systems.

3.3. Continuous-time analysis

In order to discuss the design of a stabilizing switching law for the system (3.4), we need to recall that, in analogy with [14, 15] the wheel velocities of the mobile robot in the mode associated with $\sigma = 0$ are $[v_{l0},\ v_{r0}] = [1,\ -1]$ while for the mode associated with $\sigma = 1$ are $[v_{l1},\ v_{r1}] = [-0.7,\ -1]$.Therefore, the parameter of (2.4), (2.7) and (2.9) are such that:

$$v_0 = 0, \quad v_1 < 0, \quad \omega_0 < 0, \quad \omega_1 < 0, \tag{3.9}$$

With these particular conditions on the parameters, the trajectories associated with $\sigma = 0$ are counterclockwise circular trajectories around the origin, while the trajectories associated with $\sigma = 1$ are counterclockwise circular trajectories around the equilibrium point

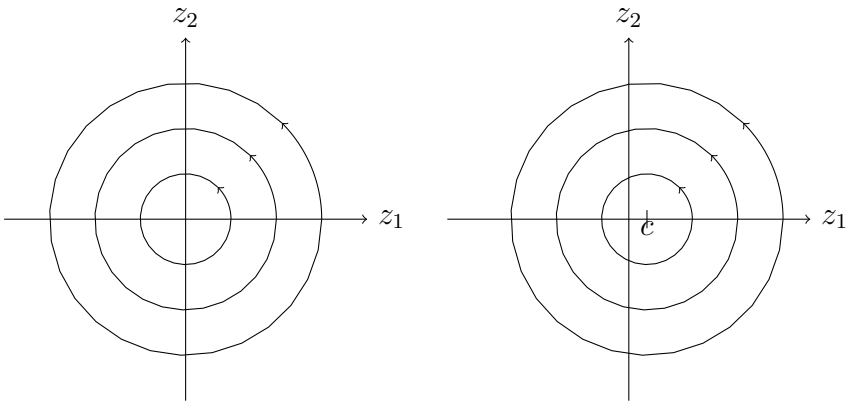


Figure 3.1: Qualitative description of the trajectories in mode 0(left) and 1 (right) $c = \frac{v1}{\omega_1}$

($\frac{v_1}{\omega_1}$, 0), as qualitatively shown in Figure 3.1. There are many stabilizing switching laws, the one that will be considered here has been chosen since it avoids sliding motions, and analysis is more streamlined.

Consider now the switching law:

$$\sigma = \begin{cases} 1 & if\ z_1 < 0\ \wedge\ z_2 > 0 \\ 0 & otherwise \end{cases} \tag{3.10}$$

Where $\wedge$ represent the logical AND.

Theorem 3.2. *the switching law* (3.10) *makes the origin of* (3.4) *a globally asymptotically stable equilibrium point of the closed loop system* (3.4).

Proof. As mentioned above, proving stability by exploiting Lyapunov function is not easy for the system under study. This fact is due to difficulties in obtaining a monotonically decreasing Lyapunov function. However, it is possible to evaluate the Lyapunov function at each switching time. In fact, if for every pair of switching times $(k, k+1)$, it holds that $V(t(k)) - V(t(k)) < 0$, stability is ensured.

To this aim, consider the distance from the origin as a candidate Lyapunov function

$$V(t(k)) = \sqrt{z_1(t(k))^2 + z_2(t(k))^2} \tag{3.11}$$

Assume for simplicity that the initial states belong to the switching surface $z_1(0) < 0$,$z_2(0) = 0$ and Lyapunov function $V(t(0))$. Therefore, the trajectories start on the negative z_1 axis. Then trajectory enters the third quadrant, where the active subsystem is $\sigma = 0$. When $\sigma = 0$ the trajectories follow a circular path around the origin, therefore

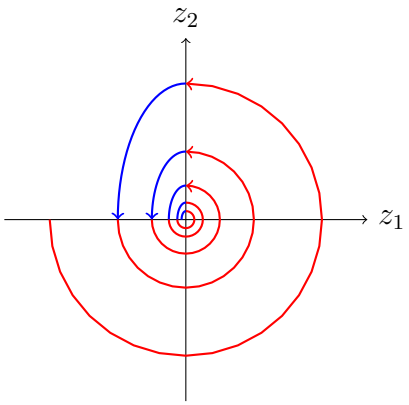


Figure 3.2: Qualitative behavior of the closed loop system, in red mode 0 is active in blue mode 1 is active

$\dot{V} = 0$ and the trajectory crosses the switching surface associated with the positive z_2 with the same value for the Lyapunov function $V(t(0))$. When the trajectory enters the second quadrant, where the active subsystem is $\sigma = 1$, it is easy to show by simple geometrical consideration that the Lyapunov function evaluated at the switching surface $z_1 < 0, z_2 = 0$ will be

$$V(t(k+1)) = \sqrt{V(t(k))^2 + \left(\frac{v_1}{\omega_1}\right)^2} - \left(\frac{v_1}{\omega_1}\right) \tag{3.12}$$

Therefore, being $V(t(k+1)) - V(t(k)) < 0$ asymptotical stability of the origin is ensured.

This result will not be used in later chapters, but it is useful to characterize the attainable performance of the system. In fact, it shows that if infinitely fast switching is admissible, it is possible to arbitrarily regulate the Cartesian position of the mobile robot. This result has been validated through simulation. Physical parameters for the simulation can be found in Table 3.1.

	Value
Robot radius	0.055 m
Inter-wheel distance	0.105 m
Wheel radius	0.016 m
Maximum linear velocity	0.2 m/s

Table 3.1: Simulation parameter

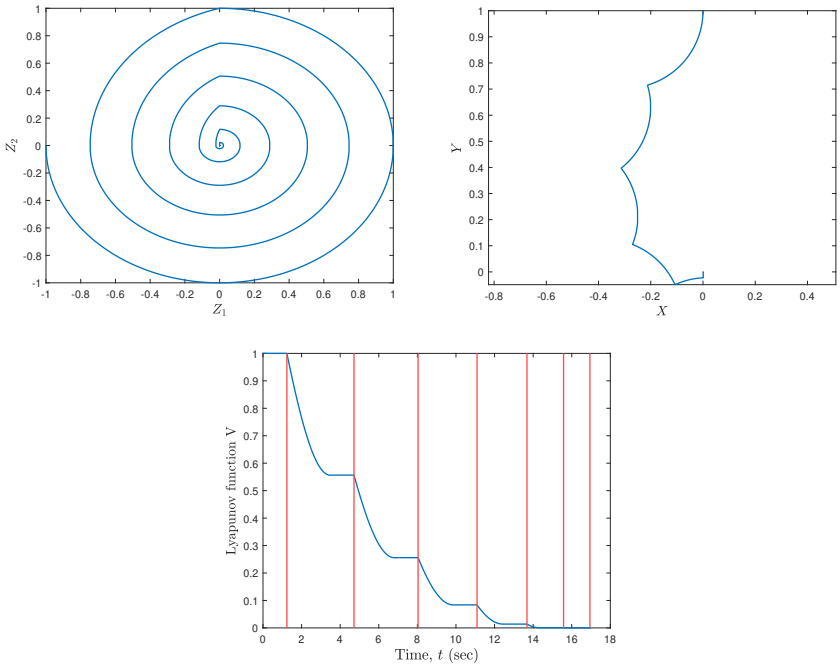


Figure 3.3: trajectory in z_1, z_2 coordinates (top-left), trajectory in the original coordinates p_x, p_y (top-right), Lyapunov function with red lines highlighting the time instants where (3.12) is evaluated (bottom)

To conclude this section it has to be noted that also Sum-of-Square (SOS) optimization, see for a general survey [24], can be considered to obtain a better switching law and a Lyapunov function exploiting the Lyapunov-Metzler inequality. Since the combination of Sum-of-Square optimization and Lyapunov-Metzler inequality has not produced feasible results, it will not be discussed.

3.4. Sampled-data analysis

In this section, analysis of the discretized version of the system will be carried out, in terms of stabilizability of the unperturbed model and disturbance rejection of the designed control law when perturbations are present.
Even in an ideal scenario, it is not possible to design a switching law for the system (3.7) capable of asymptotically stabilizing the origin of the system. It is instead common for this type of system to consider asymptotical practical stability (also called ultimate boundedness). In short, asymptotical practical stability refers to the convergence of the system trajectory not to an equilibrium point but to a neighborhood of it.
In a sampled-data setup, the presence of sliding motion is not a concern with respect to the continuous time case. For this reason, we consider the use of control Lyapunov

function (CLF) for the design of switching law. For the design of a switching law, we will consider the candidate CLF:

$$V(z) = z_1^2 + z_2^2 \tag{3.13}$$

The computation of $\Delta V_\sigma = V(f_\sigma(z_1, z_2)) - V(z_1, z_2)$, after some computation, leads to

$$\Delta V_\sigma = \frac{2v_\sigma(z_2 \sin(T\omega_\sigma) - 2z_1 \sin^2(\frac{T\omega_\sigma}{2}))}{\omega_\sigma} + \frac{4v_\sigma^2 \sin^2(\frac{T\omega_\sigma}{2})}{\omega_1^2} \tag{3.14}$$

Theorem 3.3. *Assume that* $T \neq \frac{n2\pi}{\omega_0}$ *with* $n \in \mathbb{N}$. *Then, switching law:*

$$\bar{\sigma} = \underset{\sigma}{argmin} \quad \Delta V_\sigma$$

ensure global practical asymptotical stability.

Proof. Since $v_0 = 0$, the switching law ensures that $\Delta V_{\bar{\sigma}} \leq 0$, therefore V tends to a limit as k tend to infinity. This result is not enough to verify asymptotical practical stability. However, it is possible to prove the convergence of the trajectory to a ball centered in the origin of radius R_c, where R_c is the distance of the switching surface to the origin, via LaSalle type reasoning.

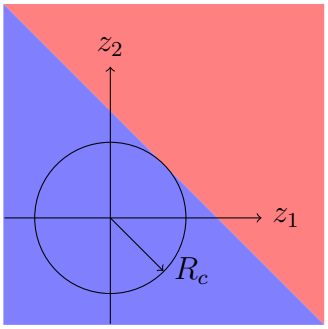


Figure 3.4: Region in red represent active mode 0 while in blue represent active mode 1

Trajectories have to converge to a set where $\Delta V_{\bar{\sigma}} = 0$, depicted in blue in Figure 3.4. Since all trajectories outside the ball of radius R_c eventually end in the set associated with $\Delta V_{\bar{\sigma}} < 0$, in red in Figure 3.4. Therefore, trajectories have to converge to the ball around the origin of radius:

$$R_c = \frac{2v_1 \sin^2(\frac{T\omega_1}{2})}{w_1^2 \sqrt{\frac{\sin(T\omega_1)}{w_1^2} + \frac{v_1 \sin^2(T\omega_1)}{\omega_1^2}}} \tag{3.15}$$

The assumption $T \neq \frac{n2\pi}{\omega_0}$ is required to avoid sampling time for which $A_{d0} = I$, which renders the system not stabilizable.
To conclude this section, simulation results for the switching law (3.3) are shown. The physical parameter can be found in Table 3.1 while the considered sampling time is $T = 0.1$.

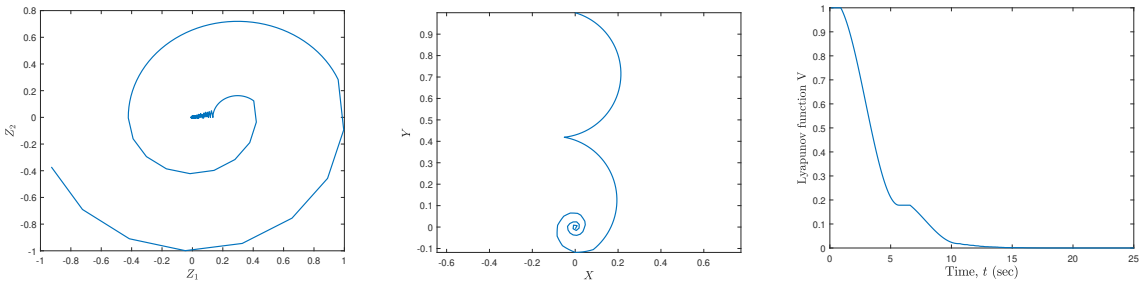


Figure 3.5: left: trajectory in z_1, z_2 coordinate ;center: trajectory in x, y coordinate; right: Lyapunov function

3.5. Perturbed case

In this section, the performance of the switching law of Theorem (3.3) will be carried out, by considering the presence of persistent disturbances. We start by recalling the model (2.17), with the change of variable $e_x = p_x - \bar{p}_x, e_y = p_y - \bar{p}_y$, where $\bar{p}_x, \bar{p}_y$ are the desired position of the robot. In particular, we will consider the exact discretization of the model with sampling time T:

$$\begin{aligned} e_x(k+1) &= e_x(k) + v_\sigma \frac{\sin(\theta(k) + T\omega_1) - \sin(\theta(k))}{\omega 1} + \bar{d}_x(k) \\ e_y(k+1) &= e_y(k+1) + v_\sigma \frac{-\cos(\theta(k) + T\omega_1) + \cos(\theta(k))}{\omega_1} + \bar{d}_y(k) \\ \theta(k+1) &= \theta(k) + \omega_\sigma T \end{aligned} \tag{3.16}$$

It is evident that the disturbances $\bar{d}_x, \bar{d}_y$ represent both external disturbances acting on the system and change of the desired reference position.

$$\begin{aligned} \bar{d}_x(k) &= d_x(k) + \Delta\bar{p}_x(k) \\ \bar{d}_y(k) &= d_y(k) + \Delta\bar{p}_y(k) \end{aligned} \tag{3.17}$$

In the following, it will be assumed that the disturbances $\bar{d}_x, \bar{d}_y$ are norm-bounded i.e. $\bar{d}_x^2 + d_y^2 \leq \bar{D}_M^2$.

Performing the change of coordinate (3.2) from (3.16) one has

$$z(k+1) = A_{d\sigma} z + B_{d\sigma} + D_z \tag{3.18}$$

Where $A_{d\sigma}, B_{d\sigma}$ are the matrix in (3.5) and $D_z = \begin{bmatrix} d_{z1} & d_{z2} \end{bmatrix}^\top$. Recalling that the change of coordinate is equivalent to a rotation of the original coordinates the disturbances d_{z1}, d_{z2} are bounded and satisfy:

$$d_{z1}^2 + d_{z2}^2 \leq \bar{D}_M^2 \tag{3.19}$$

This constraint on the admissible value of the disturbances can be outer approximated with a polytope. Details on this approximation will be given later.

3.5.1. 0-Reachability of closed-loop system

Considering the switching law of Theorem 3.3, the closed-loop system assumes the form

$$z(k+1) = A_i z(k) + B_i + D_z \quad for \quad z(k) \in \Omega_i \tag{3.20}$$

Where Ω_i is such that $\bigcup \Omega_i = \mathbb{R}^n$. Systems of this form are called piecewise affine system (PWA) and are a common model used to describe hybrid and nonlinear systems, see [25]. In this section, we are interested in assessing the effect of the disturbance through the computation of the 0-Reachable set $\mathcal{R}$ of the system, which represents the set of reachable states with the origin as the initial state.
Before discussing the reachability analysis for PWA system, it is useful to consider the simpler case of linear systems, such as:

$$x(k+1) = Fx(k) + Ed(k) \tag{3.21}$$

Denote by $\mathcal{R}_T$ the set of reachable states in T steps and $d \in \mathcal{D}$. $\mathcal{R}_T$ is given by:

$$\mathcal{R}_T = F\mathcal{R}_{T-1} + E\mathcal{D} \tag{3.22}$$

This involves an operation such as a sum of sets and the image of a set. Given $\mathcal{D}$ it is convenient to use a polyhedral set since they offer a simple description of the reachable set in terms of vertices and convex hull, see [26]. In principle, it is possible to compute in

an approximate way the infinite time reachability set as:

$$\mathcal{R}_\infty = \bigcup_{i=1}^{n} \mathcal{R}_i \tag{3.23}$$

By using a large k, this set would be an internal approximation of $\mathcal{R}_\infty$. However, it is possible to prove that there exists an integer $\bar{k}$ for all $\epsilon > 0$ such that for $k \geq \bar{k}$:

$$\mathcal{R}_k \subseteq \mathcal{R}_\infty \subseteq (1+\epsilon)\mathcal{R}_k \tag{3.24}$$

To achieve an external approximation several solutions are possible. One is to "enlarge" the disturbance set $\mathcal{D}$.
Although reachability analysis for PWA system is conceptually similar to the linear case, it is considerably more difficult due to the hybrid nature of the system. Many approaches are present in the literature which exploits s collection of polyhedra [27], hyperboxes [28], zonotope [29]. When the evolution of the system remains in a single region, it is easy to evaluate the reachable set; however, when the reachable set intersects a new region, a new reachable set computation has to be performed from the new intersection. Due to the low number of states and low number of regions of the system under study, a simple algorithm based on the propagation of a collection of polyhedra can be devised.

Algorithm 3.1 Reachable set computation

H

1: **Input:**Initial set of state $\mathcal{R}_0$, set of admissible input $\mathcal{U}$
2: **Output:** Set of reachable states $\mathcal{R}_\infty$
3: Divide $\mathcal{R}_0$ into different polyhedra according to the region Ω
4: **while** $k < k_{max}$
5: Compute the reachable set in 1 step $\mathcal{S}$ from $\mathcal{R}_k$ coherently with the partition of the system
6: Detect intersection and compute set difference to split polyhedra coherently with the partition of the system.
7: Delete redundant polyhedra
8: Define the union of polyhedra $\mathcal{R}_{k+1} = \mathcal{R}_k \bigcup \mathcal{S}$
9: **if** $\mathcal{R}_{k+1} \subseteq \mathcal{R}_k$
10: **break**
11: **end**
12: **end**
13: **Return** $\mathcal{R}$

The algorithm is not particularly scalable in terms of number of states and regions, for

which it is better to exploit algorithms with more efficient structures during set propagation.
Of particular importance is the quantity R_M defined as:

$$R_M = \max\{r|\ [z_1, z_2] \in \mathcal{R}, z_1^2 + z_2^2 \leq r^2\} \tag{3.25}$$

That represents the maximum deviation from the origin considering the defined set of disturbances. Therefore, once a switching rule $\sigma(p, \bar{p})$ and the corresponding R_M are computed for a given set of disturbances $\mathcal{D}$, it is possible to ensure that if the reference trajectory is compatible with the set $\mathcal{D}$ the maximum deviation from the desired trajectory is R_M.This result will be fundamental for the algorithms that will be developed in Chapter 5. It can be seen that the discussion about reachability could be extended in the stochastic framework; however, it becomes considerably more difficult [30, 31].

3.5.2. Results

In Figure 3.6 it is depicted the reachable space for different bounds on the disturbances acting on the system with an upper bound proportional to the maximum speed of the robot, with a constant of proportionality α, i.e., $D_M = a_1 T \alpha$. Considering the switching law (3.10) and a sampling time $T = 0.033$.

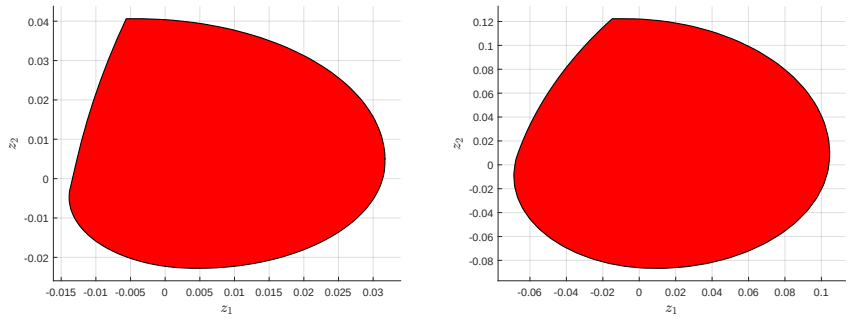


Figure 3.6: Reachable space for continuous-time switching law;left $\alpha = 0.1$, right $\alpha = 0.2$

With the considered sampling time the switching law (3.10) is able to ensure robust practical stability for a sufficiently small value of α. This result has been validated using simulation for several values of maximum disturbances allowed.
Figure 3.7 depicts the reachable space when the switching law (3.3) is used with a sampling time $T = 0.1$.

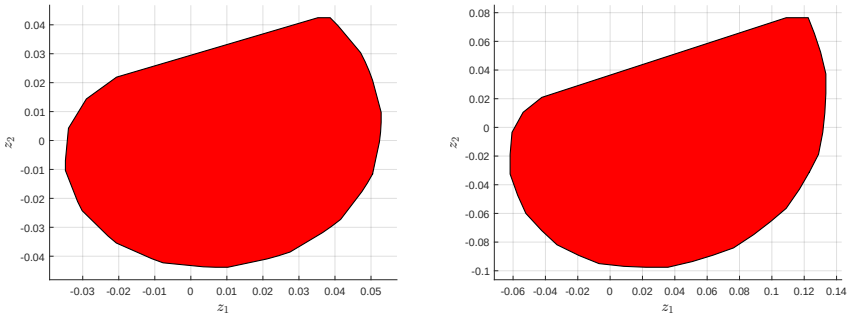


Figure 3.7: Reachable space for discrete-time switching law;right $\alpha = 0.1$,left $\alpha = 0.4$

It is relevant to note that it is possible to obtain an approximate linear upper bound for the relationship between the maximum allowed disturbance value D_M and the maximum deviation from the origin R_M.

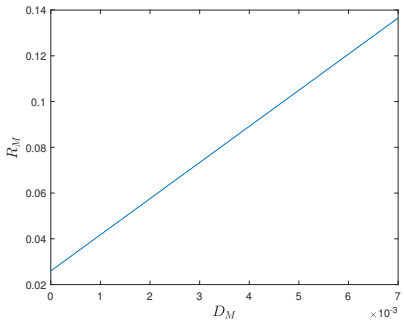


Figure 3.8: Approximated upper bound on maximum deviation

With the parameter considered for the mobile robot, see Table 3.1. The linear upper bound on the maximum deviation can be expressed as:

$$R_M(D_M) = K_1 D_M + K_2 \tag{3.26}$$

where $K_1 = 15.8$ and $K_2 = 0.026$. In order to implement the algorithm, the Matlab MPT toolbox has been exploited [32].

4 | Switching MPC

In this chapter, different MPC algorithms and relative finite-horizon optimal control problems will be presented for both the nominal and perturbed cases. The main focus will be on the so-called parking problem, which consists in the regulation of the Cartesian position of each robot. Furthermore, some other possible tasks (consensus, formation control) will be taken into account in the unperturbed case, mainly to highlight the flexibility of the proposed approach.

4.1. Switching finite horizon optimal control problem

The switching-MPC strategy exploits a system model to predict the trajectories of the system for each switching signal, while the control objectives and constraints are embedded in a finite-horizon optimal control problem (FHOCP) to be solved at each time instant. In this section the various components of the FHOCP will be presented. For a general introduction of MPC approaches, it is possible to refer to [33, 34] or a more recent survey [35].

4.1.1. Model discretization

In order to be able to predict the trajectories of each mobile robot, it is necessary to derive a discrete time model of the mobile robot i with sampling time T. Since the computational complexity of the optimization problem is not really dependent on the complexity of the model, an exact discretization of model (2.10) will be considered:

$$p^{[i]}(k+1) = f_{\sigma(k)}(p^{[i]}(k)) \tag{4.1}$$

where:

$$f_0 = \begin{bmatrix} p_x^{[i]}(k) \\ p_y^{[i]}(k) \\ \theta^{[i]}(k) + T\omega_0 \end{bmatrix} \quad f_1 = \begin{bmatrix} p_x^{[i]}(k) + v_1 \frac{\sin(\theta^{[i]}(k)+T\omega_1)-\sin(\theta^{[i]}(k))}{\omega 1} \\ p_y^{[i]}(k) + v_1 \frac{-\cos(\theta^{[i]}(k)+T\omega_1)+\cos(\theta^{[i]}(k))}{\omega_1} \\ \theta^{[i]}(k) + T\omega_1 \end{bmatrix} \tag{4.2}$$

Once the discrete model for a single robot is obtained, the discrete network model can be derived analogously to (2.12) as:

$$p(k) = \begin{bmatrix} p^{[1]}(k) \\ \cdot \\ \cdot \\ p^{N_{rob}}(k) \end{bmatrix} = \begin{bmatrix} f_{\sigma_1(k)}(p^{[1]}(k)) \\ \cdot \\ \cdot \\ f_{\sigma_{N_{rob}}(k)}(p^{[N_{rob}]}(k)) \end{bmatrix} = f_{\Sigma(k)}(p(k)) \tag{4.3}$$

Furthermore, regarding the simulations presented in this chapter, the parameter in table 3.1 will be considered

4.1.2. Cost function

Cost functions are fundamental to define the objective of an optimal control problem. Since for the case under study we are considering a reference tracking problem over a prediction horizon of length N_p and N robot, a quite natural expression to consider for the parking problem is :

$$J = \sum_{k=0}^{N_p} l(p(k), \bar{p}) = \sum_{k=0}^{N_p} \sum_{i=1}^{N_{rob}} (p_x^{[i]} - \bar{p}_x^{[i]})^2 + (p_y^{[i]} - \bar{p}_y^{[i]})^2 \tag{4.4}$$

where N_p is the prediction horizon, N is the number of robot,$\bar{p}$ is the reference signal for the $i-th$ robot.
A relevant property of the cost function is its separability in the N subproblems:

$$\begin{aligned} J &= \sum_{i=1}^{N} \bar{J}^{[i]} \\ \bar{J}^{[i]} &= \sum_{k=0}^{N_p} \bar{l}(p^{[i]}(k), \bar{p}^{[i]}) \\ \bar{l}(p^{[i]}(k), \bar{p}^{[i]}) &= (p_x^{[i]}(k) - \bar{p}_x^{[i]})^2 + (p_y^{[i]}(k) - \bar{p}_y^{[i]})^2 \end{aligned} \tag{4.5}$$

This fact will be fundamental to develop a distributed implementation of the proposed approach.

4.1.3. Constraint definition

In the following, some constraints will be taken into account in order to avoid collisions among agents and avoid obstacles in the environment.

Obstacle avoidance

During navigation, each robot must avoid collision with an obstacle present in the environment. Without loss of generality, considering a circular obstacle of radius R_{obs}, center p^{obs}, we also consider R_{rob} as the robot radius. The obstacle avoidance constraint for the i-th robot can be formulated as:

$$\left\| \begin{bmatrix} p_x^i(k) - p_x^{obs} \\ p_y^i(k) - p_y^{obs} \end{bmatrix} \right\|_2 > R_{rob} + R_{obs} \tag{4.6}$$

This type of constraint is nonlinear, but it can be approximated as a set of n linear inequalities as described in [11]. To approximate the constraint, it is possible to consider the circular obstacle with its outer polytopic approximation. Once an outer polytopic approximation is obtained, it will be possible to check for collision by evaluating n inequalities of the type:

$$H_n p^{[i]}(k) \geq S_n \tag{4.7}$$

If at least one of the inequalities is satisfied, the robot i lies in the exterior of the polytope, and the collision is avoided, see Figure 4.1. Note that given the discrete nature of the optimization problem, it is not an issue to check if at least one inequality it is satisfied. Therefore, selecting $\bar{n} = \underset{[1\ldots n]}{argmin} \left(H_n p^{[i]}(k) - S_n \right)$ it is possible to enforce obstacle avoidance with a linear constraint by including in the optimization the constraint

$$H_{\bar{n}} p^{[i]}(k) \geq S_{\bar{n}} \tag{4.8}$$

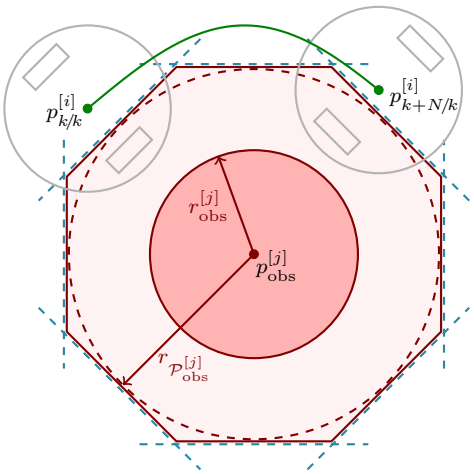


Figure 4.1: Polytope $\mathcal{P}^{[j]}_{\text{obs}}$ (blue dashed lines) associated to the jth obstacle (red circle). The pink area represents the linear approximation of the dashed red circle, while the green line depicts the predicted trajectory of the ith robot (gray circle)

This procedure in principle has to be applied to all robot-obstacle pairs. However, it is convenient to consider constraints only between the robot and obstacle which are "proximal", in order to reduce the complexity of the optimization problem. This can be done by considering an obstacle 'proximal' if it lies at a distance not greater than a predefined distance R_p. To construct the approximation, we can exploit the following algorithm:

Algorithm 4.1 Outer polytopic approximation of a circular obstacle

1: **Input:** Radius *Robs* ,center P_{obs} of the obstacle, number of edges n of the polytopic approximation and radius of the robot R_{rob}
2: **Output:** Matrix H and vector S defining the edge of the outer polytopic approximation
3: **For:**i=0:N-1
4: $\theta_i = \frac{i\pi}{n}$
5: $H_i = [\cos(\theta_i) \quad \sin(\theta_i)]$
6: $S_i = R_{obs} + [\cos(\theta_i) \quad \sin(\theta_i)]p_{obs}$
7: **end**
8: **Return** H,S

The number of edges of the polytope cannot be less than 3 (in order to obtain a closed space), and the quality of the approximation increases with the number of edges at the expense of complexity. From some experiments in simulation and as described in [11] a number of edges equal to 20 allows to achieve a good approximation for the desired scope. It has to be noted that the described procedure does not offer an advantage in case no uncertainties are present, since the complexity of the optimization problem to be solved is not affected by the nonlinear nature of the constraint. However, this reformulation is useful to obtain tightened constraints in case of uncertainty and is also useful to handle inter-robot collision avoidance in distributed implementation. Another possible approximation could have exploited hyper-rectangle as in [36].
The proposed procedure could be generalized to a generic convex obstacle and the error between the approximation and the original obstacle quantified. Since for the case under study, we consider only circular obstacles, and the complexity of the approximation is determined mainly by computational limitation, no further detail will be discussed.

Collision avoidance

The collision avoidance problem in the case of a centralized controller can be reformulated in a similar way as the obstacle avoidance problem. For a centralized implementation, it is possible to consider a robot as an obstacle with a time-varying position. Therefore,

by exploiting the same procedure for obstacle avoidance (with slight modifications due to the time-varying nature of the constraint), it is possible to construct constraints for inter-robot collision avoidance with the matrix $H_n(k)$ and $S_n(k)$. In practice, for a generic trajectory the following algorithm has to be executed to generate the constraint:

Algorithm 4.2 Collision avoidance constraint

1: **Input:** Trajectory $p^{[i]}(k|\bar{k})$ of robot i
2: **Output:**Matrix $H(k)$ and $S(k)$ which define the collision avoidance constraint to be satisfied.
3: **For**i=$\bar{k}$:$\bar{k} + N_p$
4: Execute algorithm 4.1 taking as input R_{rob} for the obstacle radius and $p^{[i]}(i|\bar{k})$ to obtain $H(i)$ and $S(i)$
5: **end**
6: **Return** $H(k), S(k)$

This procedure also allows to consider known moving obstacles, in the environment without particular difficulties.
For a distributed implementation, coupling constraints need to be considered differently since each controller has only partial information about the other robots position.

4.1.4. FHOCP and computational complexity

Having defined all the components of the optimization problem, the finite horizon optimal control problem can be formalized as:

$$\begin{aligned} &\underset{\Sigma \in \mathbb{W}}{min}\ J \\ &s.t \\ &p(l+1|k) = f_{\Sigma(k)}(p(l)) \\ &H_n^j p^{[i]}(l) \geq S_n^j \ \ \forall\ j = 1\ldots N_{obs}\ , i = 1\ldots N_{rob} \\ &H_n^j p^{[i]}(l) \geq S_n^j \ \ \forall\ , i = 1\ldots N_{rob},\ j \neq i \\ &\ \forall\ l = k\ldots k+N_p \end{aligned} \tag{4.9}$$

The optimization problem is a binary integer linear programming problem. This type of problem is known to be hard to solve since it belongs to the class of NP-hard problems. The difficulties in obtaining a solution can be highlighted by considering the cardinality of the set of possible sequences $\mathbb{W}$. The cardinality of the set $\mathbb{W}$ is equal to $2^{N_p N_{rob}}$ and it grows exponentially with the length of the prediction horizon N_p or the number of

robots N_{rob}. In order to reduce the computational complexity, several relaxations could be considered to decrease the amount of evaluations of the cost function. For instance, a possible relaxation can be obtained by considering that only one robot in the network can modify its dynamics at each time step or by considering move-blocking strategies [37]. Another possibility is to accept a suboptimal solution to the optimization problem.

4.2. Nominal switching model predictive control

In this section, a centralized and distributed formulation of switching MPC based on the finite horizon optimal control problem constructed in the previous section will be presented.

4.2.1. Centralized switching MPC

A centralized formulation can be directly obtained by solving at each time instant the FHOCP (4.9). Therefore, at each time instant, the algorithm to be executed in order to compute the switching signal to be applied is:

Algorithm 4.3 Centralized switching MPC

1: **Input:**$p(k)$
2: **Output:** First element of the optimal switching sequence Σ
3: **Foreach**$\Sigma_i \in \mathbb{W}$
4: Evaluate the associated cost J_i through simulation of the system using the switching sequence Σ_i
5: **IF** Σ_i does not violate constraint
6: Add i to set of feasible sequence $\mathbb{I}$
7: **end**
8: **end**
9: Compute index of optimal solution:

$$i^{opt} = \underset{i \in \mathbb{I}}{argmin}\ J_i$$

In case of multiple sequences with equal cost, consider the one that involves the minimum amount of movement for each robot
10: **Return** The first element of the sequence Σ_i

It is relevant to note that the proposed algorithm does not exploit stabilizing terminal constraints for which well-studied results exist [34]. There are other approaches which not

exploit terminal conditions to guarantee stability; see [38, 39]. However, for the proposed algorithm, it is possible to exploit a simpler approach to prove the following result:

Theorem 4.1. *Assume that the initial state $p(0)$ belongs to the set of states where optimization Problem (4.9) has a solution. Then the equilibrium point $\bar{p}$ defined by the N_{rob} reference signal is stable and the trajectories converge to a local minima.*

Proof. We start the proof by showing that if a feasible solution exists at the initial time instant $k = \bar{k}$, a feasible switching sequence will exist $\forall\ k$. Consider the optimal sequence computed at $k = \bar{k}$

$$\Sigma^{opt} = \begin{bmatrix} \Sigma^{opt}(\bar{k}|\bar{k}) & . & . & . & \Sigma^{opt}(\bar{k}+N_p-1|\bar{k}) \end{bmatrix} \tag{4.10}$$

Since the constraint involve only p_x, p_y and recalling that for $\sigma = 0$ we have that $p_x(k+1) = p_x(k)$ and $p_y(k+1) = p_y(k)$, a feasible sequence at $k = \bar{k}+1$ is:

$$\Sigma^{opt} = \begin{bmatrix} \Sigma^{opt}(\bar{k}|\bar{k}) & . & . & . & \Sigma^{opt}(\bar{k}+N_p-1|\bar{k}) & 0 \end{bmatrix} \tag{4.11}$$

This result ensures that at each time instant, at least one feasible sequence exists. Regarding stability, as standard in MPC theory, the cost function will be employed as a Lyapunov function. For this scope, an analysis of the property of an optimal solution has to be carried out. Let be given an optimal trajectory p_{opt}

$$p_{opt} = \begin{bmatrix} p^{opt}(\bar{k}|\bar{k}) & . & . & . & p^{opt}(\bar{k}+N_p-1|\bar{k}) \end{bmatrix} \tag{4.12}$$

with associated optimal cost $J^{opt} = \sum_{k=\bar{k}}^{N_p} l(p^{opt}(k), \bar{p})$, assume that the running cost $l(p^{opt}(k), \bar{p})$ has a minimum for $k = k^{min}$. Then, a feasible trajectory is:

$$p_{feas} = \begin{bmatrix} p^{feas}(\bar{k}|\bar{k}) & p^{feas}(\bar{k}+2|\bar{k}). & . & p^{feas}(k^{min}|\bar{k}) & . & . & p^{feas}(k^{min}|\bar{k}) \end{bmatrix} \tag{4.13}$$

With associated cost $J^{feas} = \sum_{k=\bar{k}}^{N_p} l(p^{feas}(k), \bar{p})$, being J^{opt} the optimal cost such that:

$$\begin{aligned} &J^{opt} \leq J^{feas} \\ &\sum_{k=\bar{k}}^{N_p} l(p^{opt}(k), \bar{p}) \leq \sum_{k=\bar{k}}^{N_p} l(p^{feas}(k), \bar{p}) \end{aligned} \tag{4.14}$$

Simplifying the expression:

$$\sum_{k=k^{min}}^{N_p} l(p^{opt}(k), \bar{p}) \leq \sum_{k=k^{min}}^{N_p} l(p^{feas}(k), \bar{p}) \tag{4.15}$$

and recalling that $l(p^{opt}(k^{min}), \bar{p}) \leq l(p^{opt}(k), \bar{p})$, the only possible solution for both inequality to hold is that $l(p^{min}(k), \bar{p}) = l(p^{opt}(k), \bar{p})\ \forall k = k^{min}, ., .N_p$. Therefore, for an optimal solution:

$$l(p^{opt}(\bar{k} + N_p), \bar{p}) \leq l(p^{opt}(\bar{k}), \bar{p}) \tag{4.16}$$

From this inequality, it is easy to quantify the decrease of the cost function ΔJ, since a feasible solution at $k = \bar{k} + 1$ is:

$$p_{opt} = \begin{bmatrix} p^{opt}(\bar{k}+1|\bar{k}) & . & . & . & p^{opt}(\bar{k}+N_p-1|\bar{k}) & p^{opt}(\bar{k}+N_p-1|\bar{k}) \end{bmatrix} \tag{4.17}$$

From which:

$$\Delta J = J(k+1) - J(k) = l(p^{opt}(\bar{k} + N_p), \bar{p}) - l(p^{opt}(\bar{k}), \bar{p}) \leq 0 \tag{4.18}$$

In conclusion, J is a weak Lyapunov function, which proves the stability of the proposed algorithm. In order to prove convergence, it can be simply noted that $Step9$ of Algorithm 4.3 ensures that the steady state condition is preferred to other possible limit cycles, this fact concludes the proof.

One comment is in order. The fact that only simple stability can be inferred is not surprising, since due to the limited set of motions at disposal, it is not always possible to find a trajectory that leads to a decrease of the cost. Therefore, we can conclude that the algorithm steers the robots to a (possibly only locally) optimal position.

A modification of the proposed algorithm which ensures practical stability is possible and can be applied to all algorithms that will be described in this chapter. However, such a modification involves the use of terminal constraint that due to the limited set of motions at disposal is not easy to satisfy and limits the set of feasible initial conditions. For completeness is shortly presented below.

Consider a modification to the cost function (4.5), in particular substitute $\bar{l}(p^{[i]}(k), \bar{p}^{[i]})$ with:

$$\bar{l}_0(p^{[i]}(k), \bar{p}^{[i]}) = \begin{cases} \bar{l}(p^{[i]}(k), \bar{p}^{[i]}) - R & if\ \bar{l}(p^{[i]}(k), \bar{p}^{[i]}) \geq R \\ 0 \qquad otherwise \end{cases} \tag{4.19}$$

With $R \geq R_c$, where R_c is defined as in (3.15). Consider also the additional terminal set

$\mathbb{X}_f$:

$$(p_x^{[i]} - \bar{p}_x^{[i]})^2 + (p_y^{[i]} - \bar{p}_y^{[i]})^2 < R \quad \forall i = 1, \ . \ , \ . \ , N_{rob} \tag{4.20}$$

Note that these terminal constraints could also be approximated via a similar procedure adopted for the obstacle avoidance problem.

Theorem 4.2. *Let $\mathbb{X}$ be the set of states of the optimization problem* (4.9) *with modified cost function* (4.19) *and terminal constraint* (4.20)*. The switching MPC control law with modified cost function* (4.19) *and terminal constraint* (4.20) *ensure that the closed-loop system is practically asymptotically stable with a domain of attraction $\mathbb{X}$.*

Proof. Recursive feasibility can be inferred as in the proof of Theorem 4.1. Regarding stability, we consider as before the cost function as a Lyapunov function. In particular, consider the optimal trajectory p^{opt} computed at $k = \bar{k}$. Due to the additional terminal constraint, the cost is such that:

$$l(p^{opt}(\bar{k} + N_p), \bar{p}) = 0 \tag{4.21}$$

Then, at $k = \bar{k} + 1$ a feasible solution is obtained extending the previously computed optimal switching sequence as in (4.11), therefore it exists a solution that ensures the decrease of the cost function ΔJ:

$$\Delta J = J(k+1) - J(k) = l(p^{opt}(\bar{k} + N_p), \bar{p}) - l(p^{opt}(\bar{k}), \bar{p}) = -l(p^{opt}(\bar{k}), \bar{p}) \tag{4.22}$$

Therefore $\Delta J < 0 \ \forall \ p \notin \mathbb{X}_f$, ensures the convergence of the trajectory to the set $\mathbb{X}_f$, so completing the proof.

Simulation results

In order to study the Switching-MPC controller performance, Algorithm 4.2 has been proved in a randomly generated scenario for a network of mobile robots $N_{rob} = 3$ and parameter $N_p = 2, T = 0.1$. Since the cardinality of the set of possible sequences to be explored is quite small, 2^6, it is possible to apply the algorithm without particular relaxation. In Figure 4.2 the evolution of the robot network at different time instants is represented. where red circles represent robot position, blue circles represent obstacles and squares represent desired reference positions.

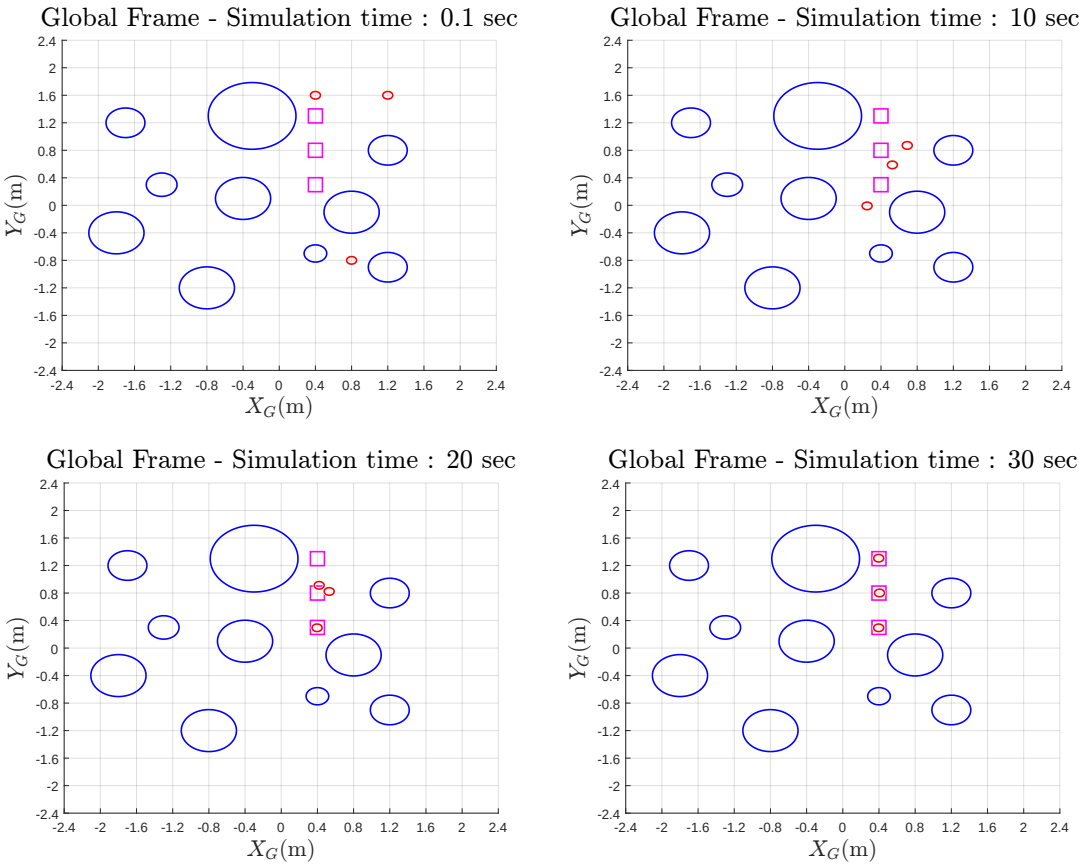


Figure 4.2: State of the robot network at different time instants

In figure 4.2 red circles represent robot position, blue circles represent obstacles and squares represent desired reference positions. The robot network reaches the desired reference position while avoiding collision. It is also interesting to look at the behavior of the cost function. In Figure 4.3 is possible to see that the cost function is always non-increasing. For completeness, the switching sequence is also presented.

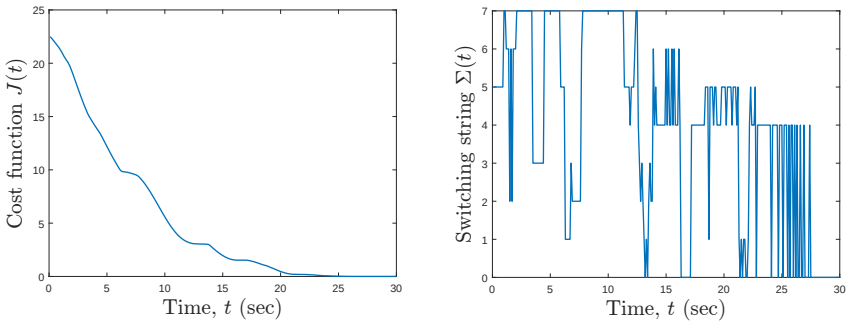


Figure 4.3: Cost function and switching string of centralized switching MPC

It is also possible to consider a larger network of robots and a longer prediction horizon. However, considering that the set of possible sequences grows exponentially and some relaxation is required. For the next example, a network of $N_{rob} = 5$ mobile robot and parameter $N_p = 5, T = 0.1$ is considered. The cardinality of the set of possible sequences is 2^{25}, which is clearly not suitable. A possible relaxation is to early terminate the optimization when a suboptimal solution, which ensures a decrease in the cost function is found. Similarly to the previous simulation, the evolution of the network will be represented at different time instants.

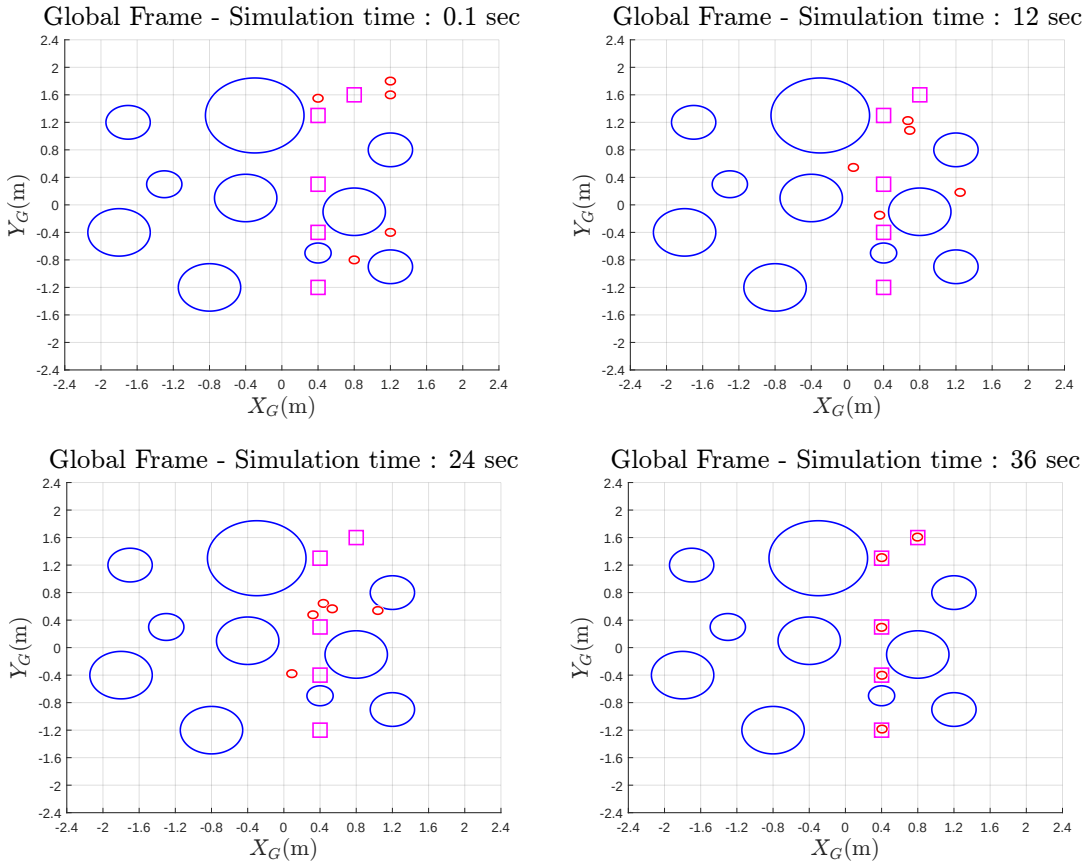


Figure 4.4: State of the robot network at different time instants

While it is possible to see that the cost function is indeed non-increasing as depicted in Figure 4.5, the obtained solution is clearly suboptimal, and the performance degrades with the number of involved robots.

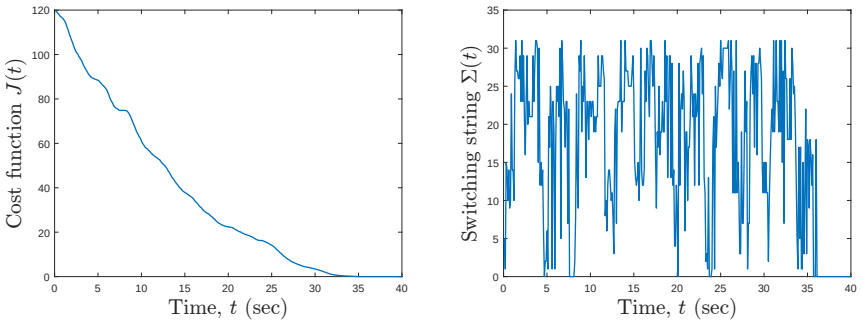


Figure 4.5: Cost function and switching string of centralized switching MPC

4.2.2. Distributed switching MPC

The scope of this section is to develop a distributed non-cooperative formulation, see [40] for a survey on distributed MPC. Since the cost function (4.5) is separable, the obstacle avoidance constraint involves only a robot at a time and the dynamics of each robot are decoupled from each other so that the distributed implementation regards the control of the independent systems subject to coupling constraint (collision avoidance). In order to obtain a suitable reformulation of the collision avoidance constraint the approach in [11] will be used. In particular, each robot will exploit the previously predicted trajectories to obtain a suitable constraint to enforce. To this scope, the algorithm needed to construct this aim is modified as follows.

Algorithm 4.4 Construction of collision avoidance constraint

1: **Input:** Radius $Rrob$ of the robots, previously predicted trajectory $p^{[i]}(k|\bar{k}), p^{[j]}(k|\bar{k})$ $k = \bar{k}, ., .\bar{k} + N_P$, number of edge n of the polytopic approximation
2: **Output:** matrix $H^{ij}(k)$ and $S^{ij}(k)$ which define the collision avoidance constraint of robot i with respect to robot j
3: **U**se Algorithm 4.1 to construct a polytopic approximation $\bar{H}^j_n(k)$,$\bar{S}^j_n(k)$ $\forall$ $k = \bar{k} + 1, ., .\bar{k} + N_P + 1$for the collision constraint, using $p^{[j]}_{pre}(k, \bar{k})$ $\forall$ $k = \bar{k} + 1, ., .\bar{k} + N_P$ and $p^{[j]}_{pre}(\bar{k} + N_p, \bar{k})$ for $k = \bar{k} + N_p + 1$
4: **Foreach** $k = \bar{k} + 1, ., .\bar{k} + N_P + 1$
5: Compute $\bar{n}(k)$ as: $\bar{n} = \underset{n}{argmax}\ \ \bar{H}^j_n(k)p^{[i]}(k|\bar{k}) - \bar{S}^j_n(k)$
6: compute $\rho^{ij}(k) = \bar{H}^j_{\bar{n}}(k)p^{[i]}_{pre}(k, \bar{k}) - \bar{S}^j_{\bar{n}}(k))$
7: **end**
8: define $H^{ij}(k) = H^j_{\bar{n}}(k)$,$S^{ij}(k) = \bar{S}^j_{\bar{n}}(k) + \frac{\rho^{ij}(k)}{2}$
9: **Return** $H^{ij}(k)$,$S^{ij}(k)$

In practice, the presented algorithm uses the predicted trajectory to divide the state space for all time instants in 2 suitable half-planes constraining the trajectory of 2 different robots. Enforcing this time-varying constraint trivially allows to satisfy the original collision avoidance constraint. It is relevant to note that the constraints should be included for all pairs of robots. Therefore, an all-to-all communication network supporting the exchange of information between the robots would be required. However, it is evident that these constraints are only meaningful if the robots are sufficiently close to each other. To this scope, define $\mathcal{C}^i(k)$ the set of neighbor robot of the robot i as:

$$\mathcal{C}^i(k) = \{j \ | \bar{l}(p^{[i]}(k|k), p^{[j]}(k|k)) < R_n\} \tag{4.23}$$

Where R_n defines the maximal distance for 2 neighbor robots and it has to be sufficiently large in order to consider as a neighbor each robot pair that can collide. Furthermore, differently from the centralized formulation, where it is easy to prove convergence due to the property of the optimal trajectory, the complex time-varying nature of the constraint defined in Algorithm 4.3 requires an additional terminal constraint which simplifies the analysis:

$$l(p^{[i]}(k+Np|k), \bar{p}^i) \leq l(p^{[i]}(k+Np-1|k-1), \bar{p}^i) \tag{4.24}$$

Having reformulated the coupling constraints, the algorithm each robot has to execute in order to compute the switching signal is the following one:

Algorithm 4.5 Distributed non-cooperative switching MPC

1: **Input:**$p^i(k)$,predicted trajectory $p^{[j]}(k|\bar{k})$ $k = \bar{k}, ., .\bar{k} + N_P \ \forall \ j \ \in \mathcal{C}^i(k)$
2: **Output:** First element of the optimal switching sequence σ and predicted trajectory $p^{[i]}(k|\bar{k}+1)$ $k = \bar{k}+1, ., .\bar{k} + N_P + 1$
3: execute Algorithm 3.3 $\forall \ j \ \in \mathcal{C}^i(k)$ in order to obtain distributed constraint
4: **For each** $\sigma_i \in \mathbb{W}$
5: Evaluate the associated cost $\bar{J}_i$ through simulation of the system using the switching sequence Σ_i
6: **IF** σ_i does not violate constraint:Add i to set of feasible sequence $\mathbb{I}$
7: **end**
8: Compute index of optimal solution: $i^{opt} = \underset{i \in \mathbb{I}}{argmin} \ \bar{J}_i$. In case of multiple sequences with equal cost, consider the one that involves the minimum amount of movement for each robot
9: **Return** The first element of the sequence σ_i and predicted trajectory $p^i(k|\bar{k}+1)$ $k = \bar{k}+1, ., .\bar{k} + N_P + 1$

The algorithm ends by applying the obtained switching signal and by communicating to the neighbors the predicted trajectory in preparation for the next execution of the algorithm. A simple initialization procedure for the predicted trajectory is:

$$p_{pre}^{[i]}(k,0) = p^{[i]}(0) \quad \forall k = 0,.,.,N_p - 1 \tag{4.25}$$

which consists in initializing the predicted trajectory with the stationary initial position. The proposed distributed algorithm has a non-cooperative nature since each controller optimizes its own local cost function. Each distributed controller is evaluated in parallel, and each subsystem is allowed to change the predicted trajectories at each time step. Other design procedures that allow cooperation are possible but typically involve sequential optimization, [41], negotiation or consider limitations regarding the frequency of update of the plan for each subsystem [42, 43]. Furthermore, artificial deadlocks due to polytopic approximation are more likely with respect to the centralized case, and a possible strategy to overcome this situation can be derived along the lines of [44].

For the proposed algorithm it is possible to prove the following result.

Proposition 4.1. *Assume that the initial state $p^{[i]}(0)$ belongs to the set of states where Algorithm 4.5 has a solution. Then, the equilibrium point $\bar{p}^{[i]}$ defined by the N_{rob} reference signal is stable and the trajectory of each subsystem i converges to a local minima with respect to $\bar{J}^{[i]}$*

Proof Recursive feasibility is ensured with a similar argument to the proof of Theorem 4.1 by considering an extended trajectory:

$$\sigma^{opt} = \begin{bmatrix} \sigma^{opt}(\bar{k}|\bar{k}) & . & . & . & \sigma^{opt}(\bar{k}+N_p-1|\bar{k}) & 0 \end{bmatrix} \tag{4.26}$$

while constrain (4.20) ensures that the algorithm converges since the positive definite function $L = l(p^{[i]}(k+Np|k)$ is non-increasing.
It has to be noted that the additional constraint (4.20) is not strictly needed, and the algorithm offers good performance even without it. However, it is possible to prove that the complex nature of the time-varying constraint can lead to sporadic increase of the cost. This situation arises more often when a longer sampling time is employed. As a matter of fact, for a short sampling time the constraint becomes constant along the prediction horizon and the same reasoning of Theorem 4.1 can be used to prove that the cost function is non-increasing.

Computational complexity

The proposed distributed algorithm presents strong computational advantages with respect to the centralized algorithm. For the centralized algorithm the cardinality of the set of the possible sequence $\mathbb{W}$ is equal to $2^{N_p N_{rob}}$ while the distributed algorithm requires to solve N_{rob} independent optimization problem each characterized by a set of cardinality 2^{N_p}. The cardinality of the set still grows exponentially with the length of the prediction horizon. Therefore a relaxation could still be useful to reduce the computational burden.

Simulation results

In order to illustrate the distributed switching-MPC controller performance, the same scenario as in Section 4.2.1 is considered, for a network composed by $N_{rob} = 5$ mobile robots and $N_p = 5, T = 0.1$. The evolution of the network is reported in (4.6).

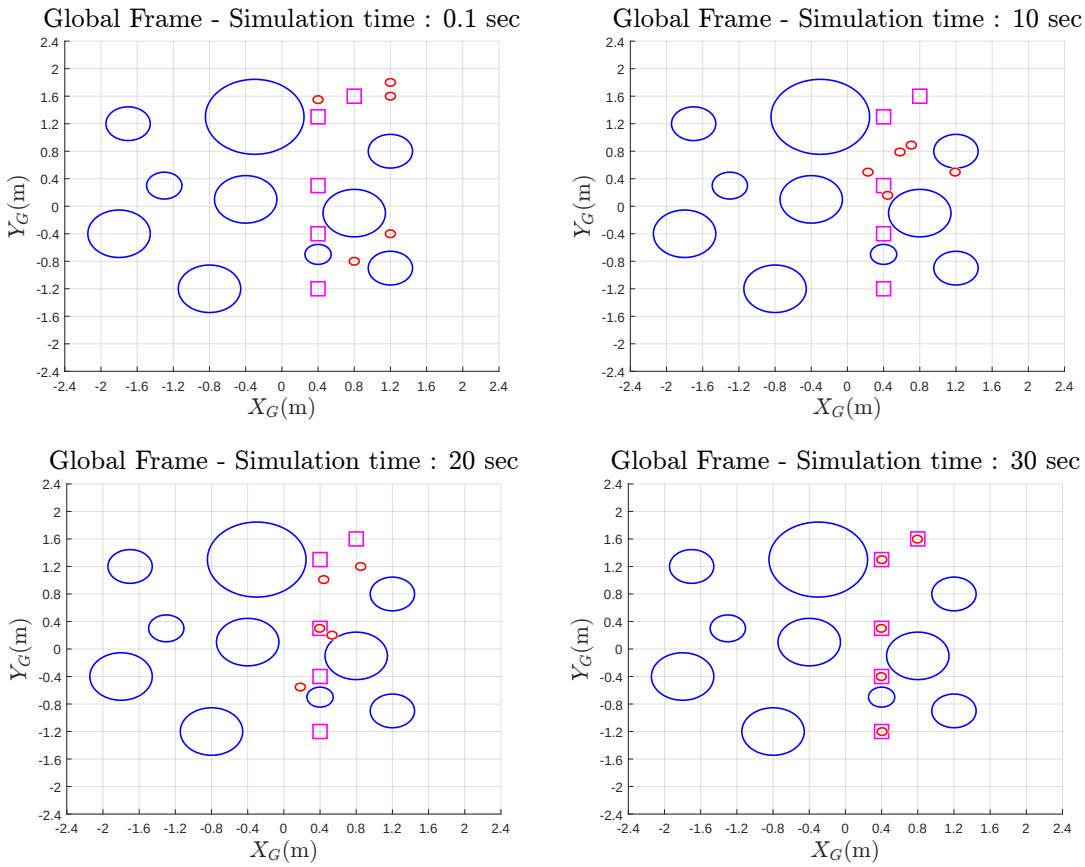


Figure 4.6: State of the robot network at different time instants

The distributed algorithm performs better than the centralized relaxed algorithm and is more scalable with respect to the number of robots.

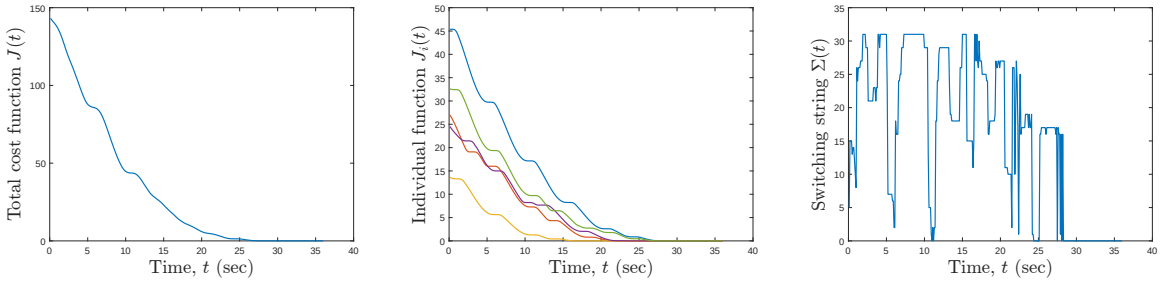


Figure 4.7: Cost function and switching string of distributed switching MPC

In order to show the scalability of the proposed algorithm a similar scenario is considered for a network composed of $N_{rob} = 15$ mobile robot and $N_p = 5, T = 0.1$.

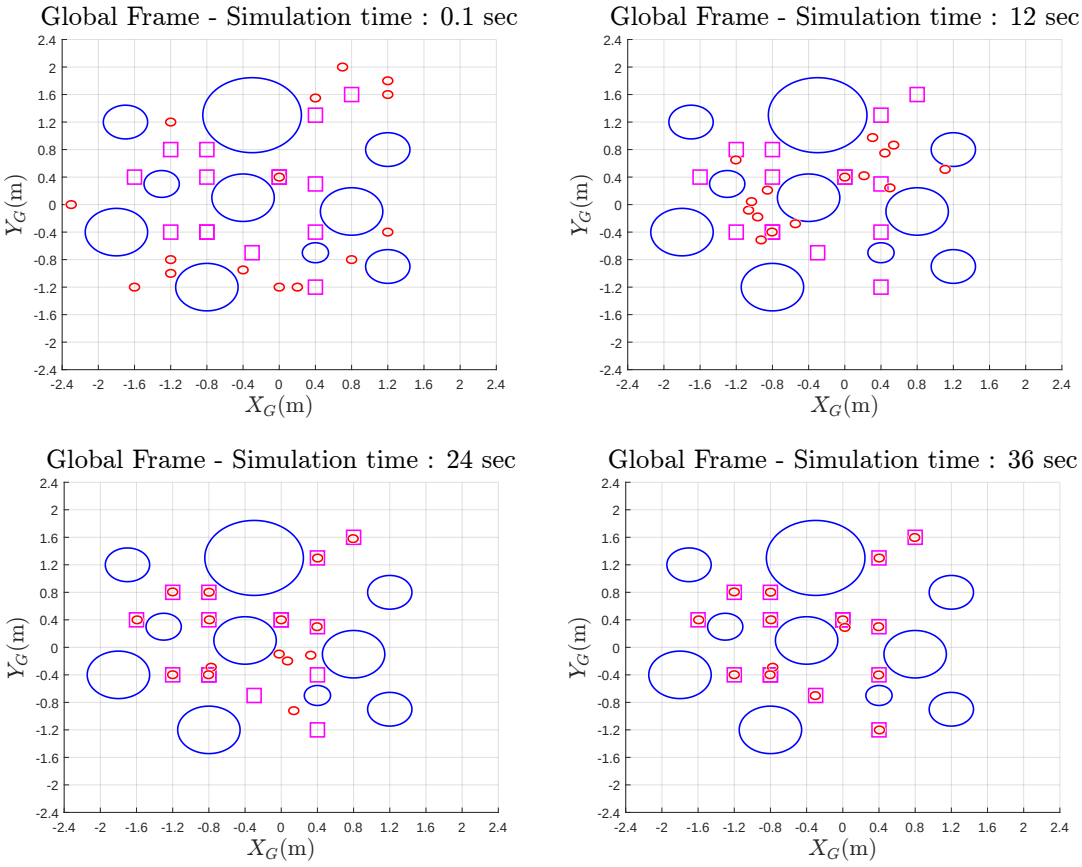


Figure 4.8: State of the robot network at different time instants

It is relevant to note the non-cooperativity of the proposed algorithm, is quite evident in the presence of multiple robots sharing the same reference position.

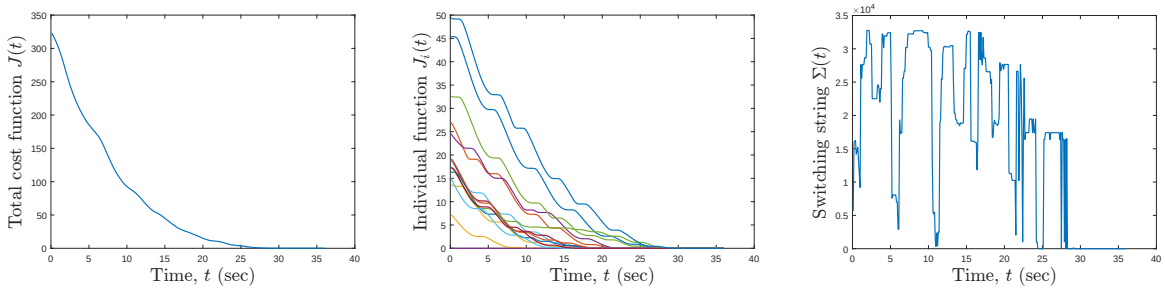


Figure 4.9: Cost function and switching string of distributed switching MPC

4.3. Reference layer

The proposed algorithms make possible to steer each mobile robot to a desired reference position while satisfying both obstacle avoidance and inter-robot collision constraints. Exploiting this fact, it is possible to tackle several other problems by properly managing the reference signal, such as path following and self-aggregation among many others.

4.3.1. Path following

Path following can be achieved by sampling the desired trajectory choosing as reference signal the nearest sampled point, and selecting the successive one once convergence is reached. This approach is valid for both centralized and distributed implementation.
It has to be noted that following this approach the cost function is allowed to increase during a target change. Moreover, during a target change convergence constraints of the distributed implementation are not to be considered in order to allow maximal freedom.

Simulation results

In order to study the performance of distributed switching-MPC for a path-following task, the proposed strategy is simulated for a network of $N_{rob} = 10$ mobile robots and parameter $N_p = 2, T = 0.1$.
In Figure 4.10 red circles represent the position of the robots, blue circles represent obstacles in the environment and the blue line represents the path to be followed by the robot network.

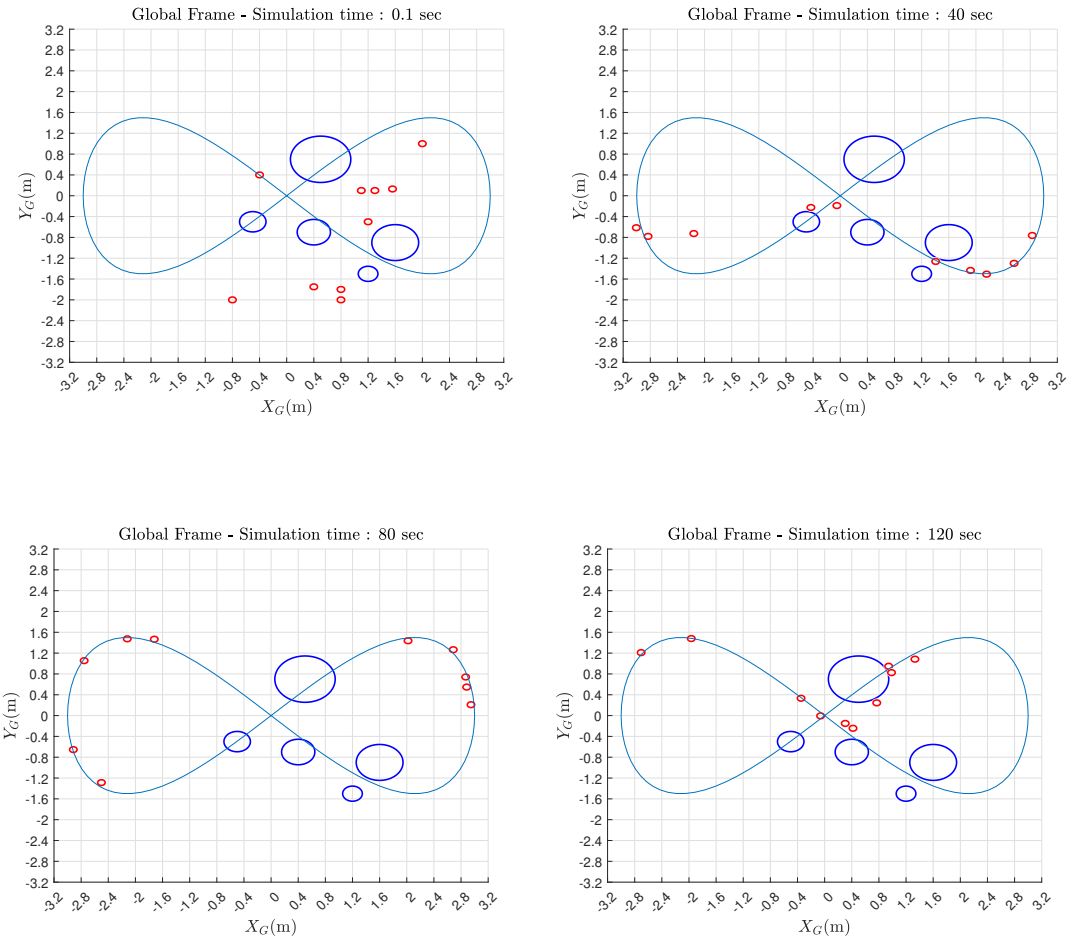


Figure 4.10: State of the robot network at different time instants

The approach makes all mobile robots follow as much as possible the given path. Furthermore, in Figure 4.11 it is possible to see that the cost function temporarily increases, this happens when the target position is changed in order to advance the reference position or due to obstacles that make impossible further improvement are present.

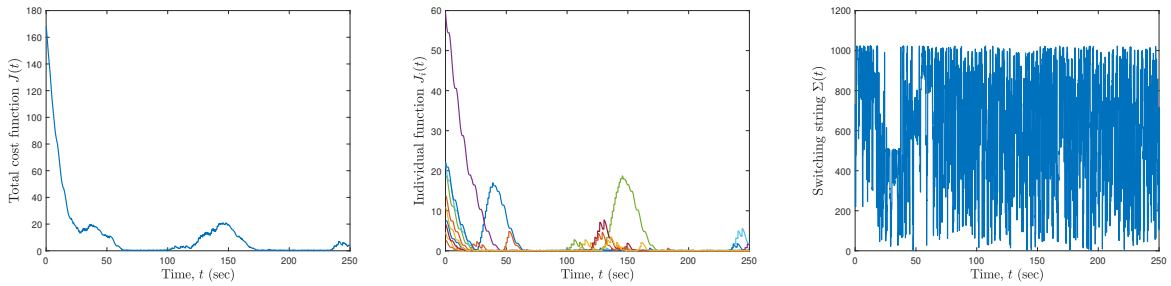


Figure 4.11: Cost function and switching string of distributed switching MPC for path following

4.3.2. Self-aggregation

Regarding problems that involve coordination, such as self-aggregation, the situation is different. In the case of a centralized implementation, no particular adjustments have to be made, since several problems can be tackled by modifying the reference signal. For instance, in the case of self-aggregation the reference signal could be chosen as the centroid of the group of robots. The case of a distributed implementation problem which involves coordination are more challenging and could require more complex reference selection and even additional constraints. For instance, a possible approach is to obtain the reference signal exploiting a discrete time averaging algorithm, see [45], to achieve consensus regarding the reference signals. Another possibility is to let each controller consider in the optimization also a hypothetical plan for a neighboring agent, see [43]. In addition, also the connectivity of the network has to be ensured which can be done by adding suitable connectivity constraints.
To summarize, a possible approach to achieve self-aggregation is

- compute for each agent the reference position using an averaging algorithm, i.e.,

$$\bar{p}^{[i]}(k+1) = average(\bar{p}^{[i]}(k), \{\bar{p}^{[j]}(k), \forall\ neighbour\ robot\ j\}) \tag{4.27}$$

  It is possible to prove that if the graph $\mathcal{G}$ composed by a robot (node) and a communication link (edge) is connected, the algorithm achieves asymptotic consensus, see [45]. It is relevant to note that the algorithm does not achieve average consensus on the initial position due to the fact that some robots can have greater influence depending on the number of neighbors.

- The averaging algorithm achieves consensus only if the communication graph $\mathcal{G}$ is connected, to this scope an additional constraint to maintain pairwise connectivity with a maximum communication distance of R_{com} can be imposed:

$$\begin{aligned} p^{[i]}(k) &\in \mathcal{B}(\frac{p^{[i]}(k) + p^{[j]}(k)}{2}, \frac{R_{com}}{2}) \\ p^{[j]}(k) &\in \mathcal{B}(\frac{p^{[i]}(k) + p^{[j]}(k)}{2}, \frac{R_{com}}{2}) \end{aligned} \tag{4.28}$$

  where $\mathcal{B}(c, r)$ is a ball centered in c with radius r_c. The constraints in practice limit the maximum distance between 2 robots to R_{com}. This nonlinear constraint can be managed in a similar way as for obstacle avoidance previously described, using inner polytopic approximation, obtaining a set of n inequalities. A main difference between the two cases is that for connectivity maintenance robots are required to lie

inside the polytope, and therefore all the following inequalities have to be satisfied:

$$H_i p^{[i]}(k) \leq S_i \quad \forall i = 1, ., ., n \tag{4.29}$$

The described constraints are quite conservative in enforcing network connectivity since they require maintaining each connection present. In order to relax the constraint it is possible to maintain the connectivity only with respect to a sub-graph of $\mathcal{G}$ which shares the same connected component, see [46] for more detail.

Simulation results

To study the performance of the proposed approach for a self-aggregation task, the algorithm has been tested on a randomly generated network of $N_{rob} = 20$ mobile robots and parameter $N_p = 2, T = 0.1, R_{com} = 1$. In Figure 4.12 red circle represents robot position while black circle represents the reference position for each robot.

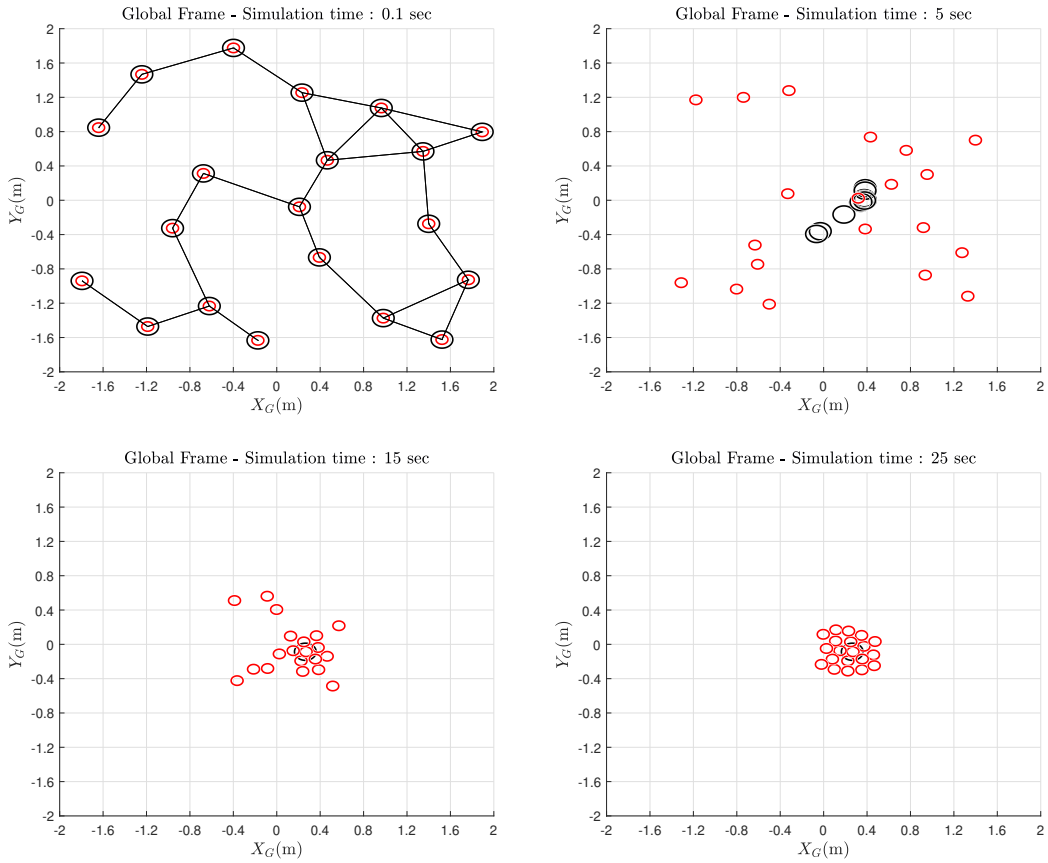


Figure 4.12: State of the robot network at different time instants

The network correctly aggregates in one cluster, due to the fact that as shown in (4.12) the communication graph is connected.

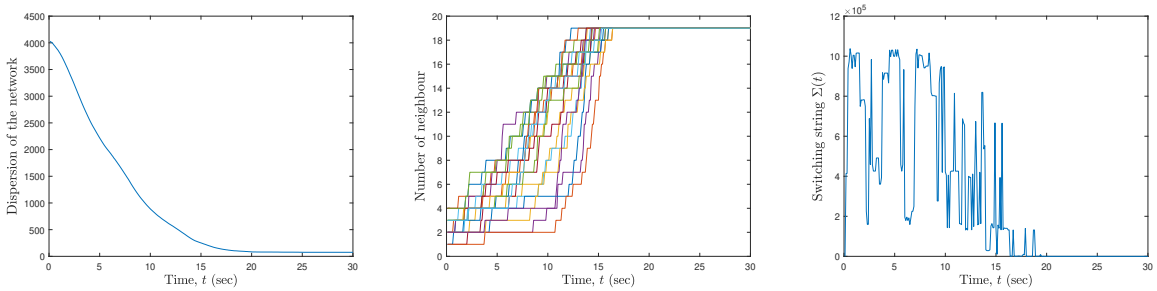


Figure 4.13: Cost function, switching string and number of neighbors

If the communication graph never becomes connected the network aggregates in multiple clusters.

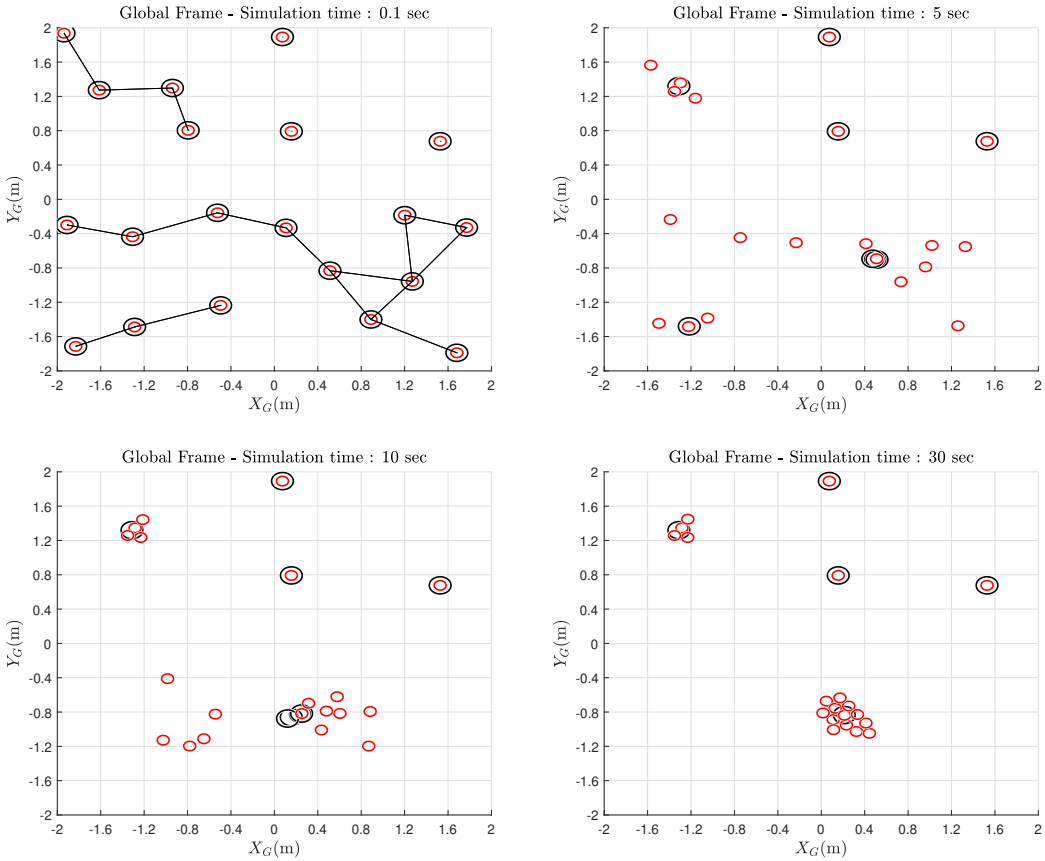


Figure 4.14: State of the robot network at different time instants

The majority of the network aggregates 2 larger clusters, while some mobile robots that never enter the communication range remain stationary.

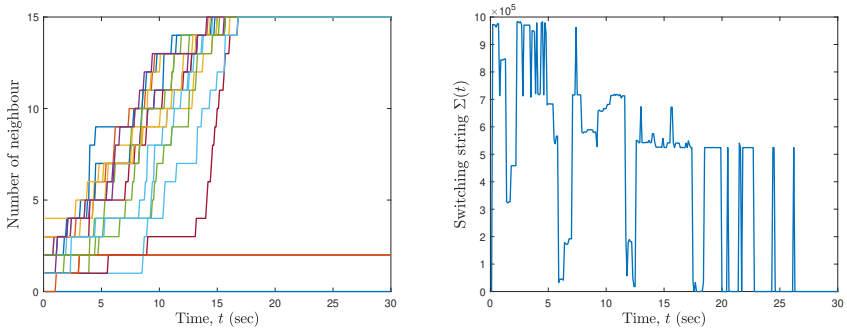


Figure 4.15: Number of neighbour and switching string

4.4. Robust switching MPC

In this section, a robust formulation for the switched MPC controller is obtained based on the multiplicative disturbances model (2.14). Algorithm 4.3 and 4.4 will be used as a baseline and a proper modification needed to obtain a robust formulation will be presented. In this section we will focus on a robust tube-based approach as formalized in [26, 47] with a fundamental modification. Due to the discrete nature of the switching signal, it is difficult to parametrize the switching law along the prediction horizon. Therefore, the algorithm is constrained to adopt an "open loop" strategy, [48]. The choice of an open-loop strategy affects the prediction of disturbances along the prediction horizon limiting the algorithm since no auxiliary control law is exploited in the prediction. In a later chapter, a different strategy able to cope with additive disturbances will be presented. Before characterizing the required constraint tightening and the modification to be applied to the cost function for a robust implementation, we recall the model equations:

$$p^{[i]}(k+1) = f_{\sigma(k)}(p^{[i]}(k)) \tag{4.30}$$

where:

$$f_0 = \begin{bmatrix} p_x^{[i]}(k) \\ p_y^{[i]}(k) \\ \theta^{[i]}(k) + T\omega_0 \end{bmatrix} \quad f_1 = \begin{bmatrix} p_x^{[i]}(k) + v_1 \frac{\sin(theta^{[i]}(k)+T\omega_1)-\sin(\theta^{[i]}(k))}{\omega_1} + d_x(k) \\ p_y^{[i]}(k) + v_1 \frac{-\cos(theta^{[i]}(k)+T\omega_1)-\cos(\theta^{[i]}(k))}{\omega_1} + d_y(k) \\ \theta^{[i]}(k) + T\omega_1 \end{bmatrix} \tag{4.31}$$

where $d_x^2 + d_y^2 < Tv_\sigma \bar{d} = D_M$ as discussed in Section 2.3.

4.4.1. Constraint tightening and worst-case cost function

A common approach used to obtain a robust formulation for MPC is to consider the worst-case cost during optimization. Obtaining analytically such a cost is quite difficult. It is possible to consider an upper bound, trying to characterize the possible trajectory of the system along the prediction horizon once that $\sigma^{[i]}(k)$ is defined.
In particular the prediction error of the cartesian position $e(k) = \begin{bmatrix} e_x(k) & e_y(k) \end{bmatrix}$ is contained in the set:

$$\begin{aligned} E(k) &= E(k-1) \oplus \sigma^{[i]}(k)\mathcal{B}(0, D_M) \\ E(0) &= \{0\} \end{aligned} \tag{4.32}$$

where $\oplus$ is the Minkowski sum[1], $B(0, D_M)$ is the set that defines a ball centered at the origin with radius D_M.
It is easy to show that the set E is equal to:

$$\begin{aligned} E(k) &= B(0, R_t(k)) \\ R_t(k+1) &= R_t(k) + D_M \\ R_t(0) &= 0 \end{aligned} \tag{4.33}$$

Therefore, it is possible to define the worst-case cost function as:

$$\begin{aligned} J_w &= \sum_{i=1}^{N} \bar{J}_w^{[i]} \\ \bar{J}_w^{[i]} &= \sum_{k=0}^{N_p} \bar{l}_w(p^{[i]}(k), \bar{p}^{[i]}, R_c^{[i]}(k)) \end{aligned} \tag{4.34}$$

Where l_w is defined as the maximum distance from the desired set point for a point located on a ball centered on $p^{[i]}(k)$ with radius $R_t^{[i]}(k)$.
Regarding robust constraint satisfaction, we can consider a progressive tightening of the original constraint defined in Section 4.1.3.
In particular, given a polytopic approximation of the distance constraint $H_{\bar{n}}p^{[i]}(k) \geq S_{\bar{n}}$ it is possible to ensure the fulfillment of the robustness property at a generic time instant $\bar{k}$ of the prediction horizon by enforcing the constraint:

$$H_{\bar{n}}p^{[i]}(k) \geq S_{\bar{n}} + R_t^{[i]}(k) \tag{4.35}$$

This constraint ensures that there is a margin of $R_t^{[i]}(k)$ from the violation of the con-

[1]The Minkowski sum of two sets A and B is formed by adding each vector in A to each vector in B.

straint, and therefore the original constraint is automatically satisfied for each possible disturbance realization. In case of a constraint that involves two robots, it is necessary to consider the uncertainty related to all the robots involved, and therefore the considered tightening will be $R_t(k) = R_t^{[i]}(k) + R_t^{[j]}(k)$. Considering the described modification, the robust optimization problem to be solved at each time instant is:

$$\begin{aligned}
&\underset{\Sigma \in \mathbb{W}}{min}\ J_w \\
&s.t \\
&p(l+1|k) = f_{\Sigma(k)}(p(l)) \\
&R_c^{[i]}(l+1) = R_c^{[i]}(l) + \sigma^{[i]}(l) D_M \quad \forall i = 1 \ldots N_{rob} \\
&H_n^j p^{[i]}(l) \geq S_n^j + R_c^{[i]}(l) \quad \forall\ j = 1 \ldots N_{obs}\ , i = 1 \ldots N_{rob} \\
&H_n^j p^{[i]}(l) \geq S_n^j + R_c^{[j]}(l) + R_c^{[i]}(l) \quad \forall\ j = 1 \ldots N_{obs}\ , i = 1 \ldots N_{rob},\ j \neq i \\
&\quad \forall\ l = k \ldots k + N_p
\end{aligned} \tag{4.36}$$

With this robust formulation of the FHOCP (4.9) it is possible to reformulate the MPC strategy described for the nominal case.

4.4.2. Robust centralized switching MPC

A centralized formulation can be directly obtained by solving at each time instant the FHOCP (4.36), therefore at each time instant the algorithm to be executed in order to compute the switching signal to apply is the following:

Algorithm 4.6 Robust centralized switching MPC

1: **Input:**$p(k)$
2: **Output:** First element of the optimal switching sequence Σ
3: **Foreach**$\Sigma_i \in \mathbb{W}$
4: Evaluate the associated cost J_{wi} and the constraint tightening $R_c^{[i]}$ through simulation of the system using the switching sequence Σ_i
5: **IF** Σ_i does not violate tightened constraint:Add i to set of feasible sequence $\mathbb{I}$
6: **end**
7: Compute index of optimal solution: $i^{opt} = \underset{i \in \mathbb{I}}{argmin}\ J_{wi}$. In case of multiple sequences with equal cost, consider the one which involves the minimum amount of movement for each robot
8: **Return** The first element of the sequence Σ_i

Similar comments to Algorithm 4.3 can be made regarding the absence of terminal constraint and the possibility of using a modified cost function as in (4.2). However, following

the same step of Theorem 4.1 it is possible to prove the following result.

Theorem 4.3. *Assume that the initial state $p(0)$ belongs to the set of states where optimization Problem 4.36 has a solution. Then the equilibrium point $\bar{p}$ defined by the N_{rob} reference signal is stable, and the trajectories converge to a local minima.*

Proof. The proof follows the same step of the proof of Theorem 4.1 once noticed that the value $\bar{l}_w(p^{[i]}(k), \bar{p}^{[i]}, R_c^{[i]}(k))$ does not depend on the particular realization of the disturbances.

Simulation results

In order to study the robust switching-MPC controller performance, it is first shown a simulation for an obstacle-free scenario for a network of $N_{rob} = 3$ mobile robots, parameters $N_p = 2, T = 0.1$ and disturbances acting on each robot characterized by $D_M = \frac{a_1 T}{2}$, corresponding to a maximum uncertainty equal to half of the movement of the robot.

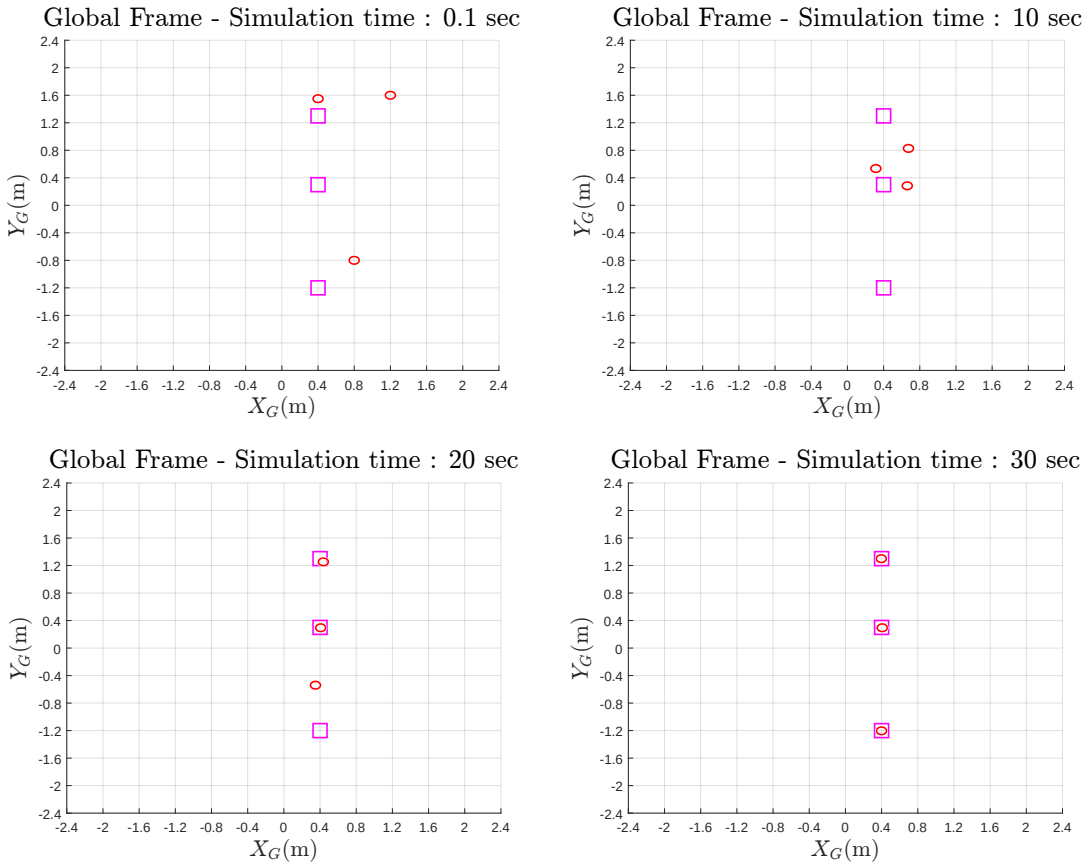


Figure 4.16: State of the robot network at different time instants

The network converges to the desired reference position. However, differently from a

scenario without disturbances acting on the system, the controller accepts a sub-optimal steady state position due to the impossibility of ensuring a decrease in cost with further movement.

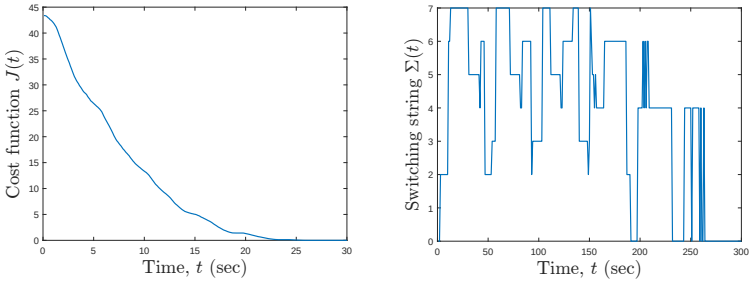


Figure 4.17: Cost function and switching string of robust centralized switching MPC

From Figure 4.17 it is possible to see that the algorithm ensures the non-increase of the cost function. The conservativism of the proposed algorithm is more evident in the case of an environment dense with obstacles.

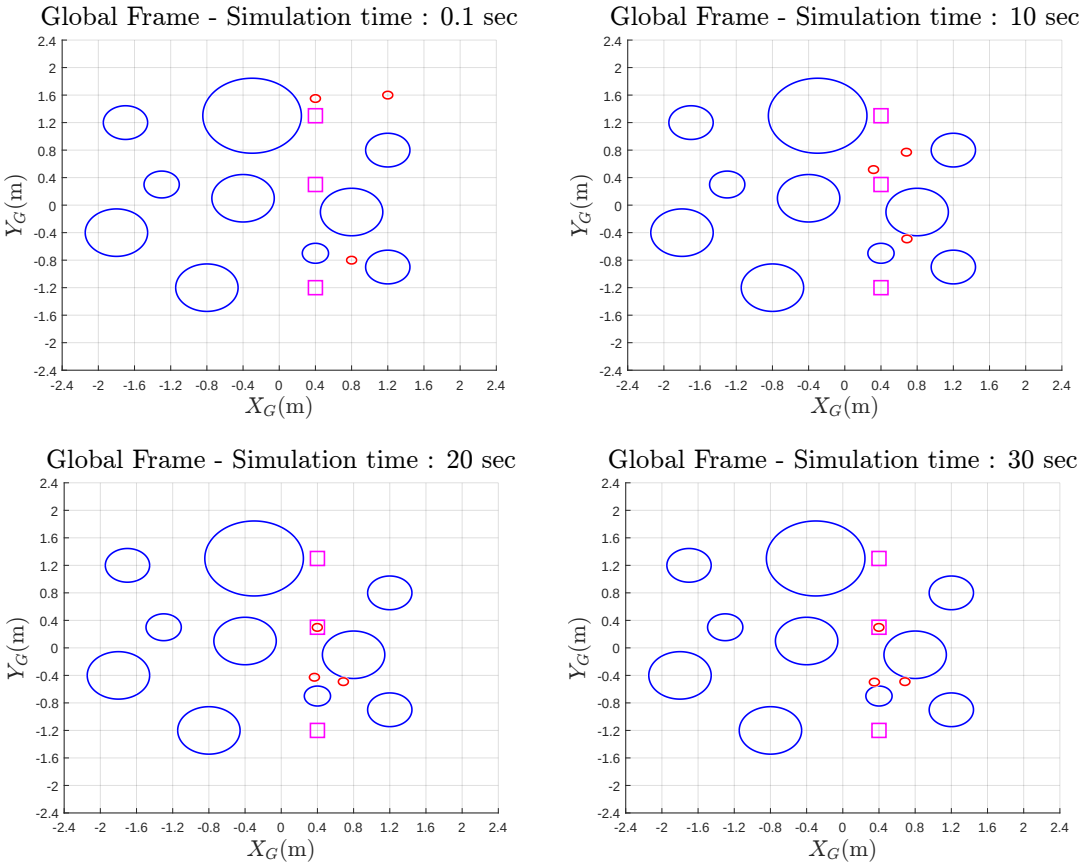


Figure 4.18: State of the robot network at different time instants

In Figure 4.19 it is possible to see that the non-increase of the cost function is ensured. The conservativism of the algorithm is evident for obstacle avoidance, whereas it can be seen in Figure 4.18, that the robots are unable to reach the desired reference position due to the tightened constraints. This type of situation is clearly more common for short prediction horizon.

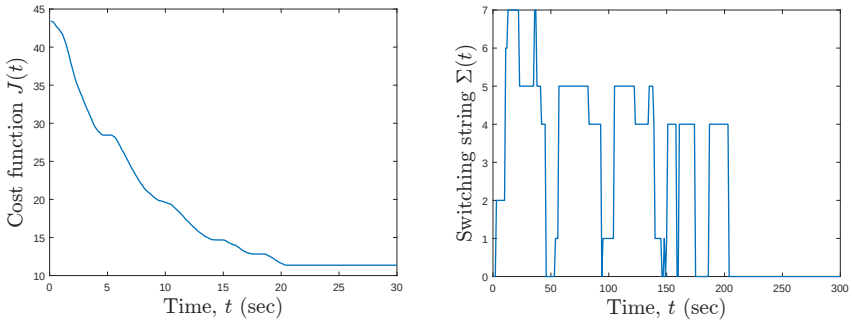


Figure 4.19: Cost function and switching string of robust centralized switching MPC

4.4.3. Robust distributed switching MPC

Obtaining a distributed formulation for the proposed robust algorithm is quite straightforward, and it can be done with mild modifications to the procedure described in section 4.2.2

- Consider the worst case cost function J_w, described in (4.34), in place of the original cost function J.
- Consider for obstacle avoidance the constraint tightening $R_c^{[i]}(k)$ as described in (4.33).
- Regarding inter-robot collision avoidance, the robots need to communicate besides the predicted trajectory, also the related constraint tightening and use in Algorithm 4.4 the tightened constraint with uncertainty $R_c^{[i]}(k|k-1) + R_c^{[j]}(k|k-1)$

Considering these modifications, the same results as in Theorem 4.1 can be obtained, without particular differences.

Simulation results

In order to study the robust switching-MPC controller performance, it is first shown a simulation for an obstacle-free scenario for a network of $N_{rob} = 15$ mobile robots, parameters $N_p = 5, T = 0.1$ and disturbances acting on each robot characterized by $D_M = \frac{a_1 T}{2}$ which correspond to a maximum uncertainty equal to half of the robot movement.

In Figure 4.20 red circles represent the position of the robots, while squares represent the desired reference position.

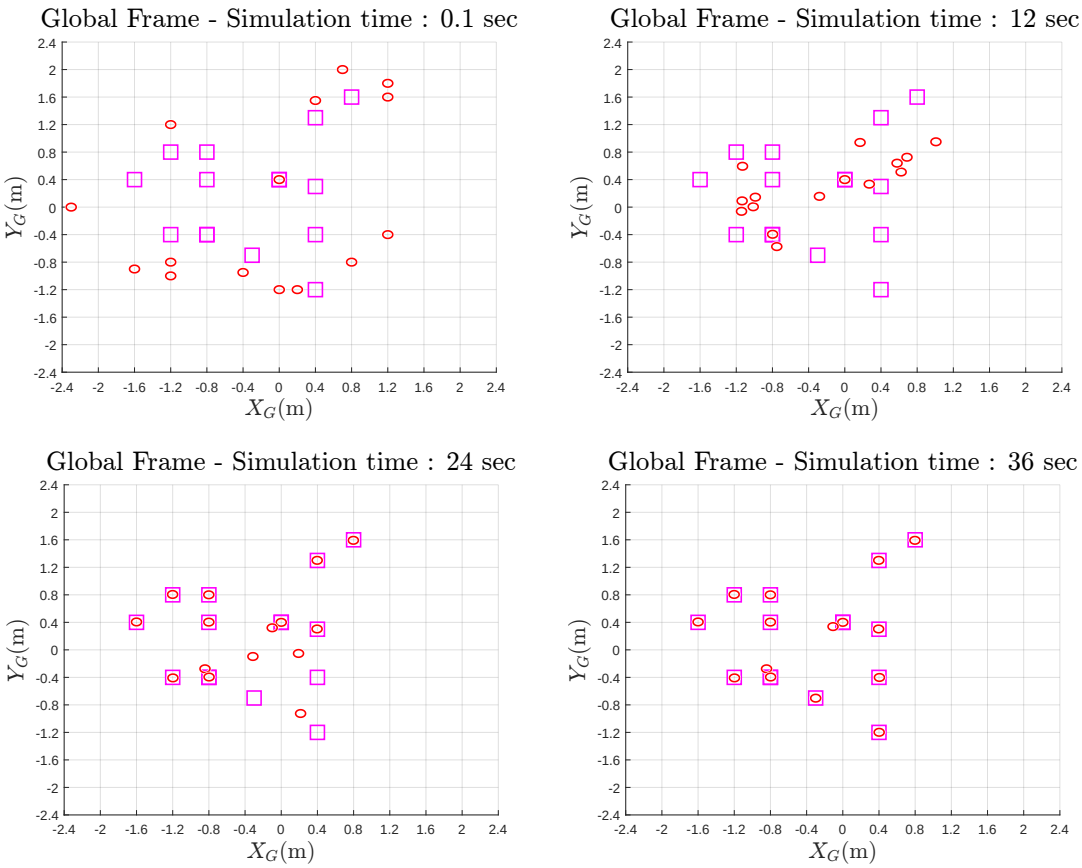


Figure 4.20: State of the robot network at different time instants

The network converges to the desired reference position, however differently from a scenario without disturbances acting on the system.

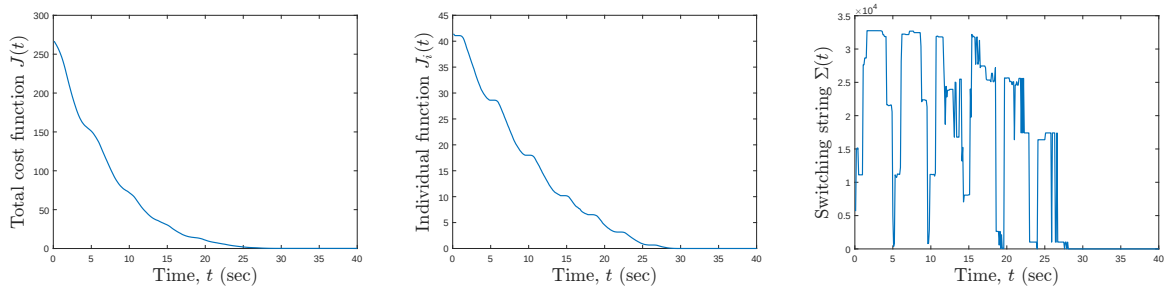


Figure 4.21: Cost function and switching string of robust distributed switching MPC

Due to the non-cooperativity of the algorithm, the conservativism is increased with respect

to the centralized case. This fact is more evident in an environment dense of obstacles.

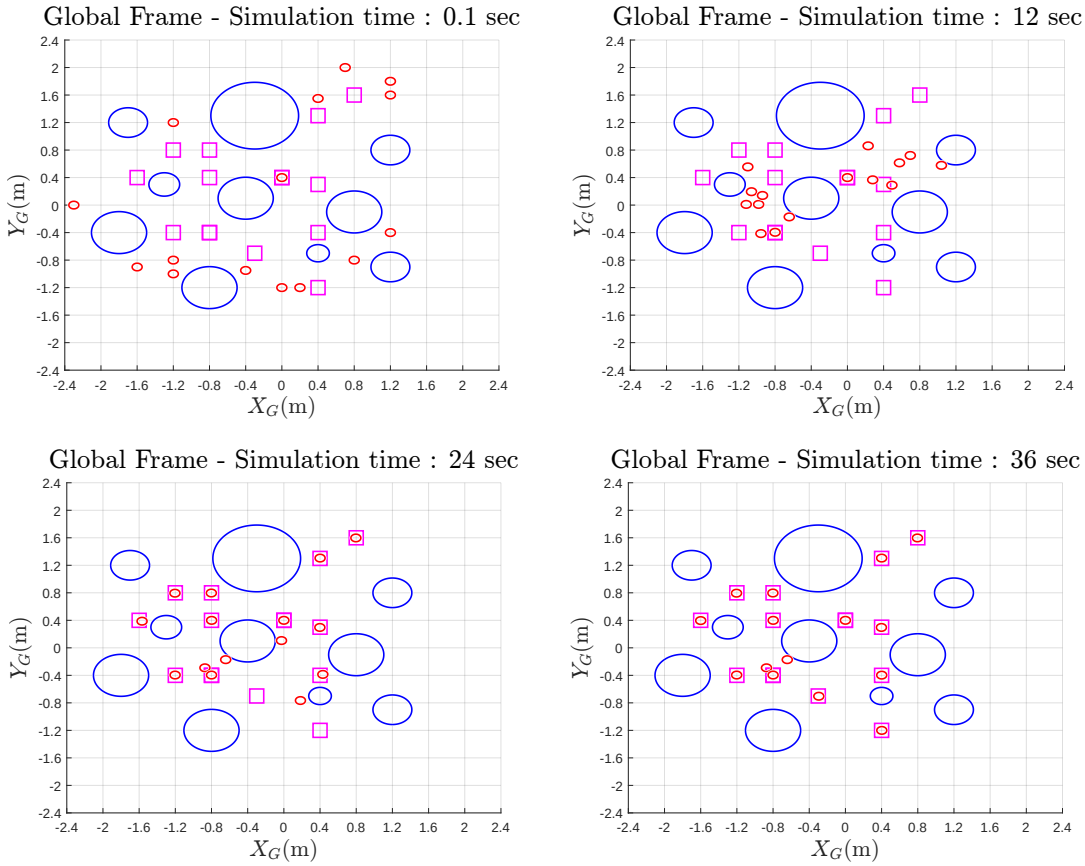


Figure 4.22: State of the robot network at different time instants

In Figure 4.22 almost all the network converge to the desired reference position. Some robots do not reach the desired reference position due to the presence of obstacles and other robots that obstruct their trajectory.

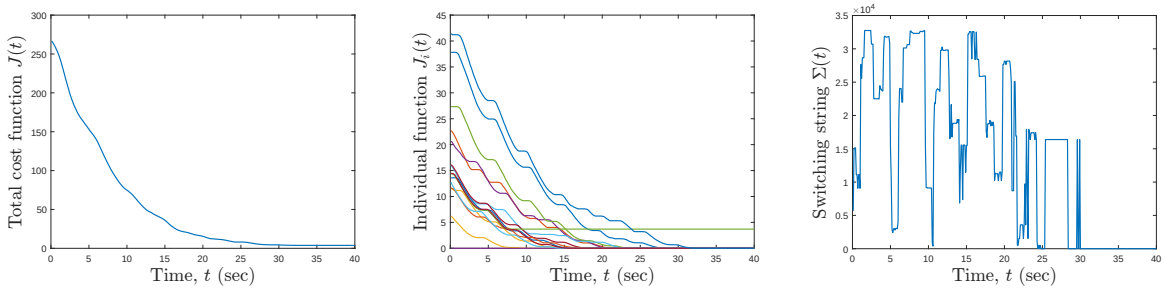


Figure 4.23: Cost function and switching string of robust distributed switching MPC

4.5. Stochastic switching MPC

The robust control laws described in the previous section are quite conservative, being based on the worst-case scenario. This could lead to quite low performance, in case of high uncertainty, since the controller could decide that the best solution is to not attempt any movement.
In order to reduce the conservativism of the approach, we propose a reformulation in a stochastic framework. For a general survey on stochastic MPC see [48, 49] where both the linear and nonlinear cases are considered. In the following, a similar reformulation of the constraint as done in [50] is adopted. Another possible approach is given by the scenario approach, see [35, 51]; however, its sampling-based nature combined with the already computationally demanding approach could lead to an excessive computational burden.

4.5.1. Expected cost function and chance-constraint

In this section, model (4.31) will be considered, with the fundamental assumption on the disturbances. In particular, the disturbances $d_x(k), d_y(k)$ are considered to be zero-mean white noise with covariance Σ_D and possibly unbounded support.
As common in stochastic MPC formulation, the cost function is the expectation of the original cost J over the distribution of the disturbance.

$$J_s = \mathbb{E}[J] = \mathbb{E}[\sum_{k=0}^{N_p} l(p(k), \bar{p})] \tag{4.37}$$

Similarly to 4.5:

$$\begin{aligned} J_s &= \sum_{i=1}^{N} \mathbb{E}[\bar{J}^{[i]}] \\ \mathbb{E}[\bar{J}^{[i]}] &= \sum_{k=0}^{N_p} \mathbb{E}[\bar{l}(p^{[i]}(k), \bar{p}^{[i]})] \\ \mathbb{E}[\bar{l}(p^{[i]}(k), \bar{p}^{[i]})] &= \mathbb{E}[(p_x^{[i]}(k) - \bar{p}_x^{[i]})^2 + (p_y^{[i]}(k) - \bar{p}_y^{[i]})^2] \end{aligned} \tag{4.38}$$

which can be easily expressed in terms of expected value $\mu_x^{[i]}, \mu_y^{[i]}$ and variance $\Sigma_x^{[i]}, \Sigma_y^{[i]}$ of p_x, p_y as:

$$\mathbb{E}[\bar{l}(p^{[i]}(k), \bar{p}^{[i]})] = (\mu_x^{[i]}(k) - \bar{p}_x^{[i]})^2 + (\mu_y^{[i]}(k) - \bar{p}_y^{[i]})^2 + \Sigma_x^{[i]}(k) + \Sigma_y^{[i]}(k) \tag{4.39}$$

The mean and variance can be easily derived from model (4.31). Having the disturbances an expected value equal to zero, the expected value $\mu_x^{[i]}, \mu_y^{[i]}$ can be directly computed from the nominal model (4.1). Furthermore, being the considered disturbances white noise signals the variance can be computed as:

$$\begin{aligned} \Sigma_x^{[i]}(k+1) &= \Sigma_x^{[i]}(k) + \sigma^{[i]}(k)\Sigma_D \\ \Sigma_y^{[i]}(k+1) &= \Sigma_y^{[i]}(k) + \sigma^{[i]}(k)\Sigma_D \end{aligned} \tag{4.40}$$

where $\Sigma_x(0), \Sigma_y(0)$ depends on the initial uncertainty of the robot position and Σ_D is the covariance of the disturbances d_x, d_y.

Another possibility to reduce conservativism is to use chance constraints, in other words, constraints that constrain the probability of violation.
Consider the chance constraint formulation of the obstacle avoidance constraint (4.7):

$$\Pr(H_n p^{[i]}(k) \leq S_n) \leq \epsilon \tag{4.41}$$

where ϵ is a design parameter that represents the maximum probability of violation allowed and has to be tuned to obtain a tradeoff between performance and constraint satisfaction. Constraints of this type can be reformulated in a deterministic framework resorting to the Cantelli inequality [50, 52] as:

$$H_n \mu_p^{[i]}(k) \geq S_n + f(\epsilon)\sqrt{H_n \Sigma_p^{[i]}(k) H_n^\top} \tag{4.42}$$

Where

$$f(\epsilon) = \sqrt{\frac{1-\epsilon}{\epsilon}} \tag{4.43}$$

In case the disturbances are distributed according to a Gaussian probability distribution it is possible to set, for $\epsilon \in (0, 0.5]$

$$f(\epsilon) = \mathcal{N}^{-1}(1-\epsilon) \tag{4.44}$$

Where $\mathcal{N}$ is the cumulative probability function of a Gaussian variable with zero mean and unitary variance. It can be verified that $\sqrt{\frac{1-\epsilon}{\epsilon}} \geq \mathcal{N}^{-1}(1-\epsilon)$, therefore if information about the Gaussianity of the disturbances is available, the less conservative solution should be used.
The inter-robot collision avoidance constraint can be considered in a similar way, considering the total uncertainty related to the robots; therefore, in (4.42) the variance

$\Sigma_p(k) = \Sigma_p^{[j]}(k) + \Sigma_p^{[i]}$, should be used.
The proposed stochastic formulation for the cost function and constraints helps to reduce the conservativism present in the robust formulation described in the previous section; however, it will be shown that the analysis of the algorithm becomes considerably more complex, especially due to the presence of chance constraints.
A possibility to avoid this complication could be to consider a stochastic formulation for the cost function while maintaining robust constraint satisfaction.

4.5.2. Stochastic centralized switching MPC

Before presenting the stochastic switching MPC algorithm, some general considerations concerning stochastic MPC are needed. In the context of stochastic MPC the problem of guaranteeing recursive feasibility is not trivial to solve, some approaches are based on imposing suitable mixed probabilistic/worst case constraint tightening [53], while others rely on a more flexible definition of the initial state of the optimal control problem at the price of disregarding part of the current data [48], which is the strategy that will be adopted. In this type of strategy, the initial state, formed by the pair of initial pose and variance $(p(0), \Sigma_p(0))$, is taken as a free variable and is free to take two different values: the current measured state of the system $(p(k), 0)$ or the predicted state and variance of the system $(p(k|k-1), \Sigma_p(k|k-1))$. In [48] three possible schemes are introduced:

1. Hybrid scheme: select the initialization strategy that, besides feasibility, guarantees the minimization of the cost function.
2. Nominal scheme: For all time instants, select the nominal strategy $(p(k|k-1), \Sigma_p(k|k-1))$.
3. Reset-based scheme: For all time instants, select the reset strategy,$(p(k), 0)$, if feasible, otherwise select the nominal one.

The three schemes offer different probabilistic characterization of the method. The nominal scheme allows to verify the fulfillment of the 'non-conditional' expectation constraint at each time instant, $\Pr(H_n p^{[i]}(k) \leq S_n | p^{[i]}(0)) \leq \epsilon$. The reset scheme guarantees that $\Pr(H_n p^{[i]}(k) \leq S_n | p^{[i]}(k)) \leq \epsilon$ is verified at all time instants when the feasibility of the reset strategy is verified, when not verified allows to guarantee the conditional expectation constraint with respect to the last feasible measurement. Hybrid schemes are optimal in terms of the cost function minimization.
In the following, we will exploit the hybrid scheme with a slight modification:

- Modified hybrid scheme: select the reset strategy if it is feasible that leads to a

decrease of the cost function with respect to the previous time instant, otherwise use the nominal strategy. Furthermore, if convergence of the nominal strategy is detected, the reset strategy is forced.

This choice offers a good trade-off between the exploitation of the information available and the minimization of the cost function. Therefore, at each time instant, the algorithm to be executed in order to compute the switching signal to apply is the following

Algorithm 4.7 Stochastic centralized switching MPC

1: **Input:**actual pose $p(k)$, predicted pose $p(k|k-1)$ and predicted variance $\Sigma_p(k|k-1)$, and previous value of the cost function $J_s(k-1)$
2: **Output:** First element of the optimal switching sequence Σ
3: **Foreach**$\Sigma_i \in \mathbb{W}$
4: Evaluate the associated cost J_{si} and the constraint tightening, through simulation of the system using the switching sequence Σ_i and considering as initial position the actual pose $p(k)$ and zero variance.
5: **IF** Σ_i does not violate the tightened restriction
6: Add i to the set of feasible sequence $\mathbb{I}$
7: **end**
8: Compute the index of the optimal solution:$i^{opt} = \underset{i\in\mathbb{I}}{argmin}\ J_{si}$. In case of multiple sequences with equal cost, consider the one that involves the minimum amount of movement for each robot.
9: **If** $J_{si^{opt}} \leq J_s(k-1)$ **or** Convergence of the nominal state $p(k|k-1)$ is detected
10: **Return** The first element of the sequence Σ_i
11: **Else**
12: Repeat points 3-8 considering the predicted pose $p(k|k-1)$ and the predicted variance $\Sigma_p(k|k-1)$ as the initial pose.
13: **Return** The first element of the sequence Σ_i

For the proposed Algorithm, it is possible to prove the following result.

Proposition 4.2. *Assume that the initial state $p(0)$ belongs to the set of states where Algorithm 4.7 has a solution. Then, the equilibrium point $\bar{p}$ defined by the reference signal N_{rob} is stable in probability, and the trajectories converge to local minima.*

Proof. The proof starts by showing that the algorithm ensures recursive feasibility in a similar way as in (4.1), considering that at least a feasible solution exists and by extending the nominal trajectory as in (4.11). Proving stability in probability is more challenging, and two aspects need to be considered:

1. When the modified hybrid scheme selects the nominal strategy with similar reasoning as in Theorem 4.1 it is possible to prove that $\Delta J_s \leq 0$. Indeed, when the reset strategy is chosen due to a possible decrease in cost trivially $\Delta J_s \leq 0$ is ensured.

2. When the reset strategy is chosen due to convergence of the nominal strategy, an increase of the cost function is allowed. However, since the nominal strategy has reached a local minimum it is ensured that:

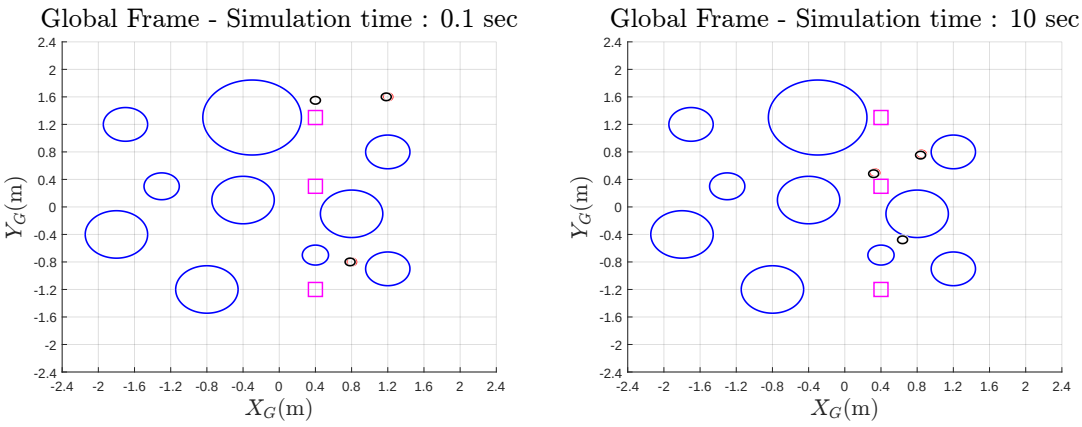
$$\mathbb{E}[l(p(\tau_2))] \leq l(p(\tau_1)) \tag{4.45}$$

   Here τ_1, τ_2, with$\tau_2 \geq \tau_1$ are two generic time instants where the reset strategy is used.

The relation, (4.45), allows us to characterize in a stochastic framework the evolution of the system. Furthermore, considering the function $l(p(\tau_1))$ as a Lyapunov function makes possible to prove stability in probability, see [54] for more detail on the stability of stochastic discrete-time system. The proposed algorithm is computationally more expensive than the other algorithm described previously, however, the parameter ϵ, which represents the maximum allowed probability of constraint violation, acts as a tuning knob in order to achieve a trade-off between conservativism and constraint satisfaction.

Simulation results

In order to study the performances of stochastic switching-MPC the same scenario as in Section 4.4.2 will be considered. Therefore, a network composed of $N_{rob} = 3$ mobile robots and parameter $N_p = 2, T = 0.1$ is considered. In order to better compare with the simulation in Section 4.4.2 a disturbance standard deviation equal to $\Sigma_D = \frac{a_1 T}{6}$ is considered with a maximum allowed probability of violation $\epsilon = 0.01$. In Figure 4.24 the red circle represents the actual position of the robot, the blue circle represents the obstacle, and the black circle represents the expected position of the robots.



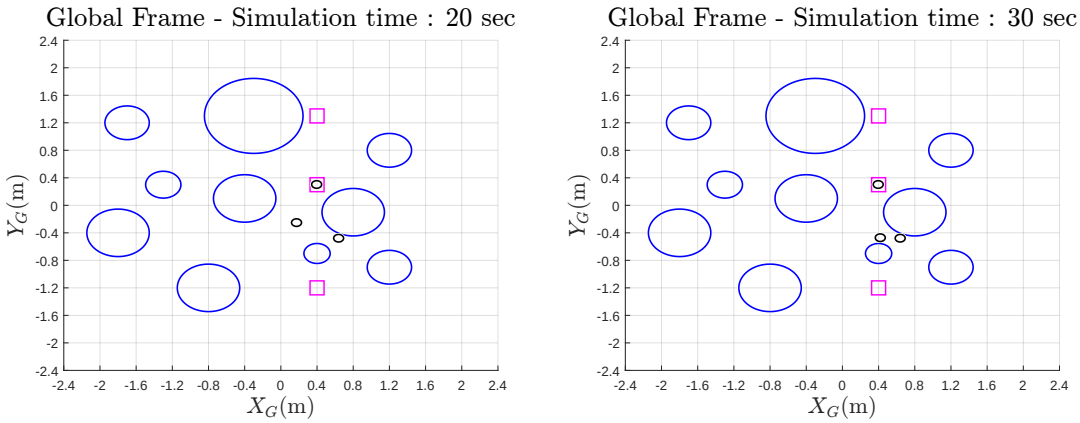


Figure 4.24: State of the robot network at different time instants

Being the allowed probability of violation of constraint very small the algorithm is very conservative.

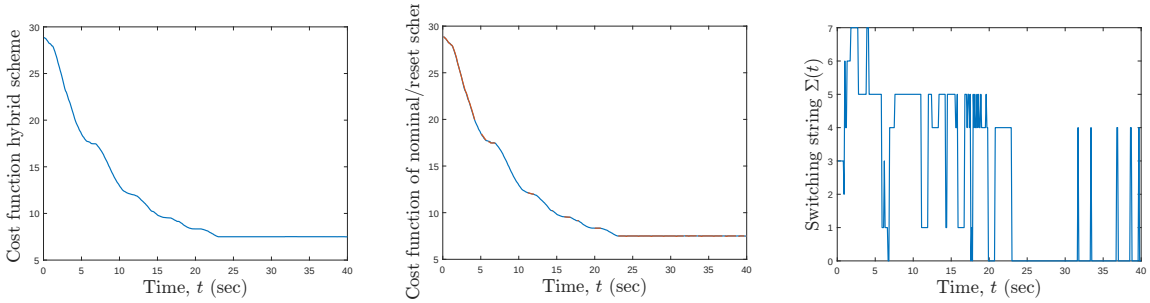
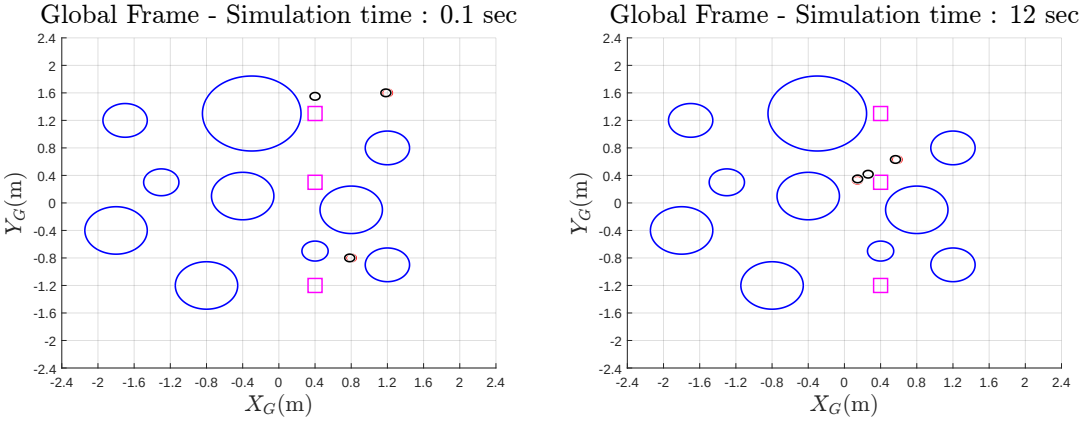


Figure 4.25: Cost function and switching string of stochastic switching MPC

Unlike the robust algorithm, it is possible to achieve a trade-off between conservativism and performance by tuning the value of ϵ. To this aim, for the next simulation, the value $\epsilon = 0.1$ is considered.



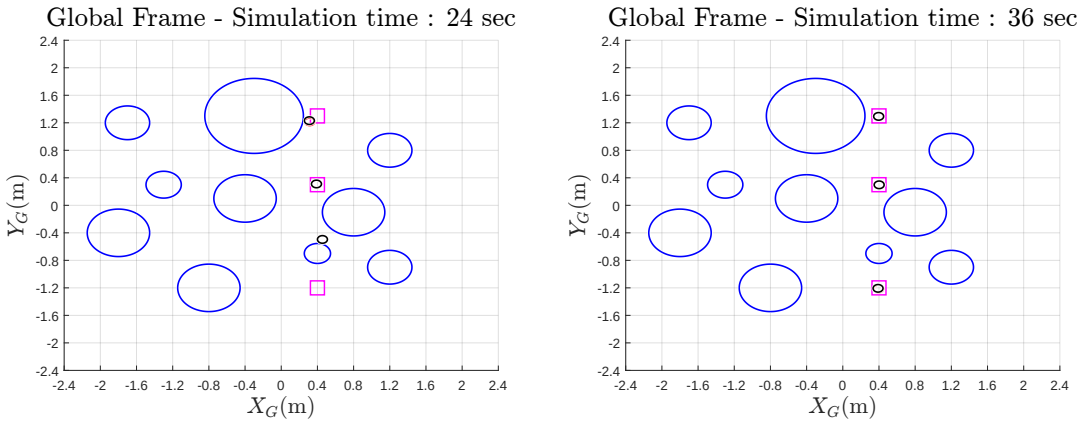


Figure 4.26: State of the robot network at different time instants

Allowing more freedom to the network the proposed control law is capable of steering each robot to the desired reference position.

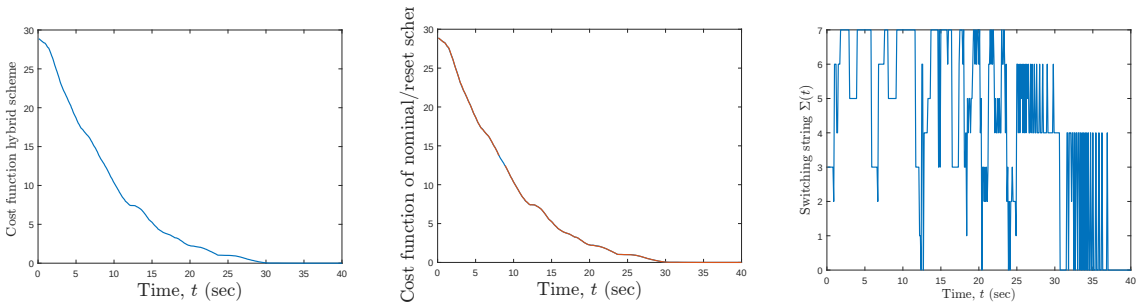


Figure 4.27: Cost function and switching string of stochastic switching MPC

4.5.3. Stochastic distributed switching MPC

A distributed formulation for the proposed stochastic switching MPC can be obtained along the lines of the procedure for robust distributed implementation, with mild modification. In more detail:

- Consider the expected value of the cost function J_s described in (4.38) in place of the original cost function J
- Consider the modified hybrid scheme described in the previous scheme for the initialization in the optimization phase
- Consider for obstacle avoidance the chance constraint (4.42)
- Regarding inter-robot collision avoidance, similarly to the distributed case, the robot

needs to communicate besides the predicted trajectory, also the related variance. Then, use Algorithm 4.4 to obtain the distributed reformulation of the chance constraint.

Similar result can be proven to the distributed nominal case described as in Theorem 4.1.

Simulation results

In order to study the performances of distributed switching MPC a network composed by $N_{rob} = 15$ mobile robots with parameters $N_p = 3, T = 0.1, \epsilon = 0.1$ is considered, and as in the case of centralized stochastic switching MPC a standard deviation of the disturbances equal to $\Sigma_D = \frac{a_1 T}{6}$ is also considered.

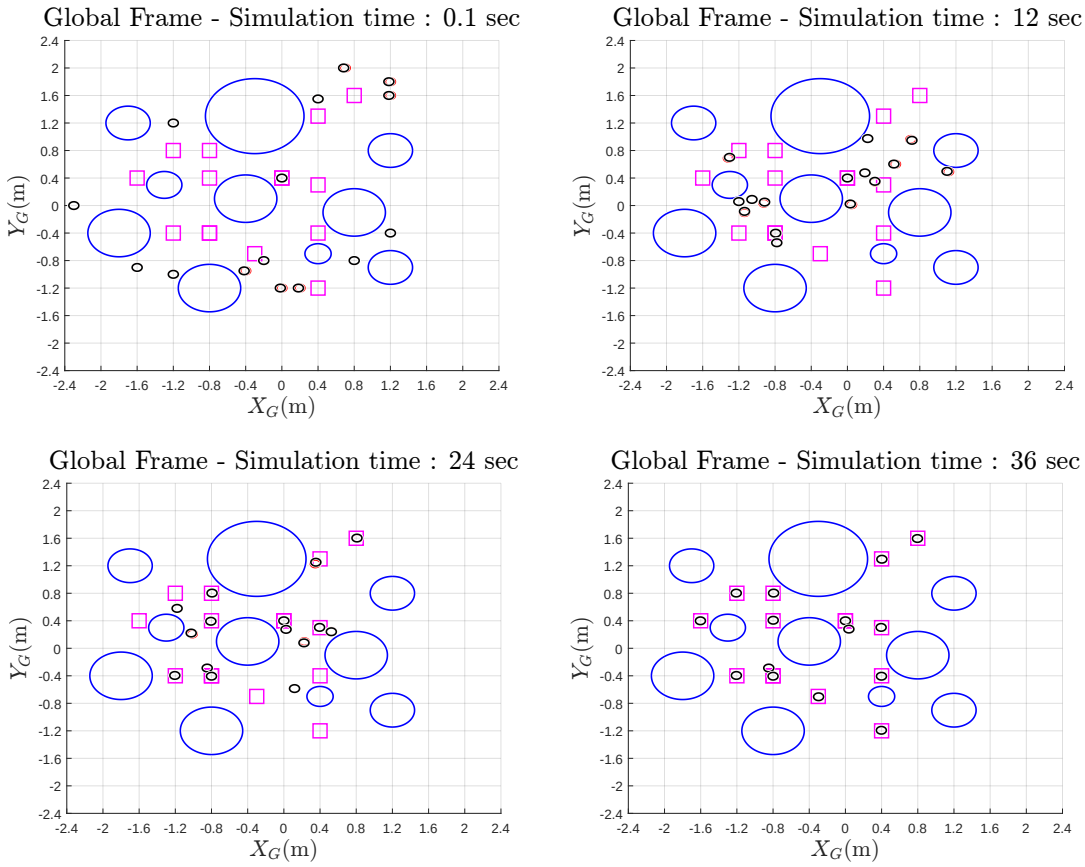


Figure 4.28: State of the robot network at different time instants

The proposed algorithm is able to steer the network to the desired reference position without particular difficulties.

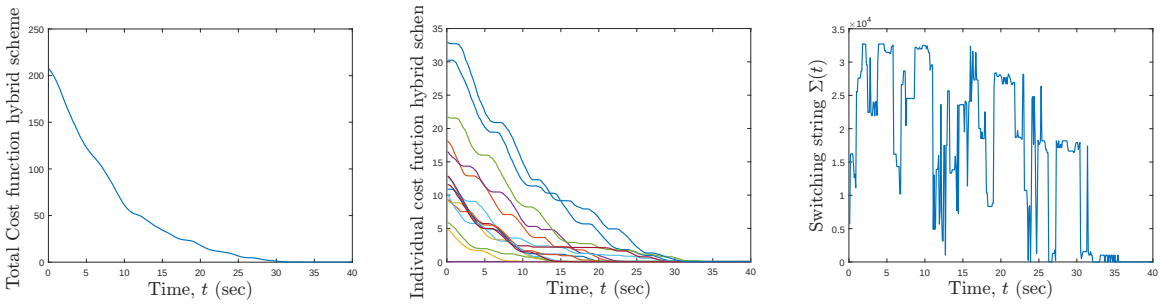


Figure 4.29: Cost function and switching string of distributed stochastic switching MPC

4.6. Comments

The algorithms described in this chapter offer a feasible approach for navigation and coordination, both in a nominal situation and in a situation where uncertainty is present. Despite the great reduction of the computational effort in the case of distributed implementation, the optimization problems to be solved are still quite demanding. In order to alleviate this limitation, in the next section, some algorithms which do not present this problem will be presented. Furthermore, even if not considered in the chapter, it is also possible to include in the analysis without particular difficulties the presence of measurement noise by acting on the initial uncertain set in the case of the robust formulation or the initial variance in the case of the stochastic formulation.

5 | Tube-based switching MPC

In this chapter, a robust tube-based MPC control law will be developed. Differently from the previous chapter, the MPC control law in this section takes advantage of the knowledge of a stabilizing switching law to maintain the trajectories of the robot in a tube centered around a nominal reference trajectory. This fact motivates the use of the nomenclature tube MPC, since it presents similarities to general tube-based approach available in the literature.

5.1. Rigid tube

For the algorithm in this chapter, the discretized version of the model with additive disturbances (2.17) will be considered:

$$p^{[i]}(k+1) = f_{\sigma(k)}(p^{[i]}(k)) \tag{5.1}$$

where:

$$f_0 = \begin{bmatrix} p_x^{[i]}(k) + d_x(k) \\ p_y^{[i]}(k) + d_y(k) \\ \theta^{[i]}(k) + T\omega_0 \end{bmatrix} \quad f_1 = \begin{bmatrix} p_x^{[i]}(k) + v_1 \frac{\sin(\theta^{[i]}(k)+T\omega_1)-\sin(\theta^{[i]}(k))}{\omega_1} + d_x(k) \\ p_y^{[i]}(k) + v_1 \frac{-\cos(\theta^{[i]}(k)+T\omega_1)+\cos(\theta^{[i]}(k))}{\omega_1} + d_y(k) \\ \theta^{[i]}(k) + T\omega_1 \end{bmatrix} \tag{5.2}$$

It will be assumed that the disturbances satisfy the norm bounded condition $d_x^2(k) + d_y^2(k) \leq D_d^2$. Furthermore, it is assumed the knowledge of an auxiliary switching law $\bar{\sigma}(p^{[i]}(k), \bar{p}(k))$ capable of practically stabilizing a single robot and the knowledge of the related quantity R_M, see Chapter 3 for a more detailed discussion about stabilizing switching law and analysis of the resulting closed loop system. In particular, the switching signal to be applied will be calculated based on the switching law $\bar{\sigma}(p^{[i]}(k), \bar{p}(k))$, while the reference $\bar{p}(k)$ will be computed in a receding horizon fashion exploiting the maximum deviation of the trajectory R_M to robustly enforce the fulfillment of the constraint.
In practice, once the set of possible external disturbances characterized by D_d and the set of possible reference changes for the auxiliary switching law $\Delta\bar{p}^2(k) \leq D_p^2$ is defined, it

is possible to exploit the procedure in Chapter 3 to compute R_M from the knowledge of $D_M = D_d + D_p$. Then obstacle avoidance constraint and inter-robot collision avoidance can be managed as described in Chapter 4. Moreover, the constraint on the admissible value of $\Delta\bar{p}^2(k)$ can also be managed with the same polytopic approximation. Therefore, at each time instant $\bar{k}$ the switching signal to be applied can be computed by solving:

$$\begin{aligned}
&\underset{\Delta\bar{p}(k)}{min} \sum_{i=1}^{N_{rob}} \sum_{k=\bar{k}}^{N_p+\bar{k}} \bar{p}^{[i]\top}(k)\bar{p}^{[i]}(k) \\
&\bar{p}(k+1) = \bar{p}(k) + \Delta\bar{p}(k) \\
&H_n^j p^{[i]}(k) \geq S_n^j + R_M \quad \forall\, j = 1 \ldots N_{obs}\ , i = 1 \ldots N_{rob} \\
&H_n^j p^{[i]}(k) \geq S_n^j + 2R_M \quad \forall\, j = 1 \ldots N_{obs}\ , i = 1 \ldots N_{rob},\ j \neq i \\
&\Delta\bar{p}(k) \in \mathcal{D}_p
\end{aligned} \tag{5.3}$$

and evaluating:

$$\sigma^{[i]}(\bar{k}) = \bar{\sigma}(p^{[i]}(\bar{k}), \bar{p}(\bar{k})) \tag{5.4}$$

It is relevant to note that differently from the algorithms described in the previous chapter which involve the solution of an integer program, Problem 5.3 is a relatively simple quadratic programming problem, which is considerably easier to solve and its complexity scales better with both the length of the prediction horizon N_p and number of robots N_{rob}. The relatively low computational complexity of the approach allows to consider without difficulties a longer prediction horizon and a larger number of robots, with only the downside that considers as an effective radius of the robot the quantity $R_M + R_{rob}$. This fact can limit the ability of the network to navigate an environment of dense obstacles or a particularly crowded scenario. This limitation is even more evident if a distributed implementation, which can be obtained using the same approach described in Section 4.2.2 to handle constraint, is considered.

Simulation results

In order to study the performances, the proposed control strategy has been proved in a similar scenario as the ones considered in Chapter 4. To start with, we considered a centralized formulation for a network of $N_{rob} = 10$ mobile robots and $N_p = 5$. It has to be noted that the previously described algorithm would need to consider, during the optimization, a set of cardinality 2^{50} which is clearly not suitable to be solved at each time instant. Furthermore, we consider a sampling time $T = 0.1$, no disturbances present, and a value $D_p = 0.4a_1T$. In Figure 5.1 the state of the robot network at different time

instants is represented, where the red circle represents the robot position, black circle represents the circle of radius R_M where robots are ensured to be.

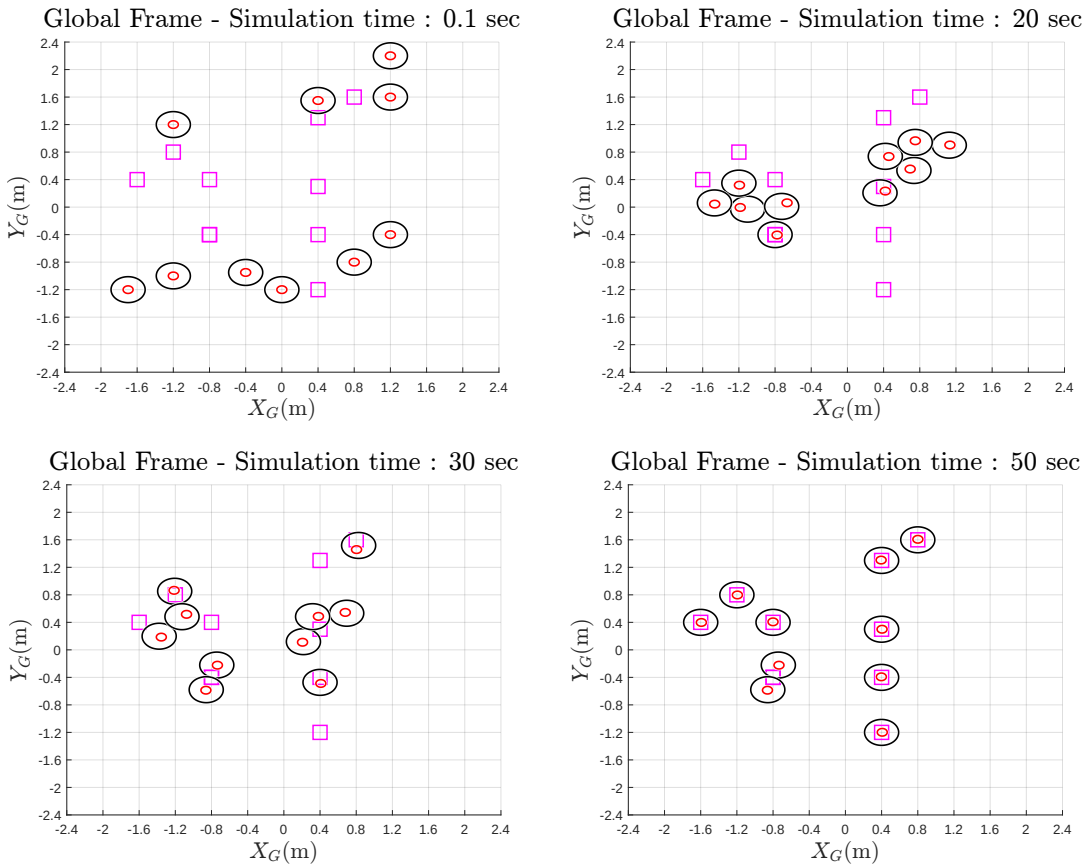


Figure 5.1: State of the robot network at different time instants

The robot network reaches the desired reference position while avoiding collision. It is also interesting to look at the behavior of the cost function.

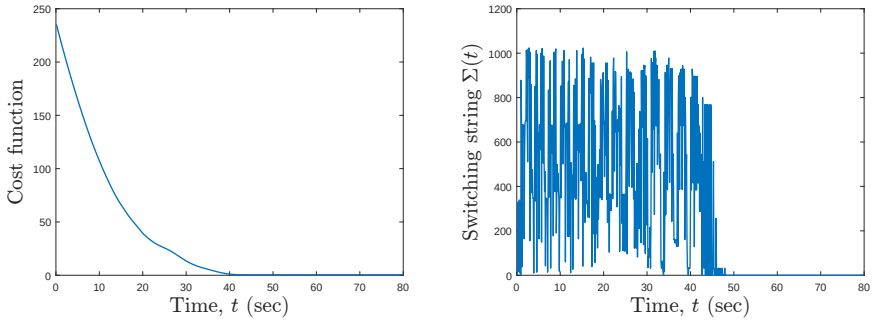


Figure 5.2: Cost function and switching string of centralized tube-based MPC

It is possible to see from Figure 5.1, that the effective radius $R_M + R_{rob}$ limits both the capability of the network to navigate and the allowed steady state position. This

limitation is enhanced by the presence of obstacles in the environment as can be seen in the next simulation where it is even chosen a smaller value of D_p, equal to $0.2a_1T$, to facilitate the navigation.

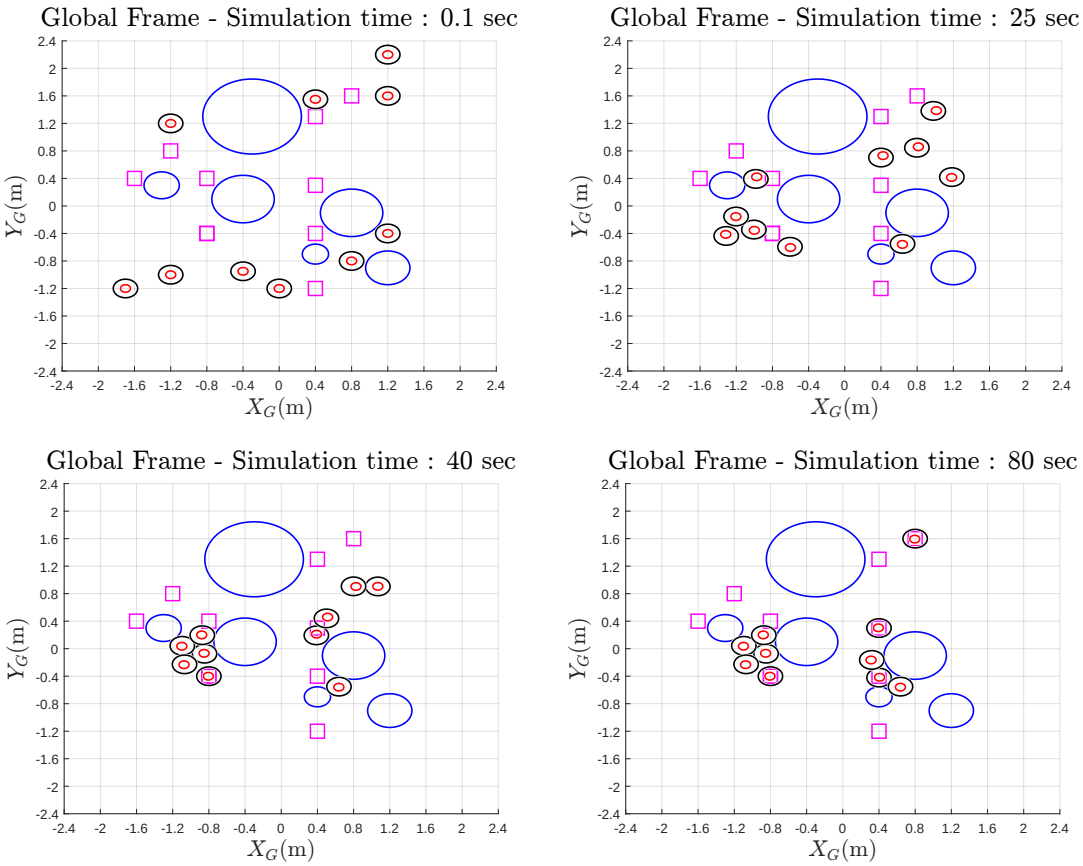


Figure 5.3: State of the robot network at different time instants

The sub-optimality of the proposed algorithm is evident. Even if part of the network reaches the desired position, several robots are unable to navigate in narrow passages only because of the effective radius $R_M + R_{rob}$.

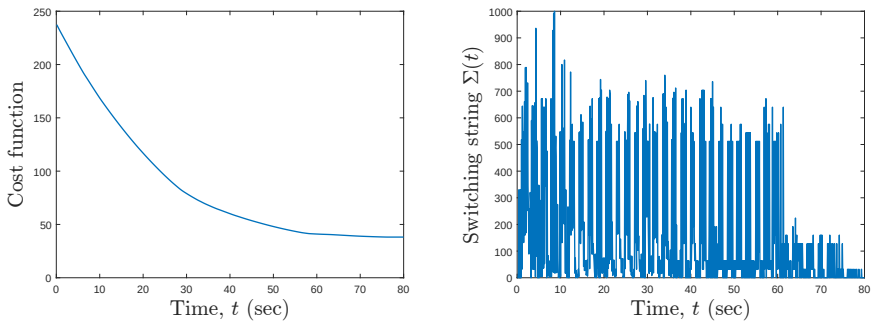


Figure 5.4: Cost function and switching string of centralized tube-based MPC

This motivates the development of the approach that will be described in the next section. Before discussing the next approach, it is shown a distributed implementation of the approach for the same scenario of Figure 5.1, with the presence of disturbances characterized by $D_d = 0.2a_1T$ and by a maximum reference change for the auxiliary law $D_p = 0.2a_1T$.

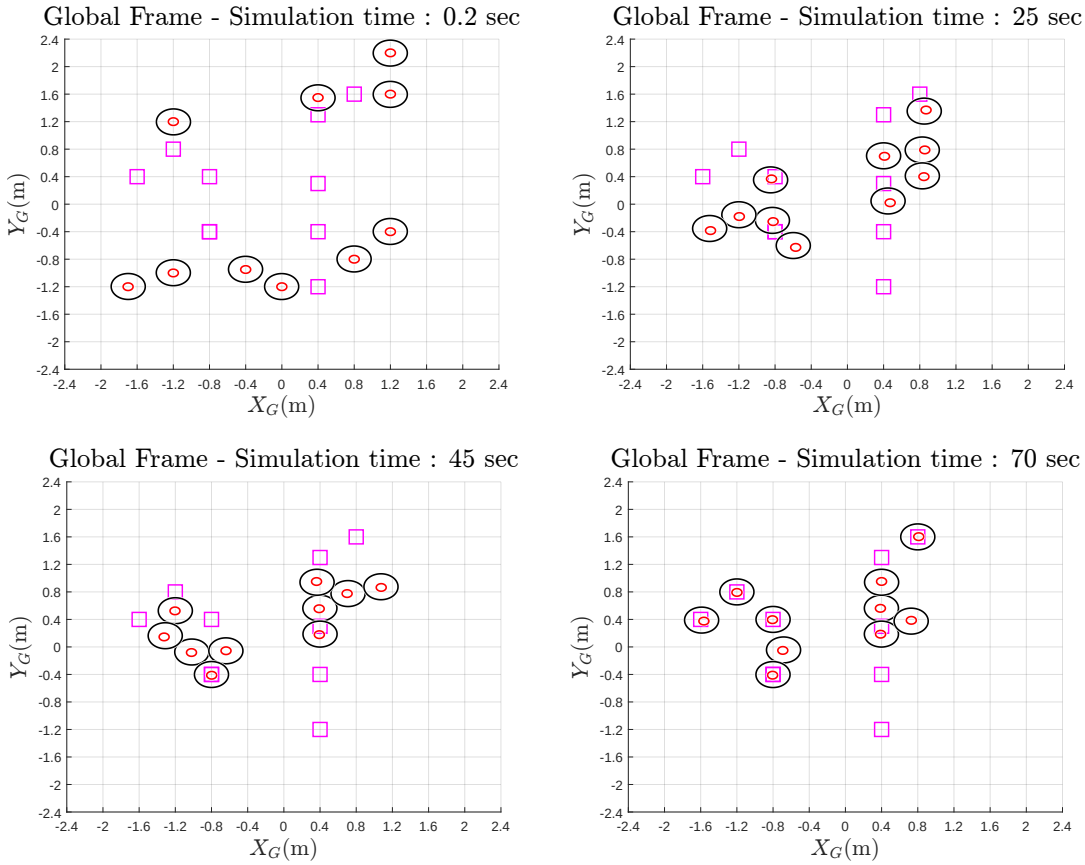


Figure 5.5: State of the robot network at different time instants

It is relevant to note that the non-cooperativity of the algorithm renders more difficult to reach the desired reference position. Moreover, the robots never reach a steady state, due to the presence of persistent disturbances, while the reference position for the auxiliary switching law reaches a steady state.

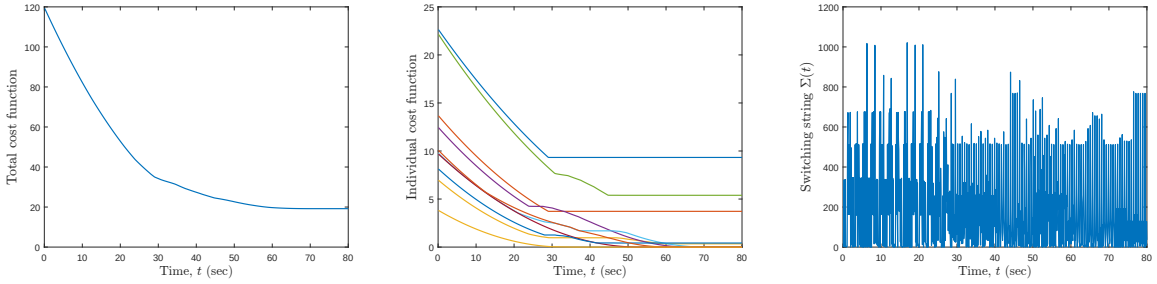


Figure 5.6: Cost function and switching string of distributed tube-based MPC

To solve (5.3) the software YALMIP has been used [55].

5.2. Time-varying tube

In order to increase the freedom of movement, it is possible to consider R_M as an optimization parameter along the prediction horizon. To this scope, it is possible to consider a linear upper bound on the value of R_M:

$$\begin{aligned} R_M(D_M) &= K_1 D_M + K_2 \\ D_M &= D_d + D_p(k) \end{aligned} \tag{5.5}$$

Using this upper limit, the quadratic Program (5.3) can be modified, allowing for a time-varying bound on the change of reference $\Delta\bar{p}(k)$ that corresponds to a time-varying constraint tightening R_M. In more detail, the quadratic Program (5.3) has to be modified by introducing additional constraints as in the following scheme:

- Constraint which defines the evolution of the time-varying constraint tightening

$$R_M(k+1) = K_1(D_d + D_p(k)) + K_2 \tag{5.6}$$

- R_M is not allowed to decrease along the prediction horizon to avoid situations where trajectories exit the tube

$$R_M(k+1) \geq R_M(k) \tag{5.7}$$

- the initial position must be compatible with the initial value of R_M

$$e_x^2(0) + e_y^2(0) \leq R_M(0) \tag{5.8}$$

  and a polytopic approximation can be used to obtain linear constraint.

- $D_p(k)$ must be upper bounded to avoid values for which it is not possible to ensure that the deviation from the center of the tube remains limited. Also, it is necessary to impose a lower bound greater than zero in order to avoid possible artificial deadlock.

$$0 < D_{MIN} \leq D_p(k) \leq D_{MAX} \tag{5.9}$$

With these modifications, the optimization problem law is allowed to act on the value R_M, resulting in a maximum freedom of motion when possible, and limiting it when a smaller

value of R_M is useful in a very obstacle-dense environment or very crowded scenarios. Furthermore, it is also possible to obtain a distributed implementation using the same approach described in Section 4.2.2.

Simulation results

In order to study the performances of the proposed approach, it is considered the same scenario with obstacles is considered in the previous section. Therefore, a network composed of $N_{rob} = 10$ mobile robots and $N_p = 5$ is considered. Furthermore no disturbances are present and $D_{MIN} = 0.2a_1T$, $D_{MAX} = 0.4a_1T$ are considered.

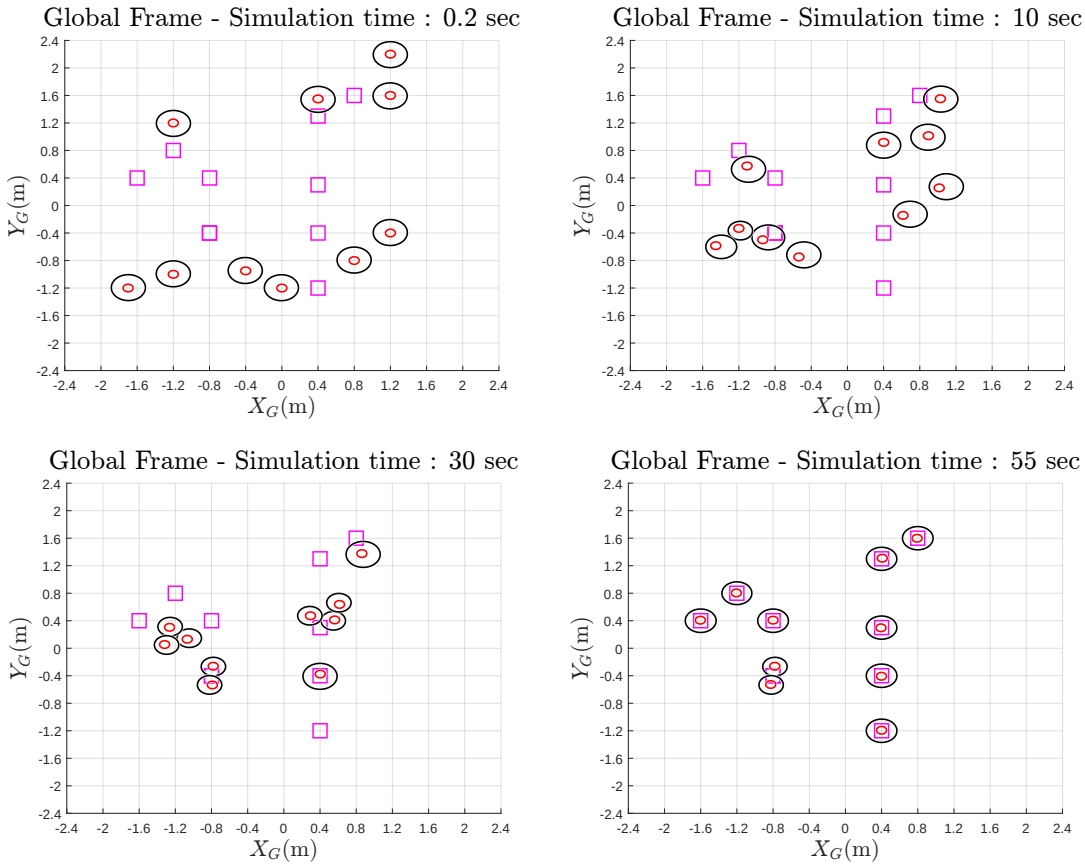


Figure 5.7: State of the robot network at different time instants

From Figure 5.7 it is clear that allowing a time-varying radius R_M can increase the capability of the network to navigate an environment rich of obstacles and particularly crowded scenarios. However, it is also evident how this approach can introduce a certain degree of sub-optimality, in particular allowing for a small value of R_M can hinder the performance of the network by limiting how large the change of references for the auxiliary control law can be. This drawback can be overcome by using a larger prediction horizon.

Even if the optimization problem at each time instant is relatively simple, the construction of the linear constraint is still a quite time-consuming task.

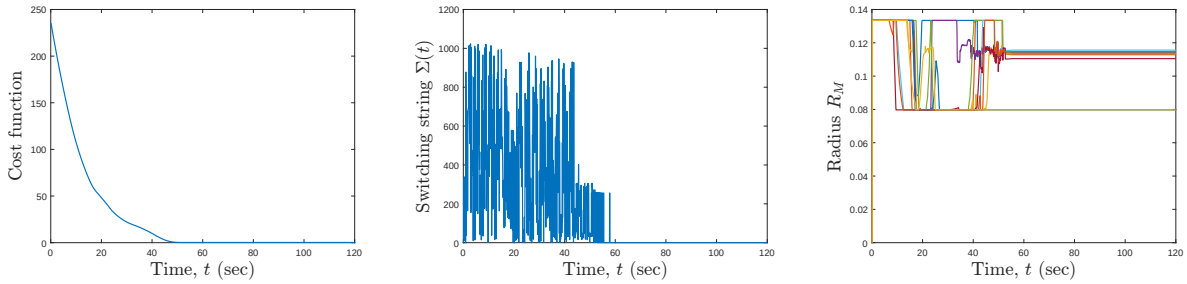
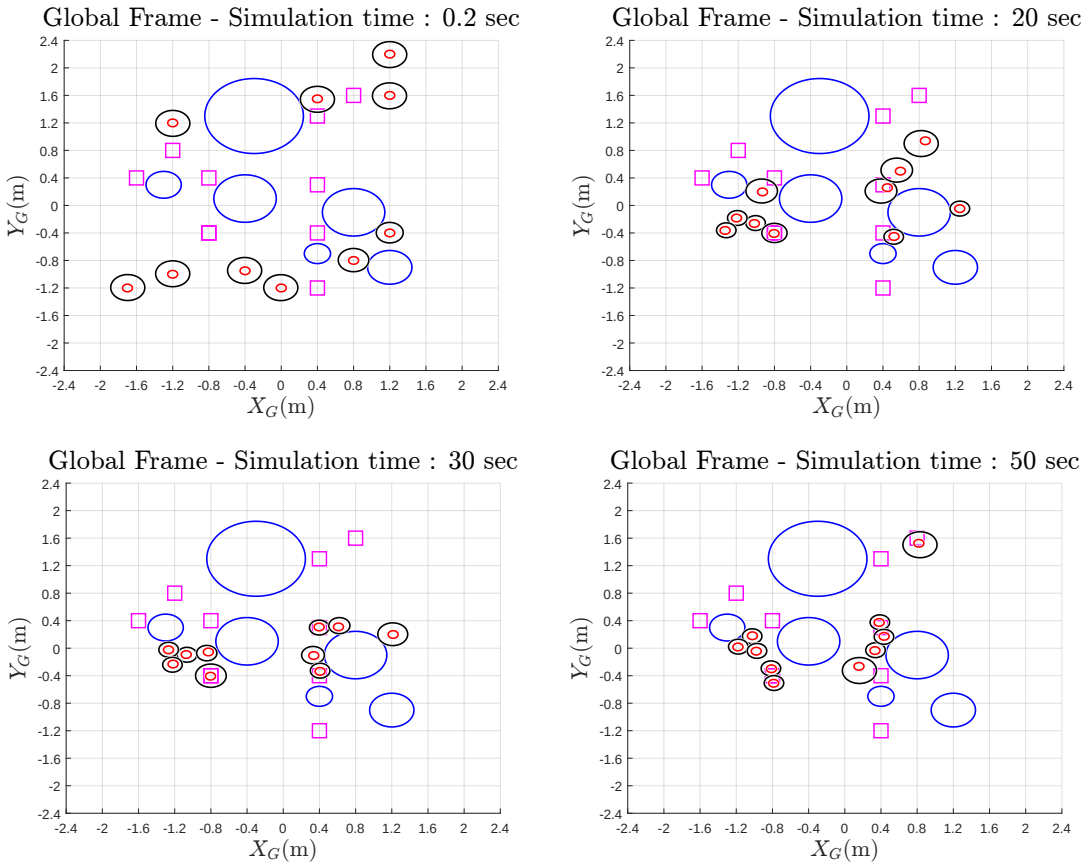


Figure 5.8: Cost function, switching string and radius R_M of modified centralized tube-based MPC

One of the advantages is that the network is more capable of navigating the environment when it is rich of obstacles as can be seen in the next simulation scenario.



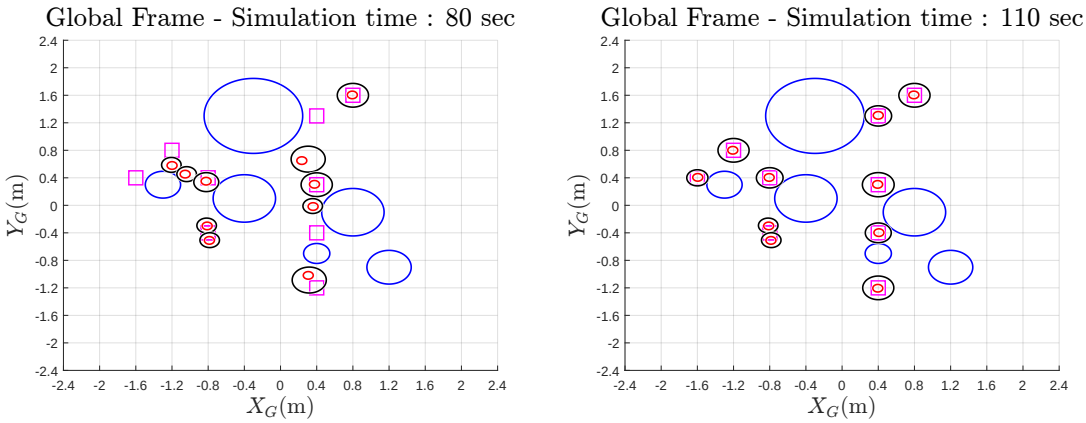


Figure 5.9: State of the robot network at different time instants

Differently from the approach of the previous section, with the additional degree of freedom R_M the network is capable of navigating the environment and reaching the desired reference position.

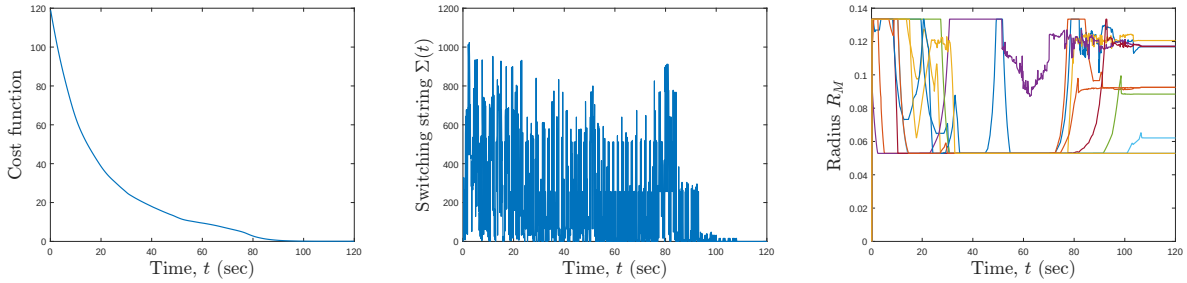


Figure 5.10: Cost function, switching string and radius R_M of modified centralized tube-based MPC

Next, a distributed implementation of the proposed algorithm is considered, in order to understand how the performances are affected by allowing for a time-varying radius R_M. To this extent, a network composed of a $N_{rob} = 10$ mobile robots is considered with a prediction horizon of length $N_p = 5$. Furthermore, presence of disturbances is also considered with maximum value $D_d = 0.2a_1T$ and value $D_{MIN} = 0.1a_1T$, $D_{MAX} = 0.4a_1T$.

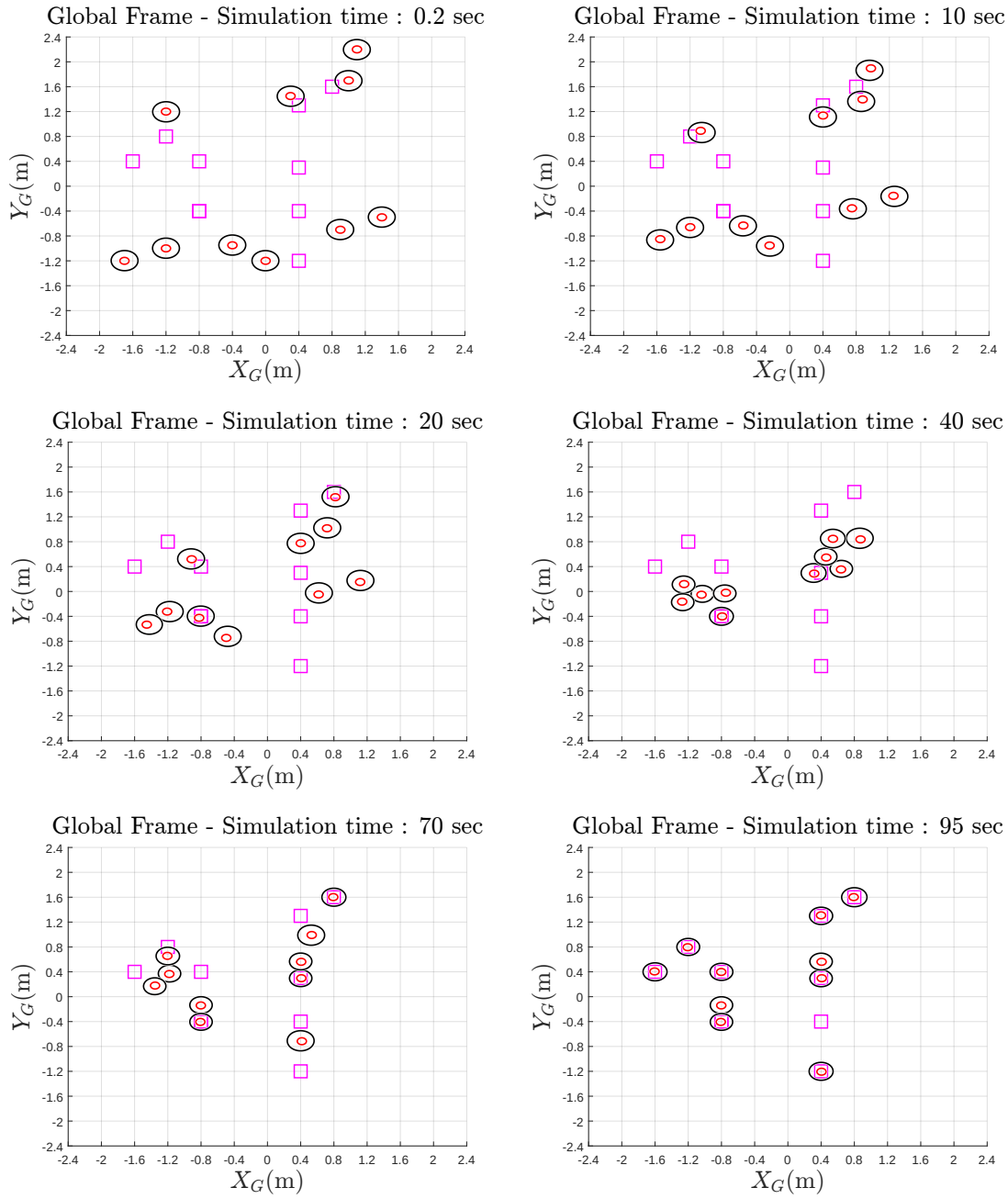


Figure 5.11: State of the robot network at different time instants

While the additional degree of freedom R_M helps the network to navigate the environment, it does not totally overcome the problem of artificial deadlock in distributed implementation. Therefore an approach like the temporary target shifting in [44] could be beneficial.

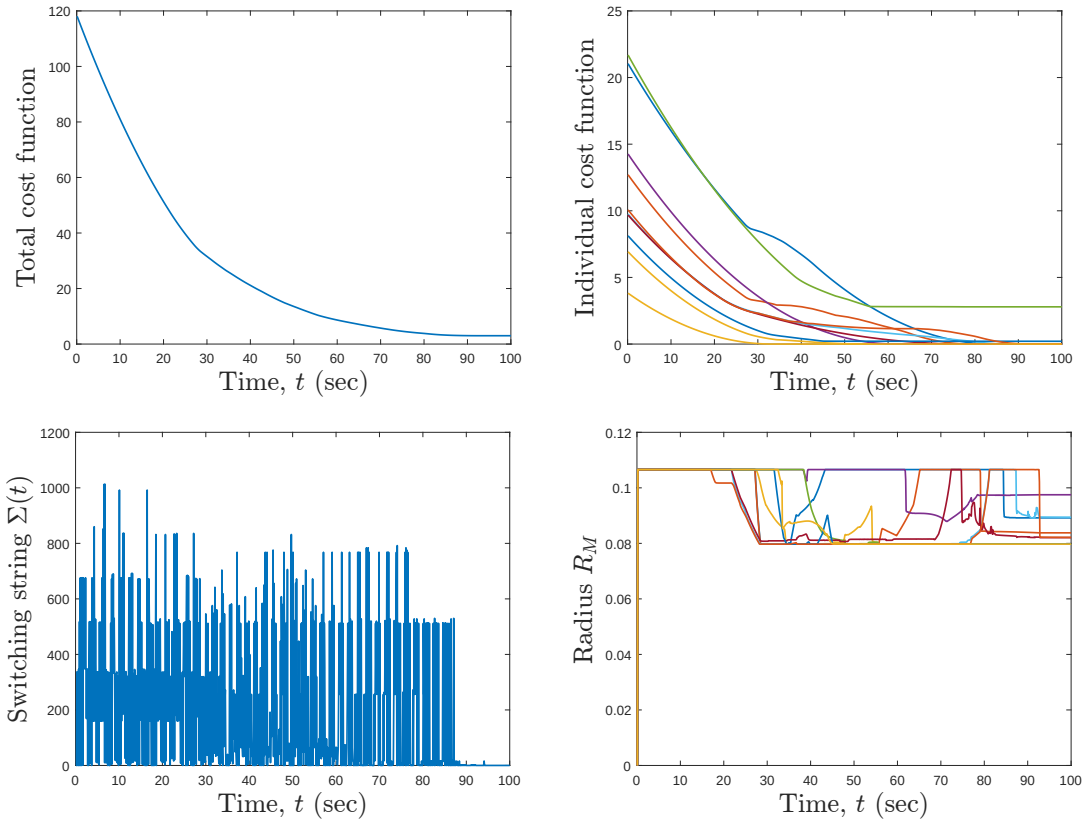


Figure 5.12: Cost function, switching string and radius R_M of modified distributed tube-based MPC

In Figure 5.12 it is possible to see that due to the presence of persistent disturbances, the robots never reaches a steady state. Furthermore due to the presence of disturbances the effective radius $R_M + R_{rob}$ is constrained to higher value with respect to previous simulations.

5.3. Comments

The approaches presented in this chapter, apart from being able to tackle a broader class of disturbances acting on the network, offer a suitable trade-off between the computational complexity required, quadratic programming instead of integer programming, and optimality of the obtained trajectories.

6 | Conclusions

The aim of this thesis was to develop control strategies for the trajectory tracking of a multiple mobile robot system in situations where disturbances are present. The approaches developed in this work rely on a switching model predictive control (MPC) strategy. These are based on a switching model of the plant dynamics in analogy with [14, 15], where a similar system setup is considered for a self-aggregation task and no disturbances acting on the system.

The first developed approaches recast the original problem into a finite horizon optimal control problem (FHOCP), relying on a discrete model for the system. Then, the FHOCP is used to formulate a switching-MPC strategy aimed at minimizing the difference between the position of the robot and the desired reference position. In order to consider the presence of disturbances, two different formulations have been proposed, one which ensures robustly the fulfillment of the constraints and one which exploits the stochastic framework to achieve a tradeoff between the conservativism of a robust formulation and satisfactory performances. The proposed approaches have also been extended to consider distributed formulations.

The last developed approaches are instead based on a deeper analysis of the underlying switched system. In particular, the approaches are based on disturbances reachable set. These approaches present similarities to the class of tube MPC strategies and offer major computational advantages with respect to them.

For all the control strategies presented in this thesis, several simulated scenarios have been illustrated, with the aim of highlighting the strengths and limitations of each approach.

Future developments can be done in many directions, such as considering more complex robot models or a greater number of allowed motions, or integratings the design of the proposed approaches with controllers located in an inner loop to limit the effect of disturbances.

Bibliography

[1] Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli. Control of wheeled mobile robots: An experimental overview. *RAMSETE: articulated and mobile robotics for services and technologies*, pages 181–226, 2002.

[2] Roger W Brockett et al. Asymptotic stability and feedback stabilization. *Differential geometric control theory*, 27(1):181–191, 1983.

[3] Jaydev P Desai, Jim Ostrowski, and Vijay Kumar. Controlling formations of multiple mobile robots. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 4, pages 2864–2869. IEEE, 1998.

[4] M Anthony Lewis and Kar-Han Tan. High precision formation control of mobile robots using virtual structures. *Autonomous robots*, 4:387–403, 1997.

[5] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on automatic control*, 51(3):401–420, 2006.

[6] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2):243–255, 2004.

[7] Veysel Gazi and Kevin M Passino. Stability analysis of swarms. *IEEE transactions on automatic control*, 48(4):692–697, 2003.

[8] N.E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, volume 3, pages 2968–2973 vol.3, 2001.

[9] Li Wang, Aaron D Ames, and Magnus Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[10] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. The nsb control: a behavior-based approach for multi-robot systems. *Paladyn, Journal of Behavioral Robotics*, 1(1):48–56, 2010.

[11] Marcello Farina, Andrea Perizzato, and Riccardo Scattolini. Application of distributed predictive control to motion and coordination problems for unicycle autonomous robots. *Robotics and Autonomous Systems*, 72:248–260, 2015.

[12] José C Geromel and Patrizio Colaneri. Stability and stabilization of continuous-time switched linear systems. *SIAM Journal on Control and Optimization*, 45(5):1915–1930, 2006.

[13] José C Geromel and Patrizio Colaneri. Stability and stabilization of discrete time switched systems. *International journal of control*, 79(07):719–728, 2006.

[14] CEP Yuca Huanca, Gian Paolo Incremona, Roderich Groß, and Patrizio Colaneri. Design of a switched control lyapunov function for mobile robots aggregation. In *ICINCO 2022: International Conference on Informatics in Control, Automation and Robotics*. SCITEPRESS Digital Library, 2022.

[15] Melvin Gauci, Jianing Chen, Wei Li, Tony J Dodd, and Roderich Groß. Self-organized aggregation without computation. *The International Journal of Robotics Research*, 33(8):1145–1161, 2014.

[16] Hongfei Sun and Jun Zhao. Control lyapunov functions for switched control systems. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 3, pages 1890–1891 vol.3, 2001.

[17] Patrizio Colaneri, José C Geromel, and Alessandro Astolfi. Stabilization of continuous-time switched nonlinear systems. *Systems & Control Letters*, 57(1):95–103, 2008.

[18] Daniel Liberzon. *Switching in systems and control*, volume 190. Springer, 2003.

[19] Antonio Russo, Gian Paolo Incremona, Alberto Cavallo, and Patrizio Colaneri. State dependent switching control of affine linear systems with dwell time: Application to power converters. In *2022 American Control Conference (ACC)*, pages 3807–3813. IEEE, 2022.

[20] Paolo Bolzern and William Spinelli. Quadratic stabilization of a switched affine system about a nonequilibrium point. In *Proceedings of the 2004 American Control Conference*, volume 5, pages 3890–3895. IEEE, 2004.

[21] Lucas N Egidio, Grace S Deaecto, and José C Geromel. Limit cycle global asymptotic stability of continuous-time switched affine systems. *IFAC-PapersOnLine*, 53(2):6121–6126, 2020.

[22] Lucas N Egidio, Helder R Daiha, and Grace S Deaecto. Global asymptotic stability of limit cycle and h2/h performance of discrete-time switched affine systems. *Automatica*, 116:108927, 2020.

[23] Lucas N Egidio and Grace S Deaecto. Novel practical stability conditions for discrete-time switched affine systems. *IEEE Transactions on Automatic Control*, 64(11):4705–4710, 2019.

[24] Graziano Chesi. Lmi techniques for optimization over polynomials in control: a survey. *IEEE transactions on Automatic Control*, 55(11):2500–2510, 2010.

[25] Pratik Biswas, Pascal Grieder, Johan Löfberg, and Manfred Morari. A survey on stability analysis of discrete-time piecewise affine systems. *IFAC Proceedings Volumes*, 38(1):283–294, 2005.

[26] Franco Blanchini, Stefano Miani, et al. *Set-theoretic methods in control*, volume 78. Springer, 2008.

[27] Sasa V Rakovic, Pascal Grieder, Michal Kvasnica, David Q Mayne, and Manfred Morari. Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 2, pages 1418–1423. IEEE, 2004.

[28] Alberto Bemporad, Fabio Danilo Torrisi, and Manfred Morari. Optimization-based verification and stability characterization of piecewise affine and hybrid systems. In *Hybrid Systems: Computation and Control: Third International Workshop, HSCC 2000 Pittsburgh, PA, USA, March 23–25, 2000 Proceedings 3*, pages 45–58. Springer, 2000.

[29] Riccardo Desimini and Maria Prandini. Robust bounded feasibility verification of piecewise affine systems via reachability computations. *IFAC-PapersOnLine*, 52(16):78–83, 2019.

[30] Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.

[31] Lukas Hewing, Andrea Carron, Kim P Wabersich, and Melanie N Zeilinger. On a correspondence between probabilistic and robust invariant sets for linear systems. In *2018 European Control Conference (ECC)*, pages 1642–1647. IEEE, 2018.

[32] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0.

In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, July 17–19 2013. `http://control.ee.ethz.ch/~mpt`.

[33] James Blake Rawlings, David Q Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

[34] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[35] Sasa V Rakovic and William S Levine. Handbook of model predictive control. 2018.

[36] Alberto Bemporad, Carlo Filippi, and Fabio D Torrisi. Inner and outer approximations of polytopes using boxes. *Computational Geometry*, 27(2):151–178, 2004.

[37] Raphael Cagienard, Pascal Grieder, Eric C Kerrigan, and Manfred Morari. Move blocking strategies in receding horizon control. *Journal of Process Control*, 17(6):563–570, 2007.

[38] Lars Grüne. Nmpc without terminal constraints. *IFAC Proceedings Volumes*, 45(17):1–13, 2012.

[39] Andrea Boccia, Lars Grüne, and Karl Worthmann. Stability and feasibility of state constrained mpc without stabilizing terminal constraints. *Systems & control letters*, 72:14–21, 2014.

[40] Panagiotis D Christofides, Riccardo Scattolini, David Munoz de la Pena, and Jinfeng Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.

[41] Arthur Richards and Jonathan P How. Robust distributed model predictive control. *International Journal of control*, 80(9):1517–1531, 2007.

[42] Paul Trodden and Arthur Richards. Distributed model predictive control of linear systems with persistent disturbances. *International Journal of Control*, 83(8):1653–1663, 2010.

[43] Paul Trodden and Arthur Richards. Cooperative distributed mpc of linear systems with coupled constraints. *Automatica*, 49(2):479–487, 2013.

[44] Danilo Saccani, Leonardo Cecchin, and Lorenzo Fagiano. Multitrajectory model predictive control for safe uav navigation in an unknown environment. *IEEE Transactions on Control Systems Technology*, 2022.

[45] Francesco Bullo. *Lectures on network systems*, volume 1. Kindle Direct Publishing Seattle, DC, USA, 2020.

[46] Francesco Bullo, Jorge Cortés, and Sonia Martinez. *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*, volume 27. Princeton University Press, 2009.

[47] Saša V Raković. Set theoretic methods in model predictive control. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, pages 41–54, 2009.

[48] Marcello Farina, Luca Giulioni, and Riccardo Scattolini. Stochastic linear model predictive control with chance constraints–a review. *Journal of Process Control*, 44:53–67, 2016.

[49] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016.

[50] Marcello Farina and Simone Misiano. Stochastic distributed predictive tracking control for networks of autonomous systems with coupling constraints. *IEEE transactions on control of network systems*, 5(3):1412–1423, 2017.

[51] Marco C Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149–157, 2009.

[52] Albert W Marshall and Ingram Olkin. Multivariate chebyshev inequalities. *The Annals of Mathematical Statistics*, pages 1001–1014, 1960.

[53] Mark Cannon, Basil Kouvaritakis, and Desmond Ng. Probabilistic tubes in linear stochastic model predictive control. *Systems & Control Letters*, 58(10-11):747–753, 2009.

[54] Wassim M Haddad and Junsoo Lee. Lyapunov theorems for stability and semistability of discrete-time stochastic systems. *Automatica*, 142:110393, 2022.

[55] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

List of Figures

List of Tables

List of Algorithms