Model Predictive Control Strategies For Constrained Unmanned Vehicles

Shima Savehshemshaki

A Thesis
in
Concordia Institute
for
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Master of Applied Science (Quality Systems Engineering) at
Concordia University
Montréal, Québec, Canada

December 2020

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By:	**Shima Savehshemshaki**
Entitled:	**Model Predictive Control Strategies For Constrained Unmanned Vehicles**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. Mohsen Ghafouri

______________________________ Examiner
Dr. Shahin Hashtrudi Zad

______________________________ Examiner
Dr. Mohsen Ghafouri

______________________________ Thesis Supervisor
Dr. Walter Lucia

Approved by ______________________________
Dr. Mohammad Mannan, Graduate Program Director

December 4, 2020 ______________________________
Dr. Mourad Debbabi, Interim Dean
Gina Cody School of Engineering and Computer Science

Abstract

Model Predictive Control Strategies For Constrained Unmanned Vehicles

Shima Savehshemshaki

This thesis deals with two control problems for unmanned vehicles, namely battery shortage prevention for electric unmanned vehicles and collision avoidance strategy for constrained multi unmanned vehicle systems.

In the battery shortage prevention problem, we design a novel control architecture, equipped with a battery manager module, capable of avoiding energy shortage by appropriately imposing time-varying upper bounds on the vehicle's maximum acceleration. Here, the dual-mode control paradigm known as set-theoretic model predictive control is applied to couple the reference tracking and the battery shortage problems. First, we offline design a conservative maximum acceleration profile capable of assuring that the electric unmanned vehicle will reach the desired target without incurring into a battery shortage along the given path. Then, online, by following a receding horizon approach, we show that the battery manager can enhance the performance by using the current battery's state-of-charge. Moreover, a simulation example is presented to clarify and show the proposed control framework's potential and features.

In the collision avoidance problem, we deal with vehicles moving in a shared environment where each UV follows a trajectory given by a local planner. We assume that the planners are uncoordinated and each vehicle is subject to different constraints and disturbances. In this context, we design a new centralized traffic manager that, in conjunction with ad-hoc designed local model predictive controller, can ensure the absence of collisions while minimizing the total vehicle's stops occurrences. In particular, in a receding horizon fashion, the traffic manager exploits available previews on the successive vehicle's waypoints to speed-up or speed-down the vehicles and minimize the chance of collisions. Moreover, by exploiting basic set-theoretic arguments, traffic manager can impose a vehicle to stop and safely prevent collisions whenever necessary. Finally, two different simulation examples are presented to better illustrate the capability of the proposed solution.

Acknowledgments

I would like to express my gratitude and sincere appreciation to my supervisor Dr. Walter Lucia for his immense help and support throughout my master's degree. This thesis would not be possible without his continuous guidance and insightful comments. I cannot be more thankful to have such a knowledgeable, considerate, and outstanding supervisor who convincingly guided me to move in the right and professional direction.

I wish to thank my friends and colleagues at Concordia University: Maryam Bagherzadeh, Kian Gheitasi, Mohsen Ghaderi, Mahsa Mesgaran, Vahid Khorasani, Kiarash Aryankia who walked by my side during the last two years, shared my moments of distress and joy and made this period the most memorable one in my life.

Last but not least, nobody has been more important to me in the pursuit of this thesis than my family. I want to thank my husband, Amirreza Mousavi, who none of this would have been possible without his encouragement and help. I must express my heartfelt appreciation to my beloved family, my parents and parents-in-law, my sister Maryam, my brother Amirreza and my brother-in-law Alireza, whose love and guidance are with me in whatever I pursue. This accomplishment would not have been possible without them.

Contents

List of Figures

List of Tables

List of Abbreviations

MPC:	Model Predictive Control
ST-MPC:	Set-Theoretic Model Predictive Control
DMPC:	Decentralized Model Predictive Control
RHC:	Receding Horizon Control
EUV:	Electric Unmanned Vehicle
EV:	Electric Vehicle
MUVS:	Multi Unmanned Vehicle Systems
UV:	Unmanned Vehicle
BM:	Battery Manager
TM:	Traffic Manager
MILP:	Mixed Integer Linear Programming
QP:	Quadratic Programming
LP :	Linear Programming
HJ:	Hamilton-Jacobi
LTI:	Linear Time-Invariant
DoA:	Domain of Attraction
RCI:	Robust Control Invariant
UUB:	Uniformly Ultimately Bounded
LQ :	Linear Quadratic
SoC :	State-of-Charge

Chapter 1

Introduction

In the past decades, Unmanned Vehicles (UV) have attracted significant attention, and there has been increasingly rapid progress in their technology both in academia and industry [1, 2]. On the other hand, transportation electrification has been attaining growing attention because people have been encouraged to find a clean, safe, and efficient solution for their mobility due to global warming, limited fossil fuel resources, and the increasing oil price [3], [4].

Hence, research endeavours have undoubtedly been spurred by the improved congestion, emission, utilization and safety of road networks that UVs and Electric Unmanned Vehicles (EUVs) would provide rather than the traditional transportation systems. To obtain a fully automated transportation system, different aspects must be addressed [5]. To name a few: each vehicle must be able to track a reference signal and reach the desired goal [6–8]; motion planning and obstacle avoidance solutions are needed to navigate in potentially unknown obstacle scenarios [9]; formation control tools are needed to guarantee coordination among vehicles [10–13]; absence of collisions must be ensured [14–16]. Besides, when EUVs are considered, then battery management systems are required to prevent battery shortage along the path [17, 18].

1.1 Literature Review

If collision avoidance and battery management systems are of interest, then receding horizon Model Predictive Control (MPC) solutions are particularly appealing given their capability to handle disturbances and constraints explicitly [19–21]. In particular, MPC solutions exploit the dynamic model of a plant and the available measurements to predict the system's future evolution over a finite prediction horizon. Such predictions are then used to compute the control actions by solving a constrained optimization problem that takes care of future costs, disturbances, and constraints [22, 23]. Detailed studies about existing MPC solutions can be found in the survey papers [24–26] and references therein.

Robust MPC strategies are of particular interest for the control problems addressed in this thesis. Robust MPC solutions are used in control problems where the plant dynamics are subject to model uncertainties and/or bounded disturbances [27]. In this setup, robust MPC schemes search for optimal admissible state-feedback control actions that minimize a given cost function for the worst-case realization of uncertainties and disturbances [28]. The main drawback of such a paradigm is that the resulting optimization problem is often computationally demanding and not suitable for systems with fast dynamics. To mitigate such a drawback, the explicit MPC scheme has been developed in [29, 30]. Explicit MPC allows to pre-calculate the MPC optimization problems offline for a given range of operating conditions of interest. By exploiting multi-parametric programming techniques, explicit MPC computes the optimal control action offline as an "explicit" function of the state and reference vectors. Then, online control action computation diminishes to a simple function evaluation that can be obtained by means of a lookup table [31]. Despite the mentioned online computational advantages, explicit strategies are only applicable for systems with a small number of states and inputs, few constraints, and short time horizons [32]. Furthermore, the resulting control actions are typically very conservative.

In [33–35], a different MPC solution, namely Set-Theoretic Model Predictive Control (ST-MPC), has been proposed to provide a compromise between the offline (explicit MPC) and online MPC schemes. ST-MPC, by exploiting set-theoretic arguments, is capable of decreasing the computational demand of traditional robust MPC controllers [36]. This is accomplished by moving part of the required computations into an offline phase and leaving online a simple and real-time affordable convex optimization problem. Different from the explicit MPC, ST-MPC does not pre-compute the control actions so rendering the overall control paradigm less conservative.

In this thesis, the ST-MPC paradigm is exploited and extended to deal with two control problems: (i) battery shortage prevention for EUVs; (ii) collision avoidance strategy for constrained Multi Unmanned Vehicle Systems (MUVS).

1.1.1 Battery Shortage Prevention Strategies For EUVs

As long as Electric Vehicle (EV) battery management systems are concerned, different solutions have been proposed. In [37], a stochastic MPC energy management system is used to predict the driver's actions and style and anticipate future power requirements. In [38], by fusing different real-time sources of information (GPS, vehicle-to-vehicle interactions, real-time traffic information), an enhanced trip prediction model has been developed for the design of an optimal charge-depletion approach. Along similar lines are [39, 40]. In [39], the authors propose bi-directional communications between the EVs and smart buildings to obtain better information on the vehicle status and road conditions and, consequently, improve the EVs' energy usage. In [40], a predictive on/off controller is proposed to fully utilize the battery electrical using historical trip information from the historical vehicle data and GPS navigation system.

From the above state-of-the-art, it appears that accurate predictions of the vehicle and driver's future behaviours are required to obtain a global optimal battery management system. Moreover, the optimal battery profile is usually trip-dependent, indicating that the same energy profile might not be optimal for different paths. Besides, another typical issue of optimization-based battery management systems is their computational complexity, which might not be suitable for their real-time implementation [38].

1.1.2 Collision Avoidance Strategies for MUVS

Of interest are cooperative control solutions capable of addressing the collision-free reference tracking control problem for MUVs moving in a shared environment [41]. In the related literature, different solutions have been proposed. In [14], collision-free movements and deadlock-avoidance for multi-robot systems are obtained by solving a quadratic optimization problem exploiting a mixture of relaxed control barrier function and braking controllers. In [42–47], different Mixed Integer Linear Programming (MILP) based solutions have been investigated. In [42], the collision avoidance problem for a group of vehicles is formulated as an MILP optimization problem where each vehicle has an apriori known fixed number of possible trajectories. Along similar lines is [43], where the proposed solution guarantees collision avoidance for a team of three vehicles, and the exponential complexity of the obtained Hamilton-Jacobi (HJ) and MILP problems is lessened using a combinational method based on HJ's reachability and MILP programming concepts. In [45], an MPC strategy is used to guarantee the absence of collision for a robot formation. First, feedback linearization and MPC controllers are used to recast the problem in terms of a mixed-integer quadratic programming problem. Then, a branch and bound algorithm is used to reduce the complexity of the obtained optimization problem. In [46], the authors have implemented a robust Decentralized Model Predictive Controller (DMPC) for a team of cooperating UVs where possibilities of collisions are modelled as coupling constraints. The resulting optimization problem is solved as an integer programming problem. Finally, in [47], a theoretical framework based on an MPC paradigm is developed to methodologically ensure collision-free urban traffic by formally deriving constraints that guarantee the absence of collisions. The provided solution, although interesting, results in being conservative.

From the existing literature, the following can be highlighted. Collision-free reference tracking movements cannot be assured if the trajectory planner/formation control schemes consider only the vehicles' kinematic unconstrained models. On the other hand, to provide a complete control solution capable of ensuring vehicles' constraints fulfillment as well as the absence of collisions, often MPC schemes represent a natural choice even though the resulting control approaches might be either conservative or computationally demanding.

To mitigate the computational demand of both centralized and decentralized MPC solutions, in [48], a different MPC-oriented control architecture has been proposed. In particular, by resorting to set-theoretic ideas [33, 49] and a receding-horizon control strategy [34], a centralized Traffic Manager (TM) is proposed. Such an entity is in charge of predicting one-step collision possibilities and stop the vehicles whenever necessary. Such a solution's peculiar capability allows decoupling the reference tracking and the collision avoidance problems partially. As a result, the control action used by each vehicle is obtained by solving a Quadratic Programming (QP) problem, while collision avoidance is obtained using simple convex set-membership tests arising from the employed set-theoretic control paradigm.

1.2 Thesis Overview and Contributions

This thesis is organized as follows:

- **Chapter 2:** The main concepts and definitions used along the thesis are defined and presented. Moreover, the basic ST-MPC control paradigm is summarized and reviewed.
- **Chapter 3:** A control architecture is proposed to deal with the reference tracking control problem for EUV equipped with batteries of limited energy capacity. The proposed solution consists of a Battery Manager (BM) module that can avoid energy shortage by appropriately imposing time-varying upper bounds on the vehicle's maximum acceleration. In particular, some key properties of the ST-MPC paradigm are exploited to couple the reference tracking and the battery shortage problems.
  - The control architecture proposed in this chapter is published in the proceeding of the 2020 IEEE Conference on Control Technology and Applications [50].
- **Chapter 4:** Starting from the solution developed in [48], a novel control architecture is

proposed to deal with the reference tracking and collision avoidance control problems for UVs moving in shared environments. We consider a scenario where each vehicle follows a sequence of waypoints (trajectory) imposed by a local planner. In this contest, a designed centralized TM in conjunction with ad-hoc designed local MPC, assure the absence of collisions while minimizing the total vehicle's stops occurrences.

  - The results in this chapter have been in part submitted to the 2021 European Control Conference [51]. Moreover a journal paper is currently under review for the Transactions on Automatic Control (TAC) journal [52].

- **Chapter 5:** The obtained results are summarized and future research directions are outlined.

1.3 Publications

- [50] Savehshemshaki S, Lucia W. "A Receding Horizon Battery Shortage Prevention Control Strategy for Electric Unmanned Vehicles." IEEE Conference on Control Technology and Applications (CCTA), pp. 108-113, 2020.
- [51] Savehshemshaki S, Lucia W. "A Receding Horizon Collision Avoidance Strategy for Constrained Multi Unmanned Vehicle Systems."European Control Conference (ECC), 2020 *(under review)*
- [52] Bagherzadeh M, Savehshemshaki S, and Lucia W. "Guaranteed Collision-Free Reference Tracking in Constrained Multi Unmanned Vehicle Systems." In IEEE Transaction on Automatic Control, 2020 *(under review)*

Chapter 2

Background Materials

In this chapter, the background material required to understand the rest of this thesis is presented. In particular, first standard definitions and preliminaries are given. Then, the dual-mode ST-MPC control paradigm is reviewed, the computational algorithm is introduced, and its main properties are highlighted.

2.1 Preliminaries and Definitions

We consider systems whose dynamics are described by the following constrained discrete-time Linear Time-Invariant (LTI) state-space representation

$$x(t+1) \;=\; Ax(t) + Bu(t) + d(t) \tag{1}$$

subject to bounded state and input constraints

$$x(t) \in \mathcal{X},\ \forall t \geq 0 \tag{2}$$

$$u(t) \in \mathcal{U},\ \forall t \geq 0 \tag{3}$$

with $d(t)$ a bounded exogenous disturbance, i.e.

$$d(t) \in \mathcal{D} \subset \mathbb{R}^d, \quad 0_d \in \mathcal{D} \tag{4}$$

where $t \in \mathbb{Z}_+ := \{0, 1, ...\}$ denotes the sampling time instants, $u \in \mathbb{R}^m$ the input vector and $x(t) \in \mathbb{R}^{n_x}$ the state-space vector. Moreover, $\mathcal{U} \subset \mathbb{R}^m$ and $\mathcal{D} \subset \mathbb{R}^d$ are compact subsets with $0_m \in \mathcal{U}$ and $0_{n_x} \in \mathcal{X}$, respectively. Moreover, A and B are matrices of suitable dimensions characterizing the system dynamical behavior.

The following definitions are used along the rest of this thesis:

Definition 1. *Given two sets $\mathcal{R} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{Q} \subseteq \mathbb{R}^{n_x}$, the **Pontryagin/Minkowski set difference and sum**, $\mathcal{R} \sim \mathcal{Q}$ and $\mathcal{R} \oplus \mathcal{Q}$ respectively, are defined as follows [53]:*

$$\begin{aligned} \mathcal{R} \sim Q &:= \{x \in \mathbb{R}^{n_x} : \; x + q \in \mathcal{R}, \, \forall q \in \mathcal{Q}\} \\ \mathcal{R} \oplus \mathcal{Q} &:= \{r + q : \; r \in \mathcal{R}, \; q \in \mathcal{Q}\} \end{aligned} \tag{5}$$

□

Definition 2. *Let $\mathcal{Q} \subset \mathbb{R}^{n_x}$ be a neighborhood region of the origin. The closed-loop trajectory of (1)-(3), is said to be **Uniformly Ultimately Bounded (UUB)** in $\mathcal{Q}$ if for all $\mu > 0$, there exists a function $T(\mu) > 0$ such that $\forall \, x(0)$, $\|x(0)\| \leq \mu \rightarrow x(t) \in \mathcal{Q}$, $\forall d(t) \in \mathcal{D}$ and $\forall \, t \geq T(\mu)$, see [33].* □

Definition 3. *A set $\mathcal{T} \subseteq \mathcal{X}$ is said **Robust Control Invariant (RCI)** for (1)-(3) if*

$$\begin{gathered} \forall x, \, x \in \mathcal{T} \rightarrow \exists u \in \mathcal{U} : \\ Ax + Bu + d \in \mathcal{T}, \;\; \forall d \in \mathcal{D} \end{gathered} \tag{6}$$

□

Definition 4. *Given the plant model (1)-(3) and a set $\mathcal{T} \subset \mathbb{R}^n$, then the set of states **robustly***

***controllable to $\mathcal{T}$ in one-step** is defined as [33], [53, 53]:*

$$Pre(\mathcal{T}) := \{x \in \mathbb{R}^{n_x} : \exists u \in \mathcal{U}\, s.t.\, Ax + Bu + d \in \mathcal{T},\, \forall d \in \mathcal{D}\} \tag{7}$$

$Pre(\mathcal{T})$ is the set of states which evolves into the target set $\mathcal{T}$ in the one time step regardless of admissible disturbances. □

Definition 5. *For a Given target set $\mathcal{T} \subseteq \mathcal{X}$ the **N-step controllable sets** ($\mathcal{T}^N$) of the system (1) subject to the constraints (2)-(3) is defined recursively as:*

$$\mathcal{T}^n = Pre(\mathcal{T}^{n-1}) \cap \mathcal{X}, \quad \mathcal{T} \equiv \mathcal{T}^0, \quad n = \{1, 2, ..., N\} \tag{8}$$

□

2.2 Set-Theoretic Model Predictive Control (ST-MPC)

The dual-mode ST-MPC control paradigm was originally developed in [33, 34, 49] to solves a robust constrained regulation problem to the equilibrium pair $(x^{eq},\, u^{eq})$ by means of a computationally low-demanding MPC solution where most of the required computations are moved into an offline phase, leaving online a simple and affordable QP problem. The offline and online steps are here summarized:

Offline operations:

(Off-1) By considering the unconstrained disturbance-free model of (1), design a stabilizing state-feedback control law,

$$u(t) := f^0(x(t), x^{eq}, u^{eq}) \tag{9}$$

to asymptotically bring the plant state trajectory to the equilibrium pair $(x^{eq},\, u^{eq})$. Such a controller is hereafter referred to as the terminal controller.

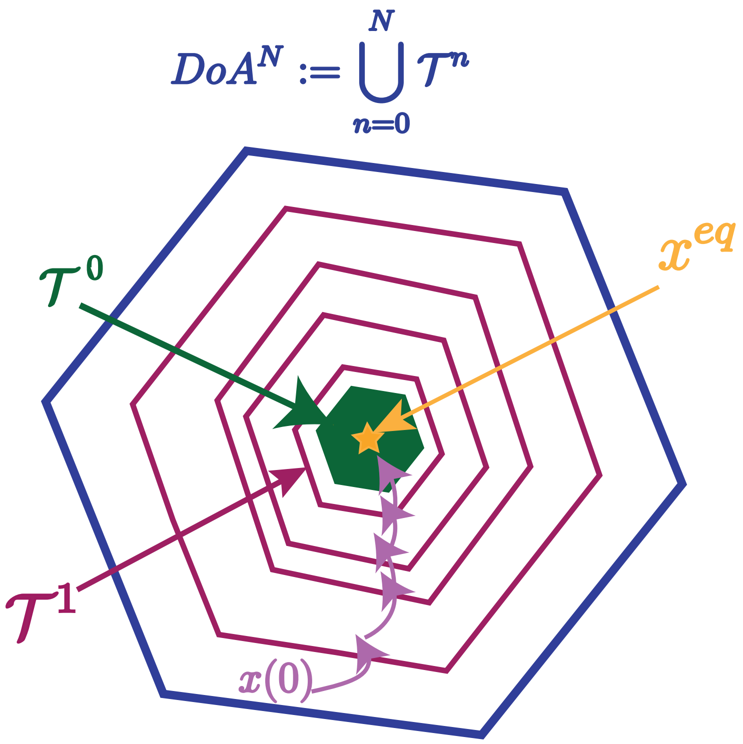

Figure 1: Family of N robust one-step controllable sets and the controller's domain (DoA)

(Off-2) The smallest RCI region, namely $\mathcal{T}^0$, associated to the terminal controller is computed as proposed in [54] under the requirements $\mathcal{T}^0 \subseteq \mathcal{X},\ u(t) \in \mathcal{U}, \forall t$. The region $\mathcal{T}^0$ is hereafter referred to either as the terminal region or as the domain of attraction of the terminal controller, namely DoA^0.

(Off-3) The controller computed in Steps 1-2 might have a very small domain. To ensure that any initial state $x(t)$ belongs to the controller admissible region, the DoA^0 must be enlarged. The latter is here achieved by computing a family of N robust one-step controllable sets, namely $\{\mathcal{T}^n\}_{n=0}^N,\ N \geq 1$, by applying the following recursive definition [33]:

$$\begin{aligned} \mathcal{T}^n &:= \{x \in \mathcal{X} : \exists u \in \mathcal{U} : \forall d \in \mathcal{D}, Ax + Bu + d \in \mathcal{T}^{n-1}\} \\ &= \{x \in \mathcal{X} : \exists u \in \mathcal{U} : Ax + Bu \in \tilde{\mathcal{T}}^{n-1}\} \end{aligned} \tag{10}$$

where $\tilde{\mathcal{T}}^{n-1} := \mathcal{T}^{n-1} \sim \mathcal{D}$ and N is the number of computed sets. The set union $\bigcup_{n=0}^N \mathcal{T}^n$

defines the controller's domain, namely DoA^N (see Fig. 1)

$$DoA^N := \bigcup_{n=0}^{N} \mathcal{T}^n \tag{11}$$

In what follows, a remark is given to clarify, from a practical point of view, how the steps *(Off-1)* and *(Off-3)* can be computed.

Remark 1. *In step (Off-1), the only requirement for the state-feedback controller $u^0(t) := (f^0(x(t), x^{eq}, u^{eq})$ is to ensure asymptotic stability for the disturbance and constraint-free UV's dynamical model (1). As a consequence, any existing state-feedback controller can be employed. In the computation algorithm shown in Section 2.2.1, we have used a linear state-feedback controller where the controller gain K^0 is the optimal gain given by the Linear Quadratic (LQ) controller:*

$$u(t) = K^0(x_i(t) - x^{eq}) + u^{eq} \tag{12}$$

In step (Off-3), the robust one-step controllable sets (10) can be built using the capability offered by the MPT Matlab toolbox [55]. Nevertheless, in the literature, different algorithms exist to compute exact or approximated robust one-step controllable sets, see e.g. [33, 53, 56], and references therein.

Online operations (assuming $x(0) \in DoA(x^{eq})$):

(On-1) Let $x(t)$ be the current vehicle's state-space vector, find the smallest set index $n(t)$ containing $x(t)$, i.e.

$$n(t) := \min_n \{n : x(t) \in \mathcal{T}^n\} \tag{13}$$

(On-2) If $n(t) = 0$ (i.e. $x(t) \in \mathcal{T}^0$) apply the control action given by terminal controller (9), otherwise apply the control action given by the solution of the following QP optimization problem:

$$\begin{aligned} u(t) = \arg\min_{u \in \mathcal{U}} J(x(t),\, x^{eq},\, u) \ s.t. \\ Ax(t) + Bu \in \tilde{\mathcal{T}}^{n(t)-1} \end{aligned} \tag{14}$$

where $J(x(t),\, x^{eq},\, u)$ is any convex cost function of interest.

2.2.1 ST-MPC Basic Computational Algorithm

The above offline and online developments are summarized in the following computational algorithm.

```
Set-Theoretic MPC (ST-MPC)

————Offline Operations————

Output: $\bigcup_{n=0}^{N} \mathcal{T}^n$
1: Compute the terminal region $\mathcal{T}^0$ and the associated controller (9)
2: Compute a family of $N$ robust one-step controllable sets according to the recursion (10) and
   store the resulted family $\bigcup_{n=0}^{N} \mathcal{T}^n$.

————Online Operations————

Input: $\{\mathcal{T}^n\}_{n=0}^{N},\, x(0) \in \bigcup_{n=0}^{N} \mathcal{T}^n$
Output: $u(t)$
1: Find the smallest set index $n(t)$ containing $x(t)$, i.e.
       $n(t) := \min_{n} n : x(t) \in \mathcal{T}^n(x^{eq})$
2: if $n(t) == 0$ then $u(t) = K^0(x(t) - x^{eq}) + u^{eq}$ (see(12))
3: else Find $u(t)$ by solving (14)
4: end if
5: $t \leftarrow t + 1$ and goto Step 1
```

2.2.2 ST-MPC Properties

It is possible to prove that the described set-theoretic control paradigm enjoys the following properties [33, 34, 49]:

- For all $x(0) \in \bigcup_{n=0}^{N} \mathcal{T}^n$, the plant's state vector evolution converges to the terminal region $\mathcal{T}^0$ in at most N steps. Moreover, prescribed state and input constraint are always fulfilled.
- The trajectory is UUB (see Definition. 2) in $\mathcal{T}^0$ regardless of any disturbance realization $d(t) \in \mathcal{D}$
- In the absence of disturbance, the stability is asymptotic
- The convex cost function $J(x(t),\, x^{eq},\, u)$ in (14) can be arbitrarily chosen and changed at each sampling time to minimize any performance index of interest, e.g.

  1. time to converge to the equilibrium point:

  $$J(x(t),\, x^{eq},\, u) = ||Ax(t) + Bu - x_{eq}||_2^2$$

  2. control effort:

  $$J(x(t),\, x^{eq},\, u) = ||u||_2^2$$

Chapter 3

A Receding Horizon Battery Shortage Prevention Control Strategy for Electric Unmanned Vehicles

This chapter deals with the reference tracking control problem for EUVs equipped with batteries of limited energy capacity. We design a novel control architecture, equipped with a battery manager module capable of avoiding energy shortage by appropriately imposing time-varying upper bounds on the EUV's maximum acceleration. In particular, we exploit some key properties of the ST-MPC paradigm to couple the reference tracking and the battery shortage problems.

The control architecture proposed in this chapter is published in the proceeding of the 2020 IEEE Conference on Control Technology and Applications [50].

3.1 Problem Formulation

We consider an EUV, described by (1)-(3) where, $u \in \mathbb{R}^2$ is the acceleration input vector and $x = [p^T, v^T]^T \in \mathbb{R}^4$ is the state-space vector ($p = [p_x,\, p_y]^T \in \mathbb{R}^2$ denotes the EUV's position vector and $v = [v_x,\, v_y]^T \in \mathbb{R}^2$ denotes the EUV's velocity vector). We consider the described

EUV is powered by a limited-energy battery and whose objective is to reach a target destination, following a prescribed path, and without incurring into a battery shortage.

Assumption 1. *The vehicle's model (1) is reachable and contains real (e.g., double integrator models) or artificially added (e.g., pre-compensated dynamics) dynamics. Therefore, the pair* $(x_{p_i}^{eq} := [p_i^T,\, 0_{n_i-2}]^T, u_i^{eq} = 0_{m_i})$ *defines, for any* $p_i \in \mathbb{R}^2$, *an equilibrium point for* UV_i.

Assumption 2. *Given the vehicle's initial position* $p(0)$ *and desired destination* p_d, *we assume that a planner module (at a kinematic level) provides a sequence of* N_p *waypoints, namely* $\mathbf{p} \in \mathbb{R}^{2\times N_r}$ *from* $p(0)$ *to* p_d *(see Fig. 2), i.e.*

$$\mathbf{p} = \{p_0, \ldots, p_{N_p}\},\ \textit{with}\ \ p_0 \equiv p(0),\ p_{N_p} \equiv p_d \tag{15}$$

Moreover, we assume that the planner is configured such than the maximum distance between two consecutive waypoints is bounded by $\delta > 0$, *i.e.*

$$||p_{k+1} - p_k||_2 \leq \delta,\ \ 0 \leq k < N_p \tag{16}$$

Assumption 3. *We assume that battery discharge rate is proportional to the vehicle's acceleration magnitude, i.e. the battery's State-of-Charge (SoC) evolution is described by the following discrete-time model [39, 57]:*

$$SoC(t+1) = SoC(t) - \alpha_{bat}||u(t)||_2 \tag{17}$$

where $\alpha_{bat} > 0$ *is a scalar coefficient whose value is a function of the specific battery model. Moreover, we assume that the BM module can override the input constraint (3) by imposing, whenever necessary, an arbitrary but admissible maximum acceleration profile* $\mathcal{U}_{BM}(t)$, *i.e.*

$$u(t) \in \mathcal{U}_{BM}(t) \subseteq \bar{\mathcal{U}},\ \bar{\mathcal{U}} \equiv \mathcal{U},\ \forall\, t \tag{18}$$

where $\mathcal{U}$ *is input constraint defined in (3).*

Remark 2. *Please notice that the assumed battery discharge model (17) (see Assumption 3) is an approximation of the actual battery discharge function, which generally depends on several other parameters, see e.g. [58,59]. Nevertheless, the model (17) accurately describes the battery discharge rate as a function of its predominant factor, i.e. the vehicle's acceleration magnitude. Please also notice that the condition (16) (Assumption 2), is only instrumental to guarantee recursive feasibility of the tracking controller proposed in [48].* $\square$

Please note that in this chapter, we use the discrete index t to denote the discrete-time evolution of variables (e.g. $x(t)$, $u(t)$), while we use the discrete-time subscript $k \in \mathbb{Z}_+ := \{0, 1, ...\}$ for variables evolving independently from t (e.g. p_k).

The control problem addressed in this paper can be formally stated as follows:

EUV Reference Tracking with Guaranteed Battery Shortage Prevention: Given the vehicle's constrained model (1)-(3), the battery's SoC discharge model (17), a sequence of waypoints (15), and a battery manager module imposing the acceleration constraint (18), we design a state-feedback controller

$$u(t) := f(x(t), p_k, \mathcal{U}_{BM}(t)) \tag{19}$$

and a BM module such that the following objectives are fulfilled:

- (**O1**) The controller (19) is able to sequentially track the waypoints $p_k, \forall k$, with a tracking error that, for each waypoint, is UUB in a finite-number of steps and irrespective of the time-varying battery-manager acceleration constraints (18) and disturbances (4).
- (**O2**) At $t = 0$, and irrespective of any possible future disturbance (4) realization, the BM module must be able to find (if it exists) a conservative acceleration profile $\mathcal{U}_{BM}(t)$, $\forall t \in [0, t_T]$ with $t_T > 0$ the time at which the target is reached, such that $SoC(t) \geq 0,\ \forall t \in [0, t_T]$. Moreover, in a receding-horizon fashion, $\forall t > 0$, and in accordance with a desired cost function, the BM must be able to online improve (whenever possible) the acceleration profile (18) computed at $t - 1$.

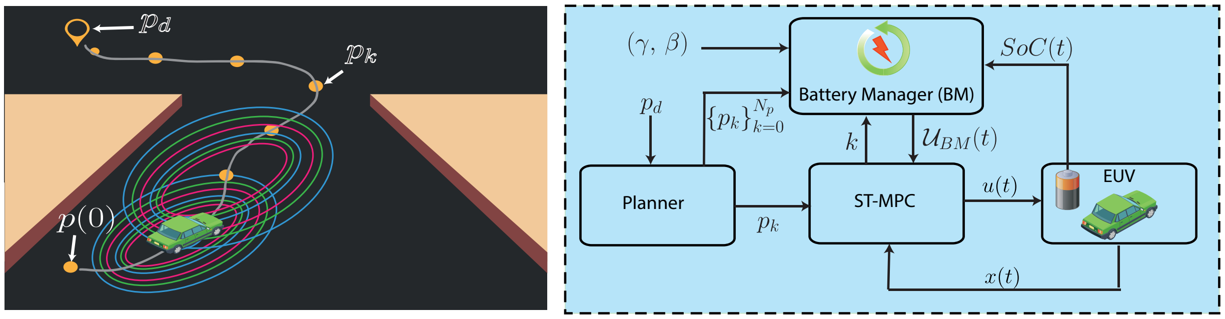

Figure 2: Proposed control architecture equipped with a ST-MPC tracking controller and a Battery Manager for energy shortage prevention.

3.2 Proposed Solution

In this section, a solution to the control problems (**O1**)-(**O2**) is presented, see Fig. 2. The developments proceed as follows: First, the proposed ST-MPC tracking controller [48] is adapted to solve (**O1**) by assuming that the available energy is unlimited, and the BM is not needed. Then, a battery manager module providing a solution to (**O2**) is designed, and it is shown how it enables the ST-MPC to solve (**O1**) in the case of limited available energy. Finally, all the developments are collected into a complete computational algorithm summarizing the joint actions of ST-MPC and BM.

3.2.1 Set-Theoretic MPC Tracking Controller

In Section. 2.2, the ST-MPC controller was presented to solve a regulation problem. Here, such a control paradigm is extended to solve a waypoint tracking problem. The proposed solution simply exploits the linearity of the model to change the desired equilibrium pair according to the prescribed waypoint. In particular, under the *Assumption* 2, and following a similar reasoning as in [48], it can be proved that (**O1**) has a solution (assuming for now unlimited battery energy) if the following set containment condition is imposed (see Fig. 3):

$$DoA^{\bar{N}}(x^{eq}_{p_{k+1}}) := \bigcup_{n=0}^{N_i} \mathcal{T}^n(p_{k+1}) \supseteq \mathcal{B}_\delta(p_{k+1}) \oplus \mathcal{T}^0(0_2) := \mathcal{W} \qquad (20)$$

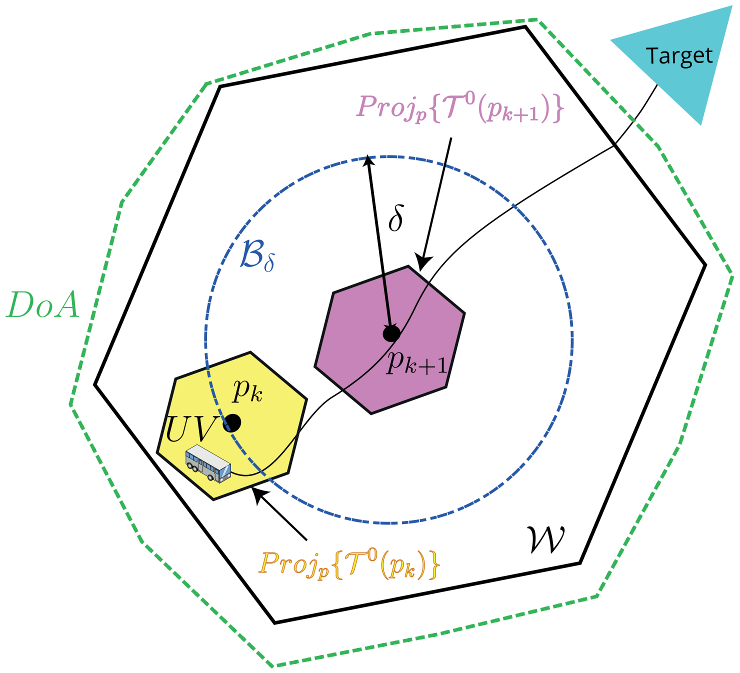

Figure 3: Graphically describes feasibility condition for waypoint switches (20).

where $x^{eq}_{p_{k+1}}$ and $x^{eq}_{p_k}$ denote the equilibrium state associated to p_{k+1} and p_k, respectively. In simply words, condition (20) assures recursive feasibility to the controller every time we switch waypoint. In particular, (20) implies that if the waypoint switch $p_k \rightarrow p_{k+1}$ occurs when $x(t) \in \mathcal{T}^0(x^{eq}_{p_k})$, then it is ensured that $x(t)$ is contained in the DoA of the regulation controller re-centered in p_{k+1}, i.e. $x(t) \in DoA^{\bar{N}}(x^{eq}_{p_{k+1}})$ (Figs. 3 and 4.a), see [48] for further details. According to the given reasonings, if the condition (20) is fulfilled, the waypoint tracking controller can be summarized as follows:

ST-MPC - Waypoints Tracking Controller

Online: (assuming $x(0) \in DoA(x^{eq}_{p_0})$)

```
1: Compute $n(t) := \min_n n : x(t) \in \mathcal{T}^n(x^{eq}_{p_k})$
2: **if** $n(t) == 0$ **then** $k \leftarrow k+1$, goto Step 1
3: **end if**
4: $x^{eq} \leftarrow x^{eq}_{p_k}$
```

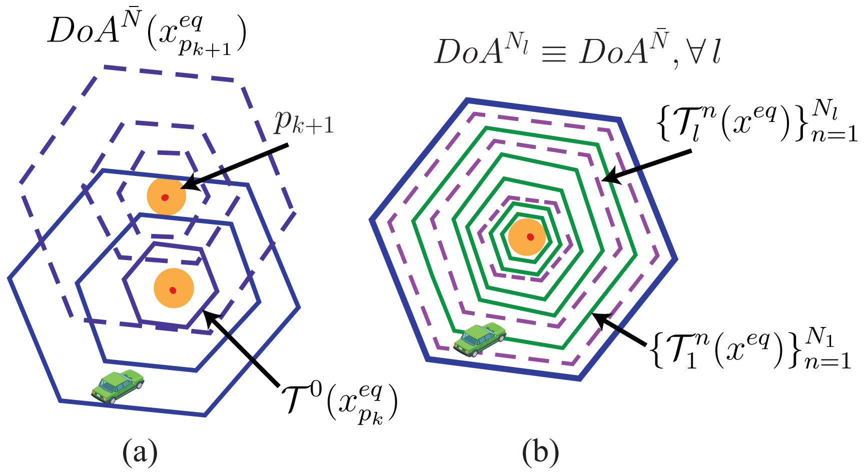
$DoA^{\bar{N}}(x^{eq}_{p_{k+1}})$
p_{k+1}
$\mathcal{T}^0(x^{eq}_{p_k})$
(a)
$DoA^{N_l} \equiv DoA^{\bar{N}}, \forall\, l$
$\{\mathcal{T}_l^n(x^{eq})\}_{n=1}^{N_l}$
$\{\mathcal{T}_1^n(x^{eq})\}_{n=1}^{N_1}$
(b)

Figure 4: Subplot (a) describes the implication of the recursive feasibility condition in (20); Subplot (b) describes the condition (25).

```
5: if $n(t) == 0$ then $u(t) = K^0(x(t) - x^{eq})$
6: else Find $\bar{u}(t)$ by solving (14) where the convex cost function of interest is $J(x(t),\, x^{eq},\, u) = ||Ax(t) + Bu - x^{eq}||_2^2$
7: end if
8: $t \leftarrow t + 1$ and goto Step 1
```

Remark 3. *It is worth to underline that the smallest terminal region (smallest RCI region, i.e. $\mathcal{T}^0$) is of interest since our objective is to steer the state trajectory as close as possible to each waypoint p_k. Indeed, the RCI region $\mathcal{T}^0$ determines the maximum waypoint tracking error we commit for any disturbance realization. Additionally, in the absence of disturbances, the controller asymptotically stabilizes $x(t)$ to the equilibrium point $x^{eq}_{p_k}$ thus reaching perfect set-point tracking. [48]*

3.2.2 Battery Manager (BM) and Energy Shortage Prevention

In this subsection, the BM logic and the battery shortage prevention strategy are presented to provide a solution to (**O2**).

First, to design the energy shortage prevention strategy, a prediction model for energy usage is needed. To this end, it is essential to remark some key properties of the used ST-MPC tracking controller:

- *(P1)*: Consider the single waypoint p_k regulation problem and a family of $\bar{N}$ robust one step controllable sets $\{\mathcal{T}^n(x_{p_k}^{eq})\}_{n=1}^{\bar{N}}$ obtained under (3)-(4). By construction, it is ensured that in at most $\bar{N}$ step, the state trajectory will be robustly confined within $\mathcal{T}^0(x_{p_k}^{eq})$, see [34] for a formal proof.
- *(P2)*: Consider the tracking problem from $p(0)$ to p_{N_p} with N_p intermediate waypoint p_k (see *Assumption* 2), then the ST-MPC tracking controller assures that in at most

$$N_{tot}(\bar{\mathcal{U}}) = N_p\bar{N} \tag{21}$$

steps, the state trajectory will reach the target destination p_{N_p}, i.e. $x(t) \in \mathcal{T}^0(x_{p_{N_p}}^{eq})$.

- *(P3)*: According to input constraint (3) and to the SoC discharge model (17), each waypoint p_k tracking requires, in the worst-case scenario, a drop in the SoC equals to

$$E_{p_k}(\bar{\mathcal{U}}) = \alpha_{bat}\bar{N}||\bar{\mathcal{U}}||, \quad \text{with} \quad ||\bar{\mathcal{U}}|| := \max_{u\in\bar{\mathcal{U}}} ||u||_2 \tag{22}$$

As a consequence, the worst-case SoC drop required to reach the goal p_{N_p}, is:

$$E_{tot}(\bar{\mathcal{U}}) = N_p\alpha_{bat}\bar{N}||\bar{\mathcal{U}}|| = N_pE_{p_k}(\bar{\mathcal{U}}) \tag{23}$$

From (23) it is possible to appreciate that the energy demand is a function of the number of waypoints (N_p), number of one-step controllable sets (N) and maximum acceleration norm ($||\bar{\mathcal{U}}||$). As a consequence, assuming fixed the number of waypoints, the BM can reduce the energy consumption by properly imposing an acceleration profile $u(t) \in \mathcal{U}_{BM}(t) \subset \bar{\mathcal{U}}$ (see *Assumption* 3). On the other hand, arbitrary changes of the input constraints overtime might invalidate the ST-MPC tracking controller's recursive feasibility.

A battery manager preserving the recursive the tracking controller's recursive feasibility can be designed as described in the following proposition.

Proposition 1. *Consider a finite-set of $L > 1$ admissible acceleration constraints profiles*

$$U_{set} = \{\mathcal{U}_1, \mathcal{U}_1 \ldots, \bar{U}_L\}, \ \ \textit{with} \ \ \mathcal{U}_1 \subset \mathcal{U}_2 \ldots \subset \mathcal{U}_L \equiv \bar{\mathcal{U}} \tag{24}$$

Build for each $\mathcal{U}_l$, $l = 1, \ldots, L$ a RCI region $\mathcal{T}_l^0(x_{eq})$ and the associated stabilizing controller, namely $u(t) = K_l^0 x(k)$ and a family of N_l robust one-step controllable sets, namely $\{\mathcal{T}_l^n(x_{eq})\}_{n=1}^{N_l}$ such that

$$DoA^{N_l} \equiv DoA^{\bar{N}}, \ \forall\, l \tag{25}$$

Then, the battery manager can arbitrarily impose, at each sampling time t, an arbitrary acceleration constraint

$$u(t) \in \mathcal{U}_{BM}(t), \ \ \textit{with} \ \ \mathcal{U}_{BM}(t) \in U_{set} \, \forall\, t \tag{26}$$

without affecting the ST-MPC tracking controller recursive feasibility.

Proof. By construction, for each admissible input constraint set in (24) a ST-MPC controller with $DoA^{N_l} \equiv DoA^{\bar{N}}$ is built. As a consequence, at each sampling time, the current state $x(t)$ belongs to all the family of one-step controllable sets $\{\mathcal{T}_l^n(x^{eq})\}_{n=1}^{N_l}$ (see Fig. 4.b) and therefore, recursive feasibility becomes irrespective of the used input constraint set (and associate ST-MPC controller) imposed by BM. □

Remark 4. *For each input constraint set $\mathcal{U}_l$, the number N_l of robust one-step controllable set needed to satisfy condition (26) changes. As a consequence, for each constraint set we have a* *different energy consumption profile from (22)-(23), i.e.*

$$E_{p_k}(\mathcal{U}_l) = \alpha_{bat} N_l ||\mathcal{U}_l|| \tag{27}$$

$$E_{tot}(\mathcal{U}_l) = N_p \alpha_{bat} N_l ||\mathcal{U}_l|| = N_p E_{p_k}(\mathcal{U}_l) \tag{28}$$

□

Given the results stated in *Proposition* 1 and *Remark* 4 , the BM task consists in finding an admissible input constraint profile (26) such that the vehicle is able to reach the desired destination p_{N_p} and a desired cost function (e.g. minimum energy, minimum-time) is minimized.

3.2.3 Offline BM operations

According to (**O2**), the BM must be able to offline define a possible conservative acceleration profile $\mathcal{U}_{BM}(t),\ 0 \leq t \leq t_T$ such that the EUV is able to reach the target with the initially available $SoC(0)$.

By parametrization the time-evolution of $\mathcal{U}_{BM}(t)$ (see Fig. 5) as:

$$\mathcal{U}_{BM}(t) = \begin{cases} \mathcal{U}_L \ \text{ if } \ 0 \leq t < \eta_L N_L \\ \mathcal{U}_{L-1} \text{ if } \ \eta_L N_L \leq t < \eta_L N_L + \eta_{L-1} N_{L-1} \\ \ \vdots \\ \mathcal{U}_1 \ \text{ if } \displaystyle\sum_{l=0}^{L-2} \eta_{L-l} N_{L-l} \leq t < \sum_{l=0}^{L-1} \eta_{L-l} N_{L-l} \end{cases} \\ \sum_{l=1}^{L} \eta_l \leq N_p, \ \ \eta_l \geq 0, \ \ \eta_l \in \mathbb{Z}_+ \tag{29}$$

with η_l denoting the number of waypoints in which the input constraint set level l is active, a conservative acceleration profile $\mathcal{U}_{BM}(t),\ 0 \leq t \leq t_T$ is given by the solution of the following integer linear programming problem

$$
\begin{aligned}
[\eta_1^*, \dots, \eta_l^*] = \arg \min_{\eta_1, \dots, n_l} J(\eta_1, \dots, \eta_l) \quad s.t. \\
\alpha_{bat} \sum_{l=1}^{L} (\eta_l N_l ||\mathcal{U}_l||) \leq SoC(0) \\
\sum_{l=1}^{L} n_l \leq N_p, \;\; \eta_l \geq 0, \;\; \eta_l \in \mathbb{Z}_+
\end{aligned}
\tag{30}
$$

with η_l^*, $l = 1, \dots, L$ denoting the optimal solutions and $J(\eta_1, \dots, \eta_l)$ any linear integer cost function. A possible multi-objective choice for $J(\eta_1, \dots, \eta_l)$ is the following

$$
J_{\gamma,\beta}(\eta_1, \dots, \eta_l) = \gamma \left(\sum_{l=1}^{L} \eta_l N_l \right) + \beta \left(\sum_{l=1}^{L} \eta_l E_{p_k}(\mathcal{U}_l) \right)
\tag{31}
$$

where depending on the weighting parameters $\gamma, \beta \geq 0$ three different cost functions can be defined (see Fig. 5):

- *Minimum Time ($\gamma = 1$, $\beta = 0$).* In this case, the cost function aims at minimizing the number of discrete time-instant needed to reach the target, i.e.

$$
J_{\gamma,\beta} = J_{time}(\eta_1, \dots, \eta_l) = \sum_{l=1}^{L} \eta_l N_l
$$

- *Minimum Energy ($\gamma = 0$, $\beta = 1$).* In this case, the cost function aims at minimizing the total amount of energy needed to reach the target, i.e.

$$
J_{\gamma,\beta} = J_{energy}(\eta_1, \dots, \eta_l) = \sum_{l=1}^{L} \eta_l E_{p_k}(\mathcal{U}_l)
$$

- *Multi-Objective ($\gamma > 0$, $\beta > 0$):* In this case, the cost function aims at finding the best compromise between the minimum time and the minimum energy solutions according to the weighting factors γ, β.

Fig. 5 provides an illustration of the input constraint profile $\mathcal{U}_{BM}(t)$ obtained for each possible choice of the cost function.

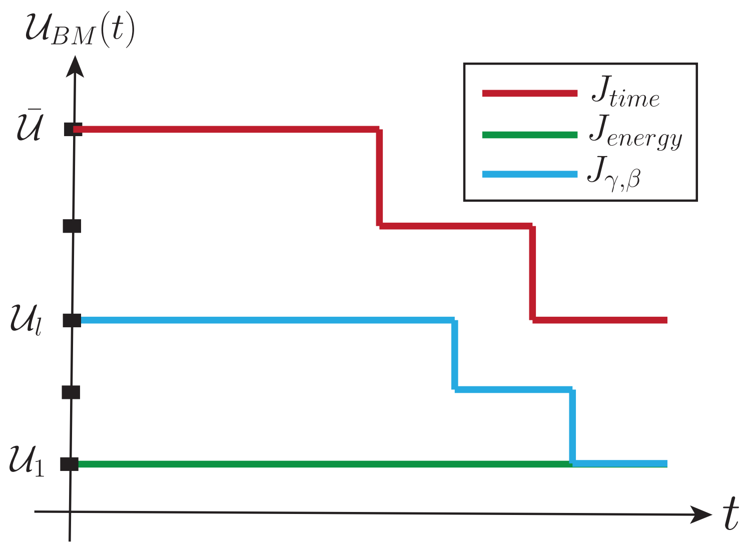

Figure 5: Examples of acceleration constraints profile $\mathcal{U}_{BM}(t)$ imposed by the BM according to different cost functions.

Remark 5. *Please notice that irrespective of the used cost function, if the optimization (30) admits a solution, then it is guaranteed that the EUV can reach the target destination with the available state-of-charge irrespective of any admissible disturbance realization satisfying (4).*

3.2.4 Online BM operations

Proposition 2. *The offline conservative acceleration profile $\mathcal{U}_{BM}(t)$, found by the BM at $t = 0$ solving the optimization problem (30) can be online improved (keeping the same cost function) or changed (changing the cost function), without affecting the ST-MPC tracking controller feasibility, by re-solving the optimization problem (30) with $SoC(t)$ instead of $SoC(0)$ and $N_p = N_p - k$, with k the index of the current active waypoint p_k.*

Sketch of proof: The following arguments can be given to prove the validity of the proposition:

- (*Changing the cost function*) In *Proposition* 1 it has been proved that the tracking controller preserve recursive feasibility regardless of the any admissible change in the acceleration constraint, see (26). As a consequence at any t, any admissible cost function J can be imposed, i.e. $J \in \{J_{\alpha,\beta},\ J_{time},\ J_{energy}\}$.
- (*Improving the same cost function*) At $t = 0$, the optimization (30) considers, for the entire prediction horizon and for each constraint level l, the worst-case realization of the command signal (in terms of energy demand). Therefore, at t, the predicted remaining charge, namely $\hat{SoC}(t)$ is

  $$\hat{SoC}(t) = SoC(0) - \alpha_{bat} \sum_{l=1}^{L} (\eta_l(t-1) N_l ||\mathcal{U}_l||) \tag{32}$$

  with $\eta_l(t-1)$ the number of waypoint in which the level l has been activated until $t-1$. On the other hand, the real $SoC(t)$ is given by the actual used command inputs $u(t)$, i.e.

  $$SoC(t) = SoC(0) - \alpha_{bat} \sum_{m=0}^{t-1} ||u(m)|| \tag{33}$$

  Therefore, since $\hat{SoC}(t)$ is a conservative estimation of $SoC(t)$, i.e. $SoC(t) \geq \hat{SoC}(t)$, if the optimization (30) is re-solved with $SoC(t)$ instead of $SoC(0)$, the conservatives of the offline computed acceleration profile can be improved. $\square$

As a conclusion, the proposed BM satisfied all the requirements in (**O2**).

3.2.5 Computational Algorithms

This section summarizes all the above developments in two final computational algorithms. The first describes the BM operations while the second the ST-MPC tracking controller.

Battery Manager ***(BM)*** $\forall\, t$

Input:

- Parameters: $\gamma, \beta, U_{set}, N_l,\ l = 1, \dots, L$
- Index k of the current waypoint p_k
- update-wanted={true, false}

Output: $\mathcal{U}_{BM}(t),\ \forall \bar{t} \geq t$

```
1: Compute $SoC(t)$ as in (33)
2: **if** update-wanted==true **then**
3:     Compute/Update $\mathcal{U}_{BM}(t)$ by solving
```

$$\begin{gathered}
[\eta_1^*, \dots, \eta_l^*] = \arg\min_{\eta_1,\dots,\eta_l} J(\eta_1, \dots, \eta_l) \quad s.t. \\
\alpha_{bat} \sum_{l=1}^{L} (\eta_l N_l ||\mathcal{U}_l||) \leq SoC(t) \\
\sum_{l=1}^{L} \eta_l \leq N_p - k,\ \ \eta_l \geq 0,\ \ \eta_l \in \mathbb{Z}_+
\end{gathered} \tag{34}$$

```
4: **end if**
5: Send $\mathcal{U}_{BM}(t)$ to ST-MPC
```

ST-MPC for Energy Shortage Avoidance ***ST-MPC-ESA***

—–*Offline*—–

Input: U_{set}

Output: $(\mathcal{T}_l^0(x^{eq}), K_l^0,)$ and $\{\mathcal{T}_l^n(x^{eq})\}_{n=1}^{N_l},\ \ \forall\, l$

```
1: For each $\mathcal{U}_l \in U_{set}$ build a RCI region $\mathcal{T}_l^0(x^{eq})$ and associated stabilizing controller, namely
   $u(t) = k_l^0 x(k)$.
2: For each $\mathcal{U}_l \in U_{set}$ build a family of $N_l$ robust one-step controllable sets, namely $\{\mathcal{T}_l^n(x^{eq})\}_{n=1}^{N_l}$
   such that the condition (25) is satisfied $\forall\, l$
```

——*Online* $\forall\, t$——

Input:

- The $l-$th active family of one-step controllable sets dictated by the BM, i.e. $\{\mathcal{T}_k^n\}_{n=0}^{N_l}$ and associated input constraint set $\mathcal{U}_l$

Output: $u(t)$

```
1: Compute $n(t) := \min_n n : x(t) \in \mathcal{T}_l^n(x_{p_k}^{eq})$
2: if $n(t) == 0$ then $k \leftarrow k+1$, goto Step 1
3: end if
4: $x^{eq} \leftarrow x_{p_k}^{eq}$
5: if $n(t) == 0$ then $u(t) = K_l^0(x(t) - x^{eq})$
6: else Find $\bar{u}(t)$ by solving
```

$$\begin{gathered} u(t) = \arg\min_u ||Ax(t) + Bu - x^{eq}||_2^2 \quad s.t. \\ Ax(t) + Bu \in \tilde{\mathcal{T}}_l^{n(t)-1}(x^{eq}),\ u \in \mathcal{U}_l \end{gathered} \tag{35}$$

```
7: end if
```

3.3 Simulation

In this section, the proposed control architecture's effectiveness is validated by means of a simulation example involving an EUV whose task is to reach a target destination in the minimum possible time with the available state-of-charge. The whole control architecture (see Fig. 2) has been emulated within the MATLAB environment and the MPT3 toolbox [55] has been used to implement the ***ST-MPC-ESA*** and ***BM*** algorithms.

We consider an EUV, whose dynamic is described by means of a double integrator model [60] and whose state space vector $x = [p^T, v^T] \in \mathbb{R}^4$ includes the 2D robot's positions $p = [p_x, p_y]^T$ and velocities $v = [v_x, v_y]^T$. The input vector $u \in \mathbb{R}^2$ is given by the the acceleration vector $u = [a_x, a_y]^T$. By using a sampling time $T_s = 0.1\,\mathrm{sec}$, the vehicle's discrete-time LTI system matrices are

$$A = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \, B = \begin{bmatrix} 0.005 & 0 \\ 0 & 0.005 \\ 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

Moreover, the EUV dynamics are subject to exogenous bounded disturbances

$$d(t) \in \mathcal{D} = \{d \in \mathbb{R}^4 : |d_j| \leq 0.07, \, j = 1, \ldots, 4\}$$

and the following state and acceleration constraints are prescribed

$$x(t) \in \mathcal{X} = \{x = [p_x, p_y, v_x, v_y]^T \in \mathbb{R}^4 : |v_x|, |v_y| \leq 100\}$$
$$u(t) \in \bar{\mathcal{U}} = \{u \in \mathbb{R}^2 : |a_x|, |a_y| \leq 25\}$$

The ***BM*** algorithm is implemented considering $L = 4$ possible levels of acceleration constraints (26)

$$U_{set} = \{\mathcal{U}_1, \ldots, \mathcal{U}_4\}$$

with

$$\mathcal{U}_1 = \{u \in \mathbb{R}^2 : |a_x|, |a_y| \leq 7\},$$
$$\mathcal{U}_2 = \{u \in \mathbb{R}^2 : |a_x|, |a_y| \leq 10\}$$
$$\mathcal{U}_3 = \{u \in \mathbb{R}^2 : |a_x|, |a_y| \leq 15\}$$
$$\mathcal{U}_4 \equiv \bar{\mathcal{U}}$$

In the performed simulation, the EUV's initial position $p(0)$ and desired target location p_d are

$$p(0) = [2,\, -25]^T,\ p_d = [9.12,\, 355]^T$$

and $N_p = 713$ waypoints p_k are generated by the path planner as follows

$$p_k = [10\sin(0.09k),\, 5k]^T,\ \ \forall\, 0 < k \leq 713$$

According to the ***ST-MPC-ESA*** algorithm, first, we have defined an LQ controller $u(t) = K_l^0 x(t)$, $\forall\, l$

$$K_l^0 = \begin{bmatrix} -27.3037 & 0 & -7.6377 & 0 \\ 0 & -27.3037 & 0 & -7.6377 \end{bmatrix}$$

as the terminal controller, and then, we have built an RCI set for each acceleration constraint level $\mathcal{U}_l$, $l = 1, \ldots, 4$. Moreover, to assure the satisfaction of the feasibility condition (25), the following families of one-step controllable sets have been computed:

$$\{\mathcal{T}_1^n\}_{n=1}^{24},\ \{\mathcal{T}_2^n\}_{n=1}^{20},\ \{\mathcal{T}_3^n\}_{n=1}^{16},\ \{\mathcal{T}_4^n\}_{n=1}^{12}$$

where $N_1 = 24$, $N_2 = 20$, $N_3 = 16$, $N_4 = 12$. Each family $\{\mathcal{T}_l^n\}_{n=1}^{N_l}$, $n = 1, \ldots, 4$ has an associated worst-case energy consumption profile (27), that for a single waypoint p_k tracking and for $\alpha_{bat} = 0.00059$ is equals to:

$$
\begin{aligned}
E_{p_k}(\mathcal{U}_1) &= \alpha_{bat} N_1 ||\mathcal{U}_1|| = 0.1402, \quad ||\mathcal{U}_1|| = 9.89 \\
E_{p_k}(\mathcal{U}_2) &= \alpha_{bat} N_2 ||\mathcal{U}_2|| = 0.1669, \quad ||\mathcal{U}_2|| = 14.14 \\
E_{p_k}(\mathcal{U}_3) &= \alpha_{bat} N_3 ||\mathcal{U}_3|| = 0.2003, \quad ||\mathcal{U}_3|| = 21.21 \\
E_{p_k}(\mathcal{U}_4) &= \alpha_{bat} N_4 ||\mathcal{U}_4|| = 0.2503, \quad ||\mathcal{U}_4|| = 35.35
\end{aligned}
$$

The battery is assumed full charged, e.g. $SoC(0) = 100$ and the ***BM*** is configured with $\gamma = 1$ and $\beta = 0$. As a consequence, the minimum time cost J_{time} is used in the opt. (34). Furthermore, we simulated twe following two scenarios:

- (S1) The ***BM*** optimization problem (34) is never online re-computed (i.e the variable "update-wanted=false" $\forall t > 0$). In this case, we denote the obtained input constraint profile as "frozen," namely $\mathcal{U}_{BM}(t) = \mathcal{U}_{BM}^{frozen}(t)$.
- (S2) The ***BM*** optimization problem (34) is periodically solved every 50 waypoints. In this case, we denote the obtained input constraint profile as "updated," namely $\mathcal{U}_{BM}(t) = \mathcal{U}_{BM}^{updated}(t)$.

The obtained simulation results are collected in Figs. 6-8. In Fig. 6, it is possible to appreciate the input constraint profile provided by the ***BM***. In particular, at $t = 0$, the battery manager predicts that it is possible to use an acceleration profile $||\mathcal{U}_{BM}||$ equals to 14.14 until $1.9\,\mathrm{sec}$ and then equals to 9.89 for the rest. This represents a conservative solution, which predicts that the vehicle will reach the destination at $t = 586\,\mathrm{sec}$ and with a remaining state of charge equal $SoC(586) = 0.096$. This profile is kept fixed by the frozen scenario (S1) and updated by (S2) when at $t = 1.9\,\mathrm{sec}$ the 50-th waypoint is reached. At this time, the ***BM*** realizes that the the current state of charge (33) is bigger than the predicted one (32), i.e. $SoC(1.9) = 99.93 \geq \hat{SoC}(1.9) = 92.94$. As a consequence, when the BM optimization (34) is re-solved, an updated and less conservative profile is found. The same reasoning applies to all the other time instance when the optimization is repeated. To better clarify this situation we shall

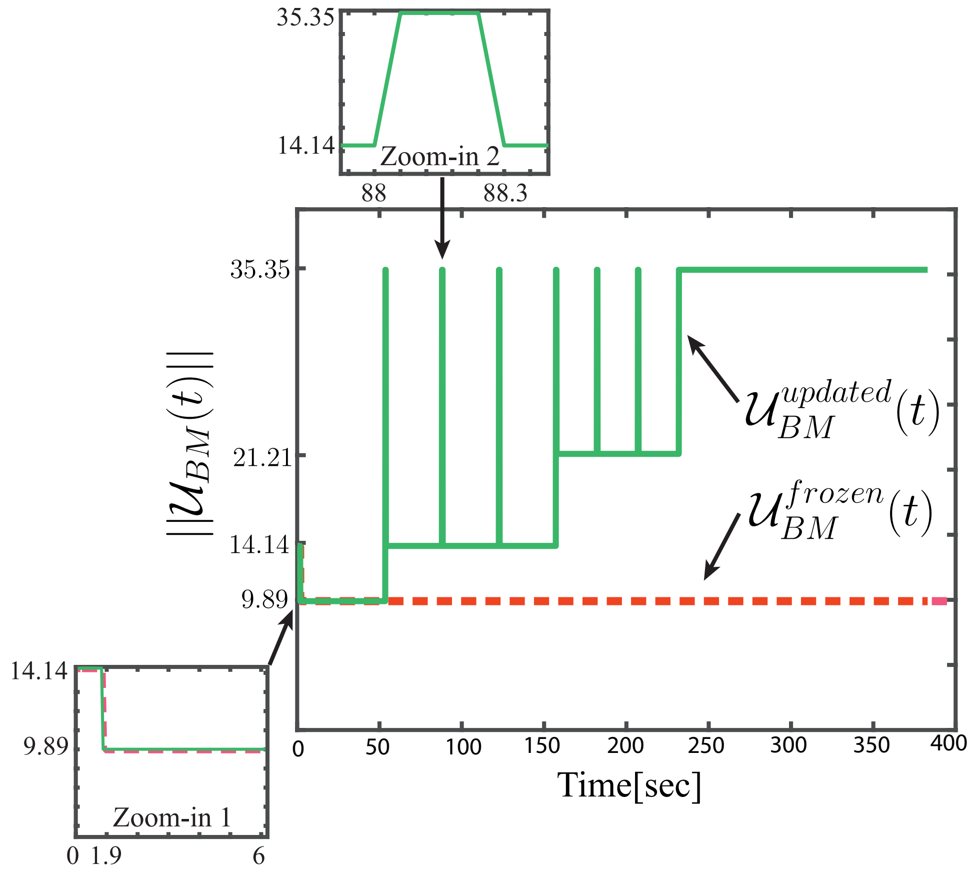

Figure 6: ***BM*** input constraint profile: Frozen (red dashed line) vs. Updated (green line).

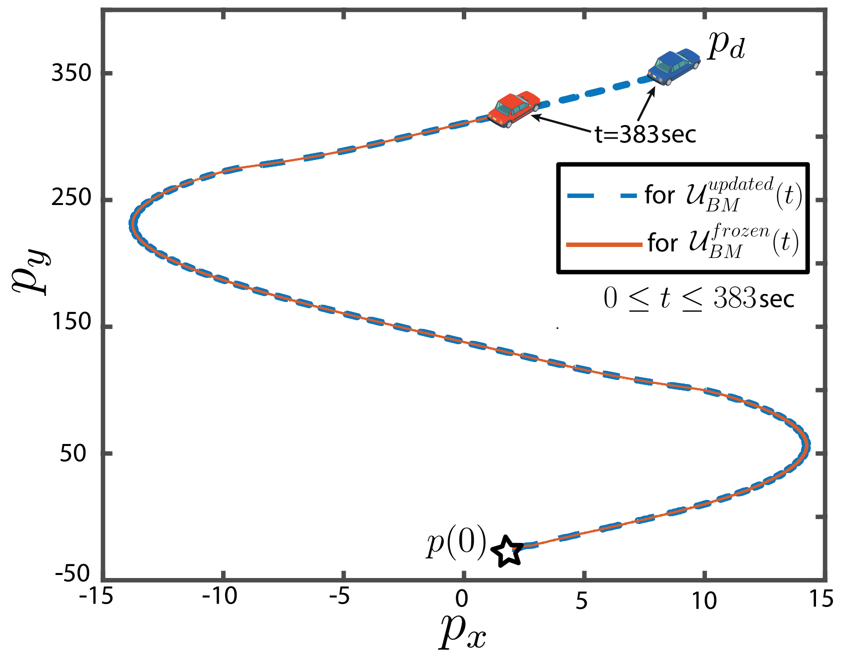

Figure 7: EUV's trajectory for $t \in [0,\ 383]\,\mathrm{sec}$: BM frozen (red solid line) vs. BM updated (blue dashed line).

refer to the “Zoom-in 2" in Fig. 6 and to the actual vehicle’s acceleration norm shown in Fig. 8. In Fig. 8, it is possible to notice that for $t \in [53.9,\, 88.2]\,\mathrm{sec}$ the acceleration vector norm is lower than the maximum allowed value (i.e. 14.14). As a consequence, the vehicle is not using all the predicted amount of energy (worst-case use) and $SoC(88.2) \geq S\hat{o}C(88.2)$. Therefore, by repeating opt. (34) every 50 waypoints a better profile can always be found (in terms of the cost function J_{time}).

The ultimately consequence of updating $\mathcal{U}_{BM}(t)$ is that the vehicle is capable of reaching the destination faster. This is shown in Fig. 7, where it is possible to appreciate that the vehicle with an “updated" ***BM*** profile is capable of reaching the destination at $t = 383\,\mathrm{sec}$, while the same vehicle equipped with a frozen battery manager requires more time, i.e. $t_{end}^{frozen} = 586\,\mathrm{sec}$. Furthermore, the final SoC at the destination, for both the scenarios, are $SoC(end)^{updated} = 51.04\%,\;\; SoC(end)^{frozen} = 67.79\%,$ Finally, the obtained numerical results confirm that by online updating the ***BM***, the EUV performance is improved. Indeed, at $t = 0$, the expected time to reach the target was $t = 586\,\mathrm{sec}$ while the actual destination time is $383\,\mathrm{sec}$.

3.4 Conclusion

In this chapter, we have proposed a novel solution to deal with the battery shortage problem for constrained electric unmanned vehicles. The proposed solution takes advantage of the ST-MPC control paradigm’s peculiar properties to couple the battery management problem with a set-theoretic-based tracking controller. We have shown that the proposed battery manager can offline find a conservative acceleration constraint profile, which assures that the vehicle can reach the target despite any admissible disturbance realization. Moreover, we have shown that such a solution can be improved online using current battery state-of-charge information. The presented simulation results have shown the effectiveness of the proposed control technology.

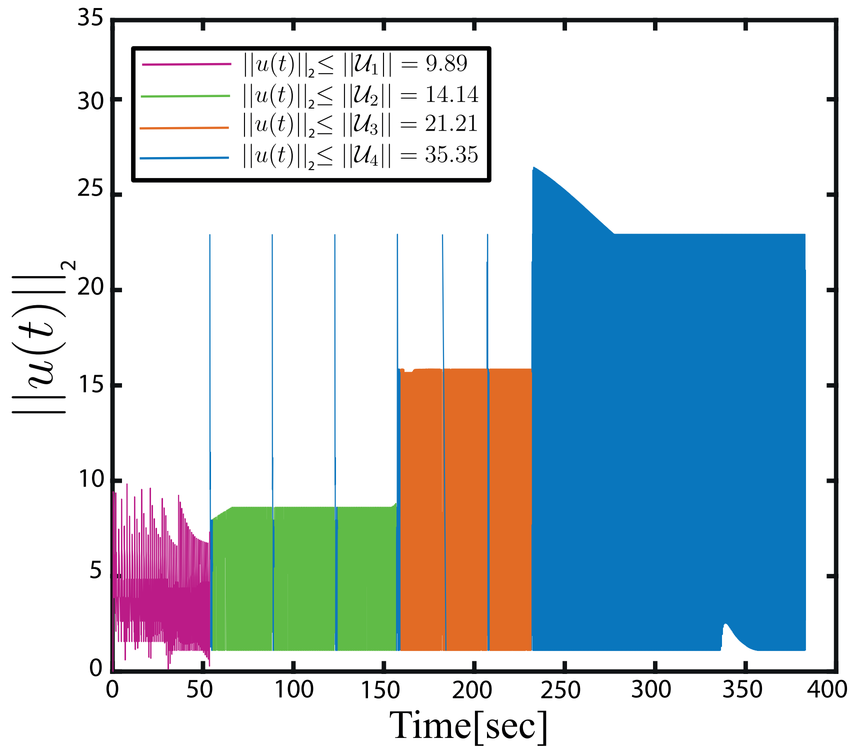

Figure 8: Control signal norm for the updated ***BM*** profile.

Chapter 4

A Receding Horizon Collision Avoidance Strategy for Constrained Multi Unmanned Vehicle Systems

This chapter deals with the reference tracking and collision avoidance control problems for MUVS moving in shared environments. In this contest, we design a centralized TM that, in conjunction with ad-hoc designed local model predictive controllers, can ensure the absence of collisions while minimizing the total vehicle's stops occurrences. The present chapter improves the solution in [48] by proposing a novel collision avoidance strategy that aims to minimize the number of vehicles' stops required to avoid collisions.

The results in this chapter have been in part submitted to the 2021 European Control Conference [51]. Moreover a journal paper is currently under review for the Transactions on Automatic Control (TAC) journal [52].

4.1 Preliminaries and Problem Formulation

4.1.1 UVs Model

Consider UVs moving in a two-dimensional planar environment of coordinates $p = [p^x,\, p^y]^T$ and whose dynamics are described by (1) as a constrained discrete double integrator model as follows [60]:

$$x_i(t+1) = Ax_i(t) + Bu_i(t) + d_i(t) \tag{36}$$

$$A = \begin{bmatrix} I_2 & T_s I_2 \\ 0_2 & I_2 \end{bmatrix},\ B = \begin{bmatrix} \frac{T_s^2 I_2}{2} \\ T_s I_2 \end{bmatrix}$$

where $t \in \mathbb{Z}_+ := \{0, 1, ...\}$ is the sampling time instants, T_s the sampling time interval, $u_i = [a_i^x,\, a_i^y]^T \in \mathbb{R}^2$ the control input vector ($a_i^x \in \mathbb{R}$ and $a_i^y \in \mathbb{R}$ the accelerations along the axes), $x_i = [p_i^T, v_i^T]^T \in \mathbb{R}^4$ the state-space vector ($v_i^T = [v_i^x, v_i^y]^T \in \mathbb{R}^2$ the velocity vector), and $d_i \in \mathbb{R}^4$ a bounded exogenous disturbance such that

$$d_i(t) \in \mathcal{D}_i \subset \mathbb{R}^4 \tag{37}$$

where $\mathcal{D}_i$ is a compact subset containing the origin. Moreover, (36) is subject to the following state and input constraints

$$x_i(t) \in \mathcal{X} := \mathbb{R}^2 \times \mathcal{V},\ \forall t \geq 0 \tag{38}$$

$$u_i(t) \in \mathcal{U}_i \subset \mathbb{R}^2 \tag{39}$$

where $\mathcal{V} \subset \mathbb{R}^2$ (velocity constraint set) and $\mathcal{U}_i$ (accelerations constraint set) are compact subsets with $0_2 \in \mathcal{V}_i$ and $0_2 \in \mathcal{U}_i$, respectively.

Remark 6. *Please note that in this chapter all the information mentioned for each $i-th$ UV is indexed by i. On the other hand, according to (36)-(39), each vehicle can be subject to different bounded disturbances as well as different acceleration constraints.* □

4.1.2 Reference Generators

Each $i-th$ UV follows a reference trajectory, uncoordinated among the vehicles, namely $r_i \in \mathbb{R}^2$, provided by a local path planner module.

Assumption 4. *The vehicle perception capabilities allow the path planner to provide the reference trajectory as a sequence of successive waypoints, namely $r_i(k) \in \mathbb{R}^2$, where $k \in \mathbb{Z}_+$ is discrete variable indexing the $k-th$ waypoint of r_i. Moreover, assuming a limited vision radius, we assume that a finite future sequence of $H_i > 0$ waypoints, namely $R(k_i, H_i)$ is available*

$$R(k_i, H_i) := \{r_i(k_i),\, r_i(k_i+1), \ldots, r_i(k_i+H_i)\} \tag{40}$$

where $r_i(k_i)$ is the current waypoint, and H_i is referred to as the waypoint prediction horizon. Moreover, the planner is assumed to be configured such that the maximum distance between two consecutive waypoints is bounded by $\delta_i > 0$, i.e.

$$||r_i(k_i+1) - r_i(k_i)||_2 \leq \delta_i \tag{41}$$

□

Assumption 5. *Given two UVs, namely UV_i and UV_j, we assume absence of collisions if $\|p_i(t) - p_j(t)\| > 0$, $\forall\, t$.* □

4.1.3 Definitions

Definition 6. *[61] (**Undirected Graph**) An undirected graph is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the vertex (or node) set and $\mathcal{E}$ is edge set which is defined as a finite subset of all* *admissible unordered pairs in $\mathcal{V}$, i.e. $\mathcal{E} \subset \{e_{ij} := \{i,j\} : i,j \in \mathcal{V}\}$.*

Definition 7. *[61] (**Adjacency Matrix**) Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the adjacency matrix $\mathcal{A}[\mathcal{G}]$ is a squared symmetric matrix such that*

$$\mathcal{A}[\mathcal{G}]_{ij} = \begin{cases} 1 & \textit{if} \quad e_{ij} \in \mathcal{E} \\ 0 & \textit{otherwise} \end{cases} \tag{42}$$

Definition 8. *[61] (**Degree Matrix**) Given an undirected graph $\mathcal{G}$, the adjacency degree of each vertex $v_i \in \mathcal{V}$, namely $d(v_i)$, is given by the number of vertices $v_j \in \mathcal{V}$ connected to v_i with an edge, i.e. $(v_i, v_j) \in \mathcal{E}$. The degree matrix $\Delta[\mathcal{G}]$ is defined as the diagonal matrix containing the adjacency degrees for all the vertices, i.e.*

$$\Delta[\mathcal{G}] = \begin{pmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & d(v_n) \end{pmatrix} \tag{43}$$

4.1.4 Problem Formulation

Consider a set $\mathcal{I} := \{1, 2, \ldots, S\}$ of S heterogeneous UVs moving in a shared environment, where each UV is described by (36)-(39) and equipped with a planner module satisfying the requirements (40)-(41) stated in *assumption* 4. Design a control illustrated in Fig. 9 architecture such that

- (***O1***) the UV's decentralized controllers are able to fulfill the constraints (38)-(39) and sequentially track the waypoints $r_i(k_i), \forall k_i$.
- (***O2***) a centralized TM is capable of predicting future collisions within the waypoint prediction horizons H_i, and minimizing collision possibilities by appropriately imposing different velocity constraints to the UV's controllers. Moreover, TM must guaranty the absence of collisions.

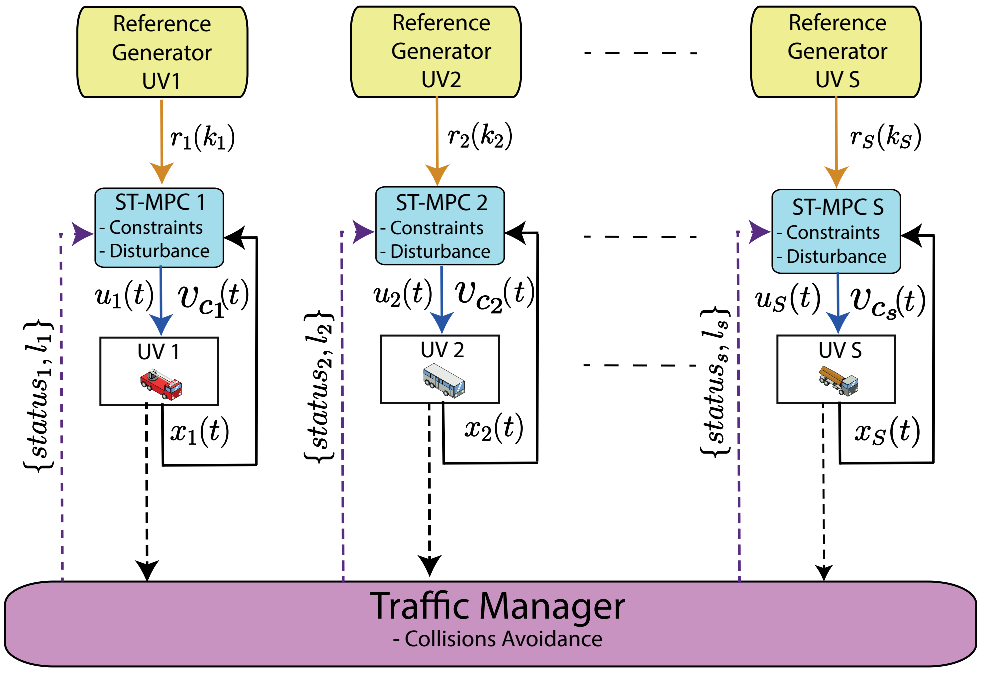

Figure 9: Proposed Control Architecture

4.2 Proposed Solution

In this section, a solution to the control objectives (***O1***)-(***O2***) is presented. The developments proceed as follows. First, an ad-hoc designed waypoint tracking controller is proposed; then, the traffic manager is designed, and its computable algorithm is illustrated.

4.2.1 Waypoint Tracking Controller

Here, the proposed waypoint tracking controller extends the set-theoretic MPC solution in [48] to deal with time-varying velocity constraints arising from the desired TM operations (see (***O2***)).

If we assume a time-invariant state constraint $\mathcal{X}$, a solution for the waypoint tracking problem is the following:

Offline operations:

The offline operations of ST-MPC are detailed in Subsection. 2.2.1 in Chapter. 2.

Online operations (assuming $x_i(0) \in DoA(r_i(0))$):

ST-MPC - Waypoint Tracking Controller

```
1: Compute $n_i(t) := \min_{0 \le n_i \le N_i} n_i : x_i(t) \in \mathcal{T}_i^{n_i}(r_i(k_i))$
2: if $n_i(t) == 0$ and the successive waypoint exists and it is enabled, then $k_i \leftarrow k_i + 1$, goto Step 1
3: end if
4: $x^{eq} \leftarrow x^{eq}_{r_i(k_i)}$
5: if $n_i(t) == 0$ then $u_i(t) = K_i^0(x_i(t) - x^{eq}_{r_i(k_i)})$
6: else Find $u_i(t)$ by solving the following optimization problem
       $$\begin{aligned} u_i(t) = \arg\min_{u_i} \|Ax_i(t) + Bu_i - x^{eq}_{p_i}\|_2^2 \quad s.t. \\ Ax_i(t) + Bu_i \in \tilde{\mathcal{T}}_i^{n_i(t)-1}(r_i(k_i)),\ u_i(t) \in \mathcal{U}_i \end{aligned} \tag{44}$$
7: end if
8: $t \leftarrow t + 1$ and goto Step 1
```

Remark 7. *Please note that the above waypoint tracking controller enjoys, by construction, recursive feasibility if the state constraints $\mathcal{X}$ is time-invariant. Moreover, it is essential to remark that, if necessary, a vehicle can be safely stopped in the terminal region associate with the current waypoint $r_i(k_i)$, i.e. $\mathcal{T}_i^0(r_i(k_i))$, by disabling the waypoint update in Step 2. TM will later exploit the latter to stop a vehicle and avoid collisions (see section 4.2.2).* □

Given the desired TM operations (see *(O2)*), the waypoint tracking controller must be able to accommodate time-varying velocity constraints. To this end, the following assumption is made:

Assumption 6. *A finite set of $L > 1$ admissible velocity constraints can be imposed for each vehicle, i.e.*

$$V = \{\mathcal{V}_1, \mathcal{V}_2 \ldots, \mathcal{V}_L\},\ \textit{with}\ \mathcal{V}_1 \subset \mathcal{V}_2 \subset \ldots \subset \mathcal{V}_L \equiv \mathcal{V} \tag{45}$$

In the sequel, the state constraint

$$x_i(k) \in \mathcal{X}_{l_i} := \mathbb{R}^2 \times \mathcal{V}_{l_i},\, l_i \in \{1, \dots, L\} \tag{46}$$

is used to denote time-varying state constraint (38) according to (45). □

The time-varying state constraints (46) can potentially invalidate the recursive feasibility of the tracking controller for the following reasons:

- If the state constraint (38) is used to offline build the controller, then, online, if $\mathcal{X} \to \mathcal{X}_{l_i}$ with $l_i < L$, $x_i(t)$ might be outside of the admissible region, i.e. $x_i(t) \notin \mathcal{X}_{l_i}$.
- Since in (44) the state constraints are implicitly embedded in the offline built robust one-step controllable sets $\{\mathcal{T}_i^{n_i}(r_i(k_i))\}_{n=1}^{N_i}$, then, online, if $\mathcal{X} \to \mathcal{X}_{l_i}$ with $l_i < L$, and $x_i(t) \in \mathcal{T}_i^{n_i}$, $n_i \leq N_i$, $x_i(t) \in \mathcal{X}_{l_i}$, then the one-step evolution $x_i(t+1) \in \mathcal{T}_i^{n_i-1}$ is not guaranteed to belong to $\mathcal{X}_{l_i}$.

The above drawbacks are here mitigated as follows:

1. Offline

   - The terminal region $\mathcal{T}_i^0(r_i(0))$ is computed as prescribed in (Off-1) and by considering instead of $\mathcal{X}$, the the worst-case velocity constraint scenario, i.e. $x_i(t) \in \mathcal{X}_1$.
   - For each admissible velocity level l in (45), a family of robust one-step controllable set, namely $\{\mathcal{T}_{(i,l)}^{n_i}(r_i(0))\}_{n_i=1}^{N_{(i,l)}}$, is computed according to (Off-2) with $\mathcal{X}_l$ instead of $\mathcal{X}$.

2. Online

   - State constraint switches $\mathcal{X}_{l_1} \to \mathcal{X}_{l_2}$, $l_1, l_2 \leq L$ and, as a consequence, one-step controllable sets switches $\{\mathcal{T}_{(i,l_1)}^{n_i}(r_i(0))\}_{n_i=1}^{N_{(i,l_1)}} \to \{\mathcal{T}_{(i,l_2)}^{n_i}(r_i(0))\}_{n_i=1}^{N_{(i,l_2)}}$, are enabled when the vehicle state $x_i(t)$ enters the domain of attraction of the controller using the current state constraint ($\mathcal{X}_{l_2}$), i.e.

$$x_i(t) \in \bigcup_{n=0}^{N} \{\mathcal{T}^{n}_{(i,l_2)}(r_i(0))\}_{n=1}^{N_{(i,l_2)}} \tag{47}$$

The following proposition formally proves the effectiveness of the proposed solution.

Proposition 3. *If $\mathcal{T}^0$ is computed as in (Off-1) with $\mathcal{X}_1$ instead of $\mathcal{X}$, a family of robust one-step controllable set $\{\mathcal{T}^{n_i}_{(i,l)}(r_i(0))\}_{n_i=1}^{N_{(i,l)}}$ is built $\forall\, l \in \{1, 2, \dots L\}$ as in (Off-2) with $\mathcal{X}_l$ instead of $\mathcal{X}$, and state-constraint switches $\mathcal{X}_{l_1} \to \mathcal{X}_{l_2}$ with $1 \leq l_1, l_2 \leq L$ are allowed when the condition (47) is satisfied, then the ST-MPC algorithm preserves recursive feasibility.*

Proof. The switching condition (47) ensures that constraints switches $\mathcal{X}_{l_1} \to \mathcal{X}_{l_2}$ occur when the current vehicle's state $x_i(t)$ belongs to the domain of attraction of the controller computed according to the desired new state constraints, e.g. $\mathcal{X}_{l_2}$. As a consequence, the optimization problem (44) is always admissible. Moreover, since all the controller shares, by construction, the same terminal region $\mathcal{T}_i^0$, the switching condition (47) is fulfilled, in the worst-case scenario, when $x_i(t)$ reaches, in a finite number of steps N_{l_1}, the terminal region associated to the current waypoint. $\square$

Remark 8. *Please note that the condition (47) is always instantaneously verified for any state constraint switch $\mathcal{X}_{l_1} \to \mathcal{X}_{l_2}$ where $l_2 > l_1$, i.e. whenever the new desired velocity constraint relaxes the current one $(\mathcal{V}_{l_1} \subset \mathcal{V}_{l_2})$.*

4.2.2 Traffic Manager (TM) and Collision Avoidance Strategy

In this subsection, starting from the UV's tracking controller's properties, a TM module fulfilling the objective *(O2)* is designed. By assuming the following exchange of data from the vehicles' local controller and the TM:

- At $t = 0$ (or offline): each $i - th$ vehicle sends $\{\mathcal{T}_i^{n_i}(r_i(0))\}_{n_i=0}^{N_i} = \{\mathcal{T}^{n_i}_{(i,L)}(r_i(0))\}_{n_i=0}^{N_i}$ and the associated controller domain $DoA^{N_i}(r_i(0))$.

- $\forall t \geq 0$ (online): each $i-th$ vehicle sends the predicted waypoints $R(k_i, H_i)$ and the current set-membership index $n_i(t)$ (see Step 1 of the ST-MPC waypoint tracking controller)

the TM operations are twofold:

1. Impose the vehicle's velocity constraints (45) in order to minimize the possibility of collisions between the vehicles.
2. Ensure the absence of collisions, by stopping, whenever strictly necessary, the minimal subset of vehicles.

and its actions can be modeled by means of the following function

$$\{status_i, l_i\}_{i=1}^{S} = TM\left(\{R(k_i, H_i), n_i(t)\}_{i=1}^{S}\right) \tag{48}$$

The first output $status_i = \{go, stop\}$ is a binary variable that defines if each vehicle can advance to the next waypoint or it needs to stop (e.g. to avoid a collision). The second denotes the velocity constraint level $\mathcal{V}_{l_i} \in \{\mathcal{V}_1, \ldots, \mathcal{V}_L\}$ that must be used by each vehicle (e.g. to minimize the chances of future collisions).

Remark 9. *Please note that the TM can stop/confine a vehicle in the terminal region of the current waypoint by disabling the waypoint update in Step 2 of the ST-MPC algorithm, see also remark 7.* □

The steps taken by the traffic manager to determine the TM outputs are the following:

- *TM-Step 1:* it computes an undirected connectivity graph $\mathcal{G} := (\mathcal{I}, \mathcal{E})$, where the nodes are the vehicles and the edges $e_{(i_1,i_2)} \in \mathcal{E},\ i_1, i_2 \in \mathcal{I}$ contain information regarding possible collisions within the waypoint prediction horizons;
- *TM-Step 2:* it checks if there are collision possibilities regarding the current waypoints (imminent collisions) and, if any, it solves the problem by defining the minimum set of vehicles that must be stopped to avoid collisions;

- *TM-Step 3:* by removing from $\mathcal{G}$ all the nodes (vehicles) that have been stopped, the connected components $\{\mathcal{C}_z\}_{z=1}^{Z},\ Z \geq 1$ are found. For each $\mathcal{C}_z$, the velocity level for each vehicle are determined to minimize the possibility of future collisions.

In the remaining of the section, the operations performed in each step are detailed.

TM-Step 1: collision graph

Given the families of robust one-step controllable sets $\{\mathcal{T}_i^{n_i}(r_i(0))\}_{n_i=0}^{N_i}$ and $DoA^{N_i}(r_i(0)),\ \forall\, i \in \mathcal{I}$, the sets of predicted waypoint $R(k_i, H_i),\ \forall\, i \in \mathcal{I}$, and the set-membership indices $n_i(t),\ \forall\, i \in \mathcal{I}$, each edge $e_{(i,j)}$ of $\mathcal{G}$ contains all the potential collisions between the vehicles i and j $\forall i, j \in \mathcal{I},\ i \neq j$:

$$\begin{aligned} e_{(i,j)} = \{ & k_i + \bar{h}_i,\ \bar{h}_i \in \mathcal{H}_i : \exists\, \bar{h}_j \in \mathcal{H}_j\ s.t. \\ & DoA^{N_i}(r_i(k_i + \bar{h}_i)) \cap DoA^{N_j}(r_j(k_i + \bar{h}_j)) \neq 0\} \end{aligned} \tag{49}$$

where $\mathcal{H}_i = \{1, \dots, H_i\}$, and $\mathcal{H}_j = \{1, \dots, H_j\}$. In addition to (49), the collision point $\bar{h}_i = k_i$ (i.e. potential collision with the current waypoint $r_i(k_i)$) is added to $e_{(i,j)}$, i.e. $e_{(i,j)} = \{k_i, e_{(i,j)}\}$ if the following condition is verified

$$\bigcup_{n_i=0}^{\max(n_i(t)-1,0)} \{\mathcal{T}_i^{n_i}(r_i(k_i))\} \bigcap \bigcup_{n_j=0}^{\max(n_j(t)-1,0)} \{\mathcal{T}_j^{n_j}(r_j(k_j))\} \neq 0 \tag{50}$$

Remark 10. *Please note that according to the used controller, the exact trajectory followed by each vehicle is not a-priori known. However, given the list of future waypoints, then it is possible to compute the robust reachable tube where the trajectory is confined withing the waypoint prediction horizon, i.e.*

$$x_i(t) \in \bigcup_{\bar{h}_i=0}^{H_i} DoA^{N_i}(r_i(k_i + \bar{h}_i))$$

Therefore, as in (49), per each waypoint it is possible to verify if, in the worst-case scenario, collisions (i.e. reachable tubes intersections) are possible. On the other hand, the condition (50) exploits the information on current set-membership index $n_i(t)$, to define, for the current waypoint, a less conservative collision condition based on the controller property that each *vehicle's one-step evolution, $x_i(t+1)$, will be inside a controllable set strictly inside the current one, i.e. $n_i(t+1) < n_i(t)$.* $\square$

Remark 11. *Please note that the graph $\mathcal{G} := (\mathcal{I}, \mathcal{E})$ must be fully computed only at $t = 0$. Indeed, in a receding horizon fashion, for $t > 0$, only a portion of $\mathcal{G}$ must be updated. Let denote with k_i^{old} and k_i the actual waypoint for the $i-th$ vehicle at the time instant $t-1$ and t, respectively. Then, at t, an edge $(i,j) \in \mathcal{E}$ requires an update if either $k_i = k_i^{old}+1$ or $k_j = k_j^{old}+1$. Moreover, if an update is required the following actions must be taken:*

- *(R1) - Remove $\bar{h}_i = k_i^{old}$ from $e_{(i,j)}$ (if present), i.e. $e_{(i,j)} = e_{(i,j)} \setminus k_i^{old}$;*
- *(R2) - Remove $\bar{h}_i = k_i$ from $e_{(i,j)}$ (if present), i.e. $e_{(i,j)} = e_{(i,j)} \setminus k_i$, if the condition (50) is not satisfied;*
- *(R3) - Add $\bar{h}_i = k_i + N_i$, to $e_{(i,j)}$, i.e. $e_{(i,j)} = \{e_{(i,j)}, k_i + N_i\}$, if $\exists\, \bar{h}_j \in \mathcal{H}_j$ s.t.:*

$$DoA^{N_i}(r_i(k_i+N_i)) \cap DoA^{N_j}(r_j(k_j+\bar{h}_j)) \neq 0.$$

TM-Step 2: vehicles to be stopped

Once the connectivity graph $\mathcal{G} := (\mathcal{I}, \mathcal{E})$ is built according to (49)-(50), the first action taken by the traffic manager is to identify the vehicles that might have a collision in the current waypoints k_i. To this end, the following aspects must be first highlighted:

- if no collisions where predicted at $t-1$, then at t collisions are possible if only if a vehicle wants to move to the successive waypoint, i.e. $k_i = k_i^{old}+1$
- if $k_i = k_i^{old}+1$, then the $i-th$ vehicle is currently in the terminal region of the previous waypoint, i.e. $x(t) \in \mathcal{T}_i^0(r_i(k_i^{old})$. The latter implies that, if necessary, the $i-th$ vehicle can be safely stopped in the current terminal region.
- if $k_i = k_i^{old}$, then the $i-th$ vehicle is not yet in the terminal region of the current waypoint, and it cannot be stopped.

Let's denote with $\mathcal{I}^{sw} \subseteq \mathcal{I}$ the subset of vehicles switching waypoint ($k_i^{old} \to k_i^{old}+1 = k_i$), and with $\mathcal{G}^{sw} = (\mathcal{I}, \mathcal{E}^{sw})$ the graph modeling collision in the current waypoint, i.e.

$$e^{sw}_{(ij)}(t) = \begin{cases} 1 & \text{if (50) is satisfied} \\ 0 & \text{otherwise} \end{cases} \tag{51}$$

Then, to stop the minimum number of UVs, namely $\mathcal{I}^{stop} \subset \mathcal{I}^{sw}$, the following procedure is used:

- (C1) - Find the vehicle $i \in \mathcal{I}^{sw}$ with the highest degree, i.e

$$i = \arg\max_{i \in \mathcal{I}^{sw}} \Delta[\mathcal{G}^{sw}] \tag{52}$$

  with $\Delta[\mathcal{G}^{sw}]$ the degree matrix containing the adjacency degrees.
- (C2) Add i to $\mathcal{I}^{stop}$ and remove i from $\mathcal{G}^{sw}$
- (C3) If $\exists\, e^{sw}_{(i,j)}(t) \in \mathcal{E}_{sw}(t) : e^{sw}_{(ij)}(t) \neq 0$, then *goto* Step 1, else *stop* the procedure.

TM-Step 3: future collision minimization

After the set of vehicle to be stopped is determined, i.e. $\mathcal{I}^{stop}$, the graph of not stopped vehicles, namely, $\mathcal{G}^{ns} = (\mathcal{I}^{ns}, \mathcal{E}^{ns})$, is here taken into consideration. Specifically, $\mathcal{I}^{ns} = \mathcal{I} \backslash \mathcal{I}^{stop}$ and $e^{ns}_{(i,j)} \in \mathcal{E}^{ns}$ if $(i, j) \in \mathcal{E}$ and $i, j \in \mathcal{I}^{ns}$.

Given $\mathcal{G}^{ns}$, and by resorting to the Laplacian matrix of $\mathcal{G}^{ns}$, first all the connected components $\{\mathcal{C}_z\}_{z=1}^{Z}$, where $\mathcal{C}_z \subseteq \mathcal{I}^{ns}$, are found [62].

Remark 12. *Please note that if two vehicles $i, j \in \mathcal{I}^{ns}$ do not belong to the same connected component $\mathcal{C}_z$, then no collisions are forecast between i and j within the waypoint prediction horizon. Moreover, if a connected component consists of a single node, e.g., the component's cardinality is 1 ($|\mathcal{C}_i| = 1$), then any velocity level l_i can be used by the vehicle i. However, if a collision possibility in the current waypoint was predicted for i, then $l_i = L$ is imposed to make sure that the vehicle leaves the collision area as soon as possible.* □

For each $i \in \mathcal{C}_z$, $|\mathcal{C}_z| > 1$, the distance to the closest collision point, namely $d^c_i \in \mathbb{R}$, with

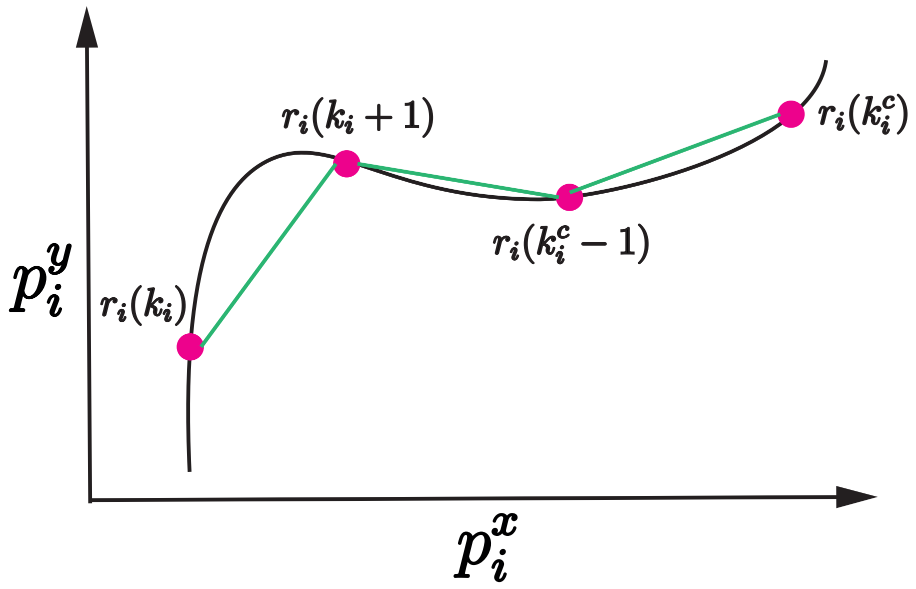

Figure 10: Waypoint distance approximation (53)

any $j \neq i \in \mathcal{C}_z$ is computed as follows:

$$d_i^c = \sum_{h=k_i}^{k_i^c-1} \|p_i(h+1) - p_i(h)\|_2, \;\; i \in \mathcal{C}_z \tag{53}$$

where

$$k_i^c = \min \{k_{(i,j)}^c, \; i,j \in \mathcal{C}_z : \; k_{(i,j)}^c = \min e_{i,j}\} \tag{54}$$

Remark 13. *It is important to underline that the exact trajectory followed by each vehicle is not a-priori known. Therefore, (54) approximates the distance to the closest waypoint as the summation of the 2-norm distance between the waypoints, see Fig. 10*

The computed distances d_i^c are collected in an ordered vector

$$d^c = [d_{v_{c_1}}^c, \dots, d_{v_{c_z}}^c] \tag{55}$$

such that v_{c_1} and v_{c_z} are the vehicles at the maximum and minimum distance from a collision waypoint, respectively, and c_z is the cardinality of $\mathcal{C}_z$, i.e. $|\mathcal{C}_z| = c_z$. Then, for each $v_{c_i} \in \mathcal{C}_z$, the velocity level $l_{v_{c_i}} \in \{1, \ldots, L\}$ is assigned, i.e.

$$\begin{array}{c} v_{c_1} \rightarrow l_{v_{c_1}} \\ \vdots \\ v_{c_z} \rightarrow l_{v_z} \end{array}, \; s.t. \quad \begin{array}{c} l_{v_{c_1}} \leq l_{v_{c_1}+1} \leq \ldots < l_{v_{c_z}} \\ l_{v_{c_i}} \in \{1, \ldots, L\}, \, \forall c_i \in \mathcal{C}_z \end{array}$$

according to the solution of the following optimization problem

$$\begin{aligned} & \max_{l_{v_{c_1}}, \ldots, l_{v_{c_z}}} \sum_{l=1}^{v_{c_z - 2}} |l_{v_{c_1}+l+1} - l_{v_{c_1}+l}| \; s.t. \\ & l_{v_{c_1}} \leq l_{v_{c_1}+1} \leq \ldots < l_{v_{c_z}-1} \\ & l_{v_{c_i}} \in \{1, \ldots, L\}, \, \forall c_i \in \mathcal{C}_z \end{aligned} \tag{56}$$

Remark 14. *The optimization (56) prescribes that the highest velocity constraint is assigned to the vehicle closer to a potential collision. Moreover, to the reaming vehicles, a velocity constraints inverse proportional to the distance d_{v_i} (slowed down) is given. Please note that the cost function in (56) aims to maximize the distance, in term of velocity levels, among all the vehicles. The objective of such a heuristic solution is to allow the closest vehicle to overcome the collision point before other vehicles could reach the same collision area.* □

Finally, to better clarify the *TM-Step 1- TM-Step 3* operations, we can refer to Fig. 11, where such tasks are shown for a team of 8 UVs, i.e. $\mathcal{I} = \{1, 2, \ldots, 8\}$. The subplot Fig. 11.a shows the connectivity graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ built in *TM-Step 1*, where solid lines between any two vehicles i and j denote risk of collisions in the waypoint prediction horizons $\bar{h}_i \in \mathcal{H}_i$ and $\bar{h}_j \in \mathcal{H}_j$, see (49), and dashed links define possibility of collisions in the current waypoints k_i and k_j, see (50). Moreover, red circles nodes denotes vehicles that did not change the waypoint ($k_i = k^i_{old}$), while green circled nodes defines vehicles that have switched waypoint $k_i = k_i^{old} + 1$. The subplot Fig. 11.b shows the operations in *TM-Step 2*. In particular the graph $\mathcal{G}^{sw}$ of the current collisions is considered (see 51) and the minimum number of vehicles is stopped according to the (C1)-(C3) procedure, i.e. $\mathcal{I}^{stop} = \{3\}$. In subplot Fig. 11.c, the operations in *TM-Step 3* are shown. Specifically, it show that all the connected components in the graph of non stopped vehicles ($\mathcal{G}^{ns}$) are found, e.g. $\mathcal{C}_1 = \{1,2,6,7\}$, $\mathcal{C}_2 = \{4,5\}$, $\mathcal{C}_3 = \{8\}$. Such components are finally used to assign a velocity constraint to each vehicles according to (53)-(56).

TM Algorithm

All the above TM operations are here summarized into the following computational algorithm:

Traffic Manager *(TM)* $\forall\, t$

```
1: Compute ($t = 0$) or update ($t > 0$) the connectivity graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ as prescribed by (49)-(50) and (R1)-(R3), respectively.
2: Given $\mathcal{I}^{sw}$, find the vehicles to be stopped ($\mathcal{I}^{stop} \subseteq \mathcal{I}^{sw}$) by applying the procedure (C1)-(C3);
3: Consider $\mathcal{G}^{ns} = (I \setminus \mathcal{I}^{stop}, e^{ns}_{(i,j)})$, $(i,j) \in \mathcal{E}$, $i, j \notin \mathcal{I}^{stop}$ and found all its connected components $\{\mathcal{C}_z\}_{z=1}^{Z}$;
4: for all $\mathcal{C}_z$ do
5:     for all $i \in \mathcal{C}_z$ do
6:         Compute the distance $d_i^c$ from the closest collision point as in (53)
7:     end for
8:     Collect the distances for all the vehicles in an ordered vector $d^c$ as in (55);
9:     Assign to each vehicle $v_{c_i}$ a velocity constraint level $l_{v_{c_i}}$ by solving the opt. (56);
10: end for
11: Send $status_i = stop$ (e.g. disable the waypoint switch in Step 2 of ST-MPC) to all vehicle $i \in \mathcal{I}^{stop}$. Send $status_i = go$ to the others;
12: $\forall\, i \notin \mathcal{I}^{stop}$ send the computed velocity constraint level $l_{v_{c_i}}$.
```

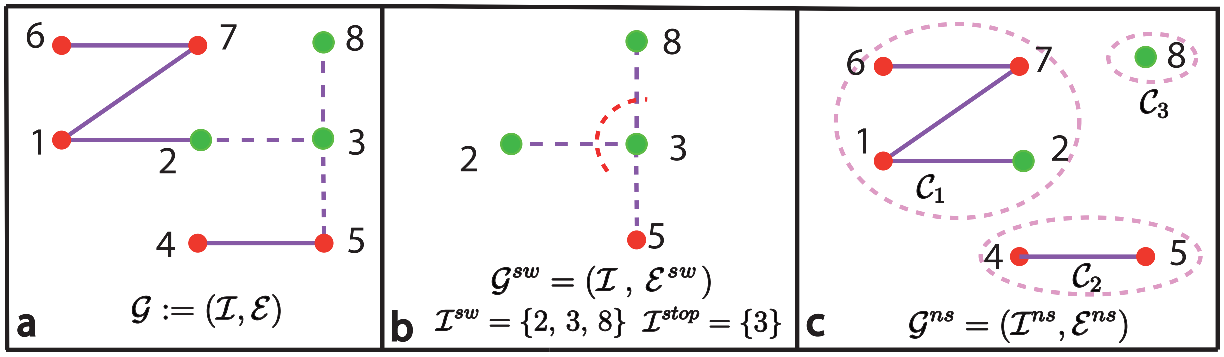
6
7
8
1
2
3
4
5
a
$\mathcal{G} := (\mathcal{I}, \mathcal{E})$
8
2
3
5
$\mathcal{G}^{sw} = (\mathcal{I}, \mathcal{E}^{sw})$
b
$\mathcal{I}^{sw} = \{2, 3, 8\}$ $\mathcal{I}^{stop} = \{3\}$
6
7
8
$\mathcal{C}_3$
1
2
$\mathcal{C}_1$
4
5
$\mathcal{C}_2$
c
$\mathcal{G}^{ns} = (\mathcal{I}^{ns}, \mathcal{E}^{ns})$

Figure 11: Traffic Manager Operations: *TM-step 1* (subplot (a)), *TM-step 2* (subplot (b)), *TM-step 3* (subplot (c))

Remark 15. *The computational complexity of the proposed TM algorithm is mainly related to the update of $\mathcal{G}(t)$. In particular, to test if between two vehicles i and j there is a collision possibility (i.e., $e_{ij}(t) = 1$), then a set-membership test must be performed. Assuming a polyhedral representation for the robust one-step controllable sets $\mathcal{T}_i^n$, each test requires the solution of a simple Linear Programming (LP) optimization problem solvable in polynomial time. Therefore, to update $\mathcal{G}(t)$, $|\mathcal{I}^{sw}(t)|(S-1)$ LP problems must be solved for both the updating steps (R2) and (R3), see Remark. 11. Therefore, at each graph update, the total number of LP problems to be solved is:*

$$2|\mathcal{I}^{sw}(t)|(S-1)$$

where $|\mathcal{I}^{sw}|$ is the number of vehicles making a waypoint switch request at the time t and S is the total number of vehicles. On the other hand, the local ST-MPC is mainly related to the solution of the QP optimization problem defined in (44). Therefore, contrary to the existing DMPC solutions, the proposed approach does not require inter-vehicle communications and non-convex optimizations. As an example, at each iteration, the distributed approach in [46] requires the computation of S MILPs while the proposed solution requires $2S(S-1)$ LPs (worst-case) and 1 *QP per vehicle.*

Proposition 4. *Consider a set $\mathcal{I}$ of UVs modeled as in (36)-(39), equipped with the ST-MPC waypoint tracking controller. If the UVs start from a feasible collision-free initial conditions*

Task Solved	Proposed Sol.	[46]
Collision Avoidance	$H_i = 1: \quad S(S-1)$LPs	(S) MILPs
	$H_i > 1: \quad 2S(S-1)$LPs	
Reference Tracking	(1) QP per vehicle	

Table 1: Computational Cost: proposed solution vs [46]

satisfying the condition (50), the TM operations guarantees the absence of collisions, regardless of the UVs reference trajectories.

Proof. The proof can be obtained collecting all the above developments. First, if the UVs start form a feasible condition satisfying (50), then no collisions are possible until any of the vehicles requires a waypoint switch. The latter finds justification in the fact that if $x_i(0) \in \mathcal{T}_i^{n_i(t)}(r_i(k_i))$ the one-step evolution will belong to a set included within $\mathcal{T}_i^{n_i(t)}(r_i(k_i))$, i.e. $x_i(t+i) \in \mathcal{T}_i^{n_i(t+1)}(r_i(k_i)) \subset \mathcal{T}_i^{n_i(t)}(r_i(k_i))$, with $n_i(t+1) < n_i(t)$ (see [33]). Moreover, when waypoint switches $r(k_i) \to r(k_i+1)$ occur, Step 2 of the *TM* algorithm ensures that potential collisions are avoided by stopping the minimum number of vehicles that make the connectivity graph $\mathcal{G}^{sw}$ completely disconnected. The latter is equivalent to ensure that families of robust one-step controllable set $\{\mathcal{T}_i^n(r_i(k_i))\}$ don't have any intersection, so guaranteeing that no admissible state-trajectories could lead to collisions. □

4.3 Simulation

In this section, the proposed control architecture's effectiveness is testified by means of two simulation examples where the whole system has been emulated using MATLAB, and the MPT3 toolbox has been used to implement the proposed strategy [55]. The following scenarios have been considered:

- (S1) In the first scenario, 10 UVs, i.e. $\mathcal{I} = \{1, \ldots, 10\}$, are considered. Moreover, only one admissible velocity constraint is assumed, i.e. $L = 1$, and the prediction horizon is fixed to one, i.e. $H_i = 1$. This simulation aims to evaluate the capability of the proposed solution to avoid collisions when the minimum prediction horizon is considered.

- (S2) In the second scenario, 2 UVs, i.e. $\mathcal{I} = \{1, 2\}$, are considered. Moreover, three admissible velocity levels are assumed, i.e. $L = 3$. In this setup, the strategy is evaluated for a set of different waypoint prediction horizons, i.e. ($1 \leq H_i \leq 300$, $i \in \mathcal{I}$) in order to investigate the beneficial effects of the waypoint precition horizon.

4.3.1 Scenario One

The UVs dynamics are described by (36) where $T_s = 0.1\,\mathrm{sec}$ and subject to the disturbance set and the constraints as follows

$$\begin{gathered} d_i(t) \in \mathcal{D}_i = \{d \in \mathbb{R}^4 : |d(s)| \leq \bar{d}_i,\ s = 1,\ 2\} \\ \mathcal{V} = \{v \in \mathbb{R}^2 : |v(s)| \leq \bar{v}_i,\ s = 1, \ldots, 10\} \\ u_i(t) \in \mathcal{U}_i = \{u \in \mathbb{R}^2 : |u_i(s)| \leq 8,\ s = 1,\ 2\} \end{gathered}$$

where s denotes the $s - th$ component of each vector. And for,

$$\begin{gathered} i \in \{1, 2, 3\} \rightarrow \bar{u}_i = 20,\ \bar{d}_i = 0.06 \\ i \in \{4, 5\} \rightarrow \bar{u}_i = 25,\ \bar{d}_i = 0.085 \\ i \in \{6, 9, 10\} \rightarrow \bar{u}_i = 8,\ \bar{d}_i = 0.07 \\ i \in \{7, 8\} \rightarrow \bar{u}_i = 18,\ \bar{d}_i = 0.065 \\ \forall i \rightarrow \bar{v}_i = 4 \end{gathered}$$

Each UV's reference generator provides a sequence of waypoints $r_i(k_i)$ as shown in Fig. 12. The maximum distance δ_i, $\forall\, i \in \mathcal{I}$ between two successive waypoints are:

$$\begin{gathered} i \in \{1, 2, 3\} \rightarrow \delta_i = 4.02 \quad i \in \{4, 5\} \rightarrow \delta_i = 5.62 \\ i \in \{6, 9, 10\} \rightarrow \delta_i = 3.22 \quad i \in \{7, 8\} \rightarrow \delta_i = 3.91 \end{gathered}$$

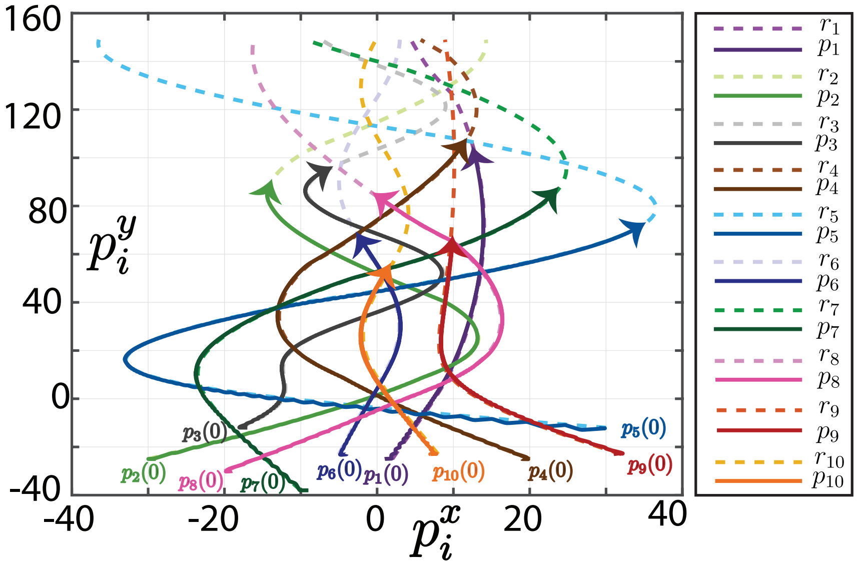

Figure 12: UV Waypoints and trajectories for $t \in [0,\ 100]s$.

According to the proposed ST-MPC strategy, a terminal controller and an RCI set have been offline computed for each vehicle as in [48]. Moreover, to assure that the UV's controller domains satisfy the waypoint switching feasibility condition (47), a family of N_i robust controllable sets $\{\mathcal{T}_i^n\}_{n=1}^{N_i}$ has been computed for each vehicle, where for

$$i \in \{1,2,3\} \rightarrow N_i = 21 \quad i \in \{4,5\} \rightarrow N_i = 19$$

$$i \in \{6,9,10\} \rightarrow N_i = 15 \quad i \in \{7,8\} \rightarrow N_i = 18$$

By considering the UVs' initial positions as shown in Fig. 12, the obtained simulation results are collected in Figs. 12-14.

In Fig. 12, the UVs' trajectories are depicted for the time interval $[0,\ 100]s$. The trajectories show how the **ST-MPC** controllers are able to track the switching waypoints despite constraints and disturbances. Moreover, it is possible to notice that the obtained paths have potential collision points. Therefore, it is worth investigating how **TM** actions are essential to avoid collisions. To better understand the **TM** *modus operandi*, we shall refer to Fig. 13 where the simulation has been paused at $t = 8.9s$. Specifically, at the considered screenshot, the UVs' set-membership

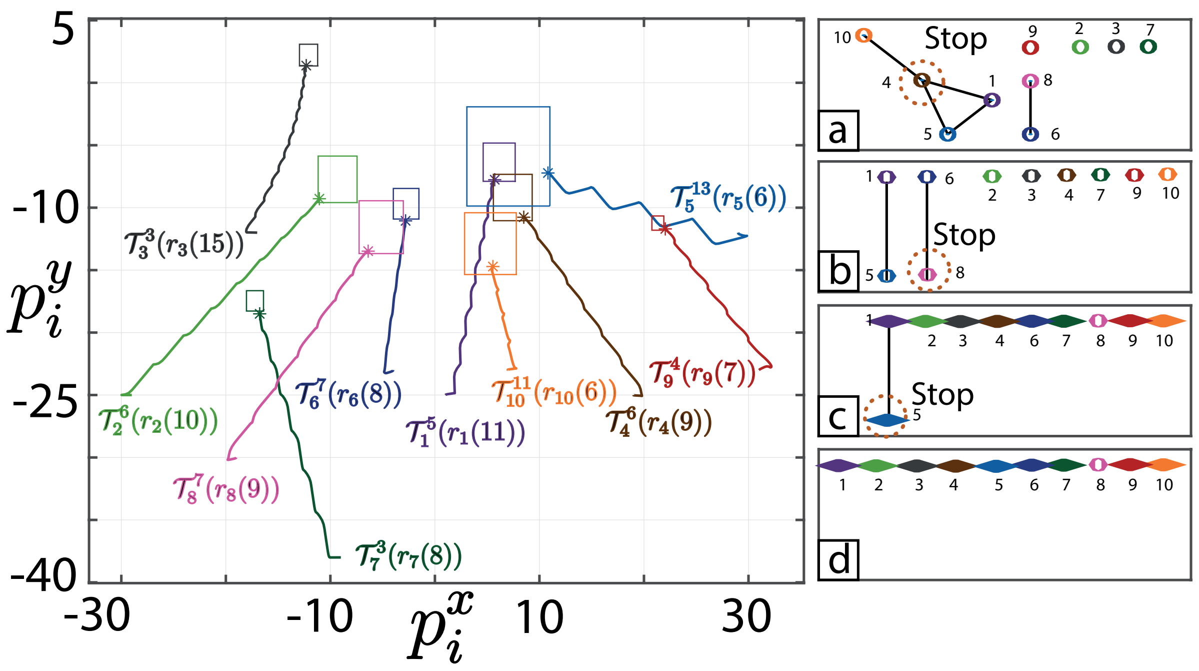

Figure 13: Potential collisions at $t = 8.9s$ and connectivity graphs.

scenario is the following:

$$x_1 \in \mathcal{T}_1^0(r_1(10)),\ x_2 \in \mathcal{T}_2^0(r_2(9)),\ x_3 \in \mathcal{T}_3^0(r_3(14)),\ x_4 \in \mathcal{T}_4^0(r_4(8)),\ x_5 \in \mathcal{T}_5^0(r_5(5))$$
$$x_6 \in \mathcal{T}_6^8(r_6(8)),\ x_7 \in \mathcal{T}_7^4(r_7(8)),\ \ x_8 \in \mathcal{T}_8^0(r_8(8)),\ x_9 \in \mathcal{T}_9^5(r_9(7)),\ x_{10} \in \mathcal{T}_{10}^0(r_{10}(5)).$$

TM first collects all the waypoint switch requests and set-membership indices. At $t = 8.9s$, the set of UVs in a terminal region making a waypoint switch request is $\mathcal{I}^{sw} = \{1, 2, 3, 4, 5, 8, 9\}$. Given the collected information, the **TM** builds the connectivity graph $\mathcal{G}(8.9)$ according to (51).

In Fig. 13, the current UVs' positions are shown with a star symbol, and the rectangular areas (matched by color) represent the regions where the one-step evolution (at $t = 9s$) of each agent will be confined. By construction, according to the waypoint switch feasibility condition (47), the vehicles in the terminal region also belong to the family of the one-step controllable set associated with the successive waypoint. In particular, at $t = 8.9s$:

$$x_1 \in \mathcal{T}_1^5(r_1(11)),\ x_2 \in \mathcal{T}_2^6(r_2(10)),\ x_3 \in \mathcal{T}_3^3(r_3(15))$$
$$x_4 \in \mathcal{T}_4^6(r_4(9)),\ x_5 \in \mathcal{T}_5^{13}(r_5(6)),\ x_8 \in \mathcal{T}_8^7(r_5(9))$$
$$x_{10} \in \mathcal{T}_{10}^{11}(r_{10}(6))$$

Since the constructed families of one-step controllable sets are nested, in Fig. 13, we show only the outer sets. The connectivity graph in Fig. 13.a summarizes all the possible collisions (50), at $t = 8.9s$. In particular, the graph presents potential collisions among the vehicles $1-4$, $4-5$, $1-5$, $4-10$ and $6-8$. Therefore, according to the **TM** algorithm, the (C1)-(C3) procedure is activated to avoid collisions by stopping the minimum number of vehicles among the ones making a waypoint switch request. In the first iteration, the UV_4 (the node with the highest degree) is stopped and added to $\mathcal{I}^{stop}$. As a consequence, all the edges connected to UV_4 are also removed. The resulting connectivity graph and remaining intersections are shown in Fig. 13.b. Since collisions between the vehicles, $1-4$ and $6-8$ are still possible. The second iteration of (C1)-(C3) is executed and the vehicle UV_8 is added to $\mathcal{I}^{stop}$. Fig. 13.c. shows the resulting graph with a single collision possibility between UV_1 and UV_5. Therefore, since both vehicles have the same connectivity degree either UV_1 or UV_5 could be stopped. In the simulation, UV_5 is added to $\mathcal{I}^{stop}$, i.e. $\mathcal{I}^{stop} = \{4, 5, 8\}$. The completely disconnected graph in Fig. 13.d results where no collisions are possible. Therefore, the **TM** operations are concluded: UV_4, UV_5 and UV_8 are stopped while UV_1, UV_2, UV_3, and UV_{10} are allowed to switch waypoint.

In Fig. 14, the vehicles set-membership index signal is shown for UV_3, UV_5 and UV_9 in the time interval $[0, 20]s$ (the time interval has been shortened to improve the figure's readability). According to the **TM** operations previously described, such a signal allows us to better clarify the STOP and GO commands received by these vehicles. In the absence of collisions, the signals $n_i(t)$, by construction, have a reverse sawtooth shape. The wave ramps downward while the vehicles move within the family of computed one-step controllable sets and sharply rises when a waypoint switch occurs. On the other hand, when the signal $n_i(t)$ holds constant for more

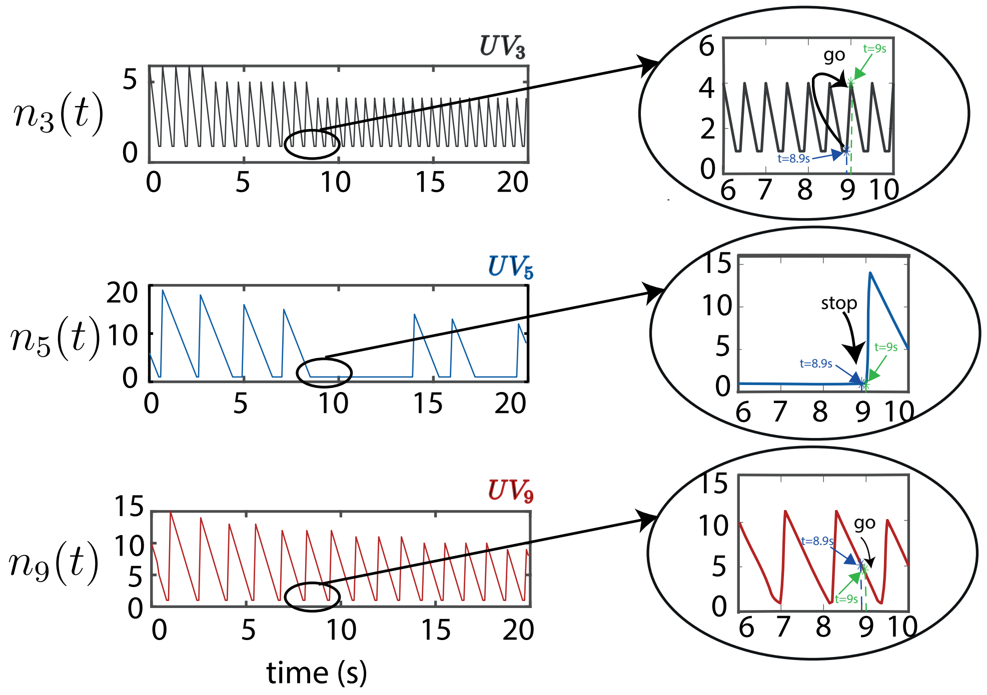

Figure 14: UVs' set-membership indices in the time interval $[0 - 20]s$.

than one sampling time, it means that the vehicle i has received a STOP command. It is worth noticing in Fig. 14 that, according to the developed theory, a STOP signal can be received only by the vehicles making switch request, i.e. from the vehicles within a terminal region ($n_i(t) = 0$). As an example, in the previously described potential collision happening at $t = 8.9s$, the TM imposes a STOP on three of the seven vehicles making a switch request, i.e. UV_4, UV_5 and UV_8. As a consequence, in the zoom-in subplot in Fig. 14, it is possible to appreciate what follows: the signal $n_5(8.9)$ stays constant to zero, meaning that the waypoint switch has been denied for UV_5; the index $n_3(8.9)$ jumps from 0 to $n_3(9) = 11$, testifying that the waypoint switch has been granted to UV_3; the signal $n_9(8.9)$ keeps decreasing showing that UV_9 moves closer to the current waypoint (vehicles that do not belong to the terminal region will not be stopped).

Finally, for the interested reader, the demo of the performed simulation for 20 vehicles is available at the following weblink: `https://tinyurl.com/y23gzeu5`.

4.3.2 Scenario Two

The UVs' dynamics are described by (36) where $T_s = 0.1\,\mathrm{sec}$ and subject to the disturbance sets

$$d_1(t) \in \mathcal{D}_1 = \{d \in \mathbb{R}^4 : |d(s)| \leq 7 \times 10^{-4},\ s = 1, \ldots, 4\}$$
$$d_2(t) \in \mathcal{D}_2 = \{d \in \mathbb{R}^4 : |d(s)| \leq 9 \times 10^{-4},\ s = 1, \ldots, 4\}$$

The following state and acceleration constraints are prescribed

$$\mathcal{V} = \{v \in \mathbb{R}^2 : |v(s)| \leq 5,\ s = 1,\, 2\}$$
$$u_1(t) \in \mathcal{U}_1 = \{u \in \mathbb{R}^2 : |u(s)| \leq 8,\ s = 1,\, 2\}$$
$$u_2(t) \in \mathcal{U}_2 = \{u \in \mathbb{R}^2 : |u(s)| \leq 7,\ s = 1,\, 2\}$$

where s denotes the $s - th$ component of each vector. And, for the TM operations, $L = 3$ different velocity constraints levels (45) are considered, i.e. $V = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3\}$:

$$\mathcal{V}_1 = \{v \in \mathbb{R}^2 : |v_i(s)| \leq 0.8,\ s = 1, 2\}$$
$$\mathcal{V}_2 = \{v \in \mathbb{R}^2 : |v_i(s)| \leq 2,\ s = 1, 2\},\ \ \mathcal{V}_3 \equiv \mathcal{V}$$

The vehicles start from the initial positions $p_1(0) = [10,\ -25.5]^T$ and $p_2(0) = [-10.5,\ -20]^T$ while the planner provides the UVs' waypoints according to the discrete function (see Fig. 15)

$$\begin{bmatrix} r_1(k)^T \\ r_2(k)^T \end{bmatrix} = \begin{bmatrix} 10\sin(0.3k + 14) & 2k \\ 11\sin(0.2k - 1) & 2k \end{bmatrix}$$

which prescribe that the maximum distances δ_i between two successive waypoints are $\delta_1 = 1.76$ and $\delta_2 = 1.85$.

According to the **ST-MPC** algorithm, and *Proposition* 3 prescriptions, first a common terminal RCI region is computes as in [48]. Then, offline, for each velocity level $1 \leq l \leq 3$ a family

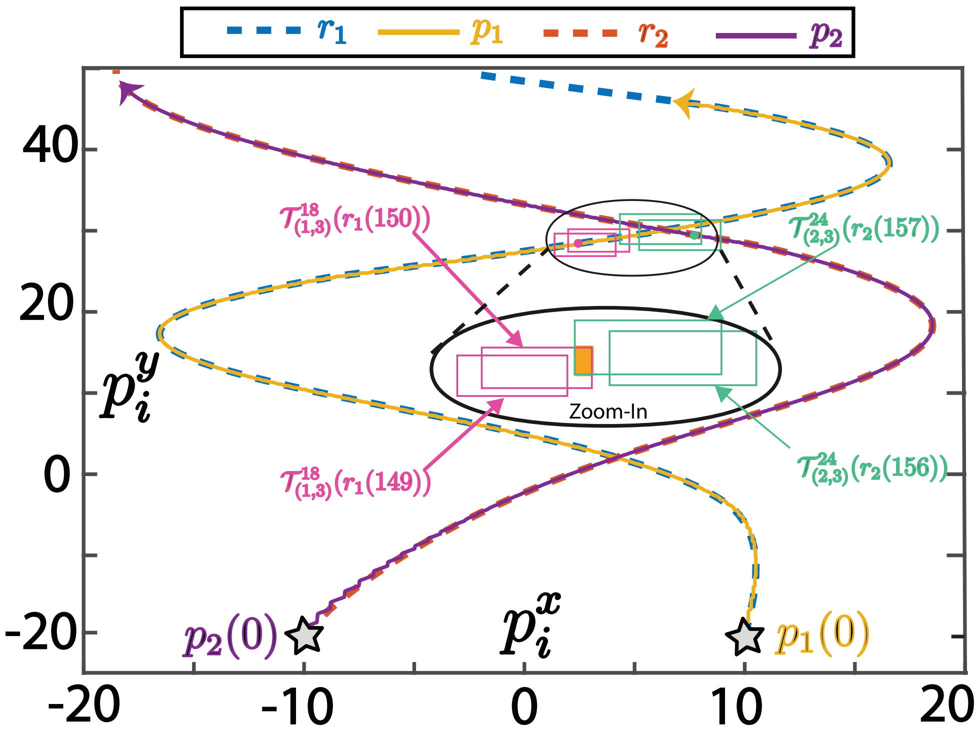

Figure 15: Vehicles' trajectories for $t \in [0,\ 130]s$

of robust one-step controllable set has been determined under the feasibility condition (c).

$$\{\mathcal{T}^n_{(1,1)}\}^{10}_{n=0},\ \{\mathcal{T}^n_{(1,2)}\}^{14}_{n=0},\ \{\mathcal{T}^n_{(1,3)}\}^{18}_{n=0}$$
$$\{\mathcal{T}^n_{(2,1)}\}^{6}_{n=0},\ \{\mathcal{T}^n_{(2,2)}\}^{12}_{n=0},\ \{\mathcal{T}^n_{(2,3)}\}^{24}_{n=0}$$

The obtained simulation results are collected in Figs. 15-16.

By considering a time interval $[0,\ 130]s$ and a waypoint prediction horizon $H_i = 15$, Fig. 15 shows how the UVs' local ST-MPC controllers are able to track the switching waypoints despite constraints and disturbances. To better understand the **TM** *modus operandi*, we shall refer to the zoom-in in Fig. 15, where it shows a simulation screenshot at $t = 85.5$. At the considered time, both UVs are in their terminal regions, i.e.

$$\{\mathcal{T}^0_{(1,3)}(r_1(149)\},\ \{\mathcal{T}^0_{(2,3)}(r_2(156)\}$$

and would like to move to the successive waypoint, namely $r_1(150)$ and $r_2(157)$, i.e. $\mathcal{I}^{sw} =$

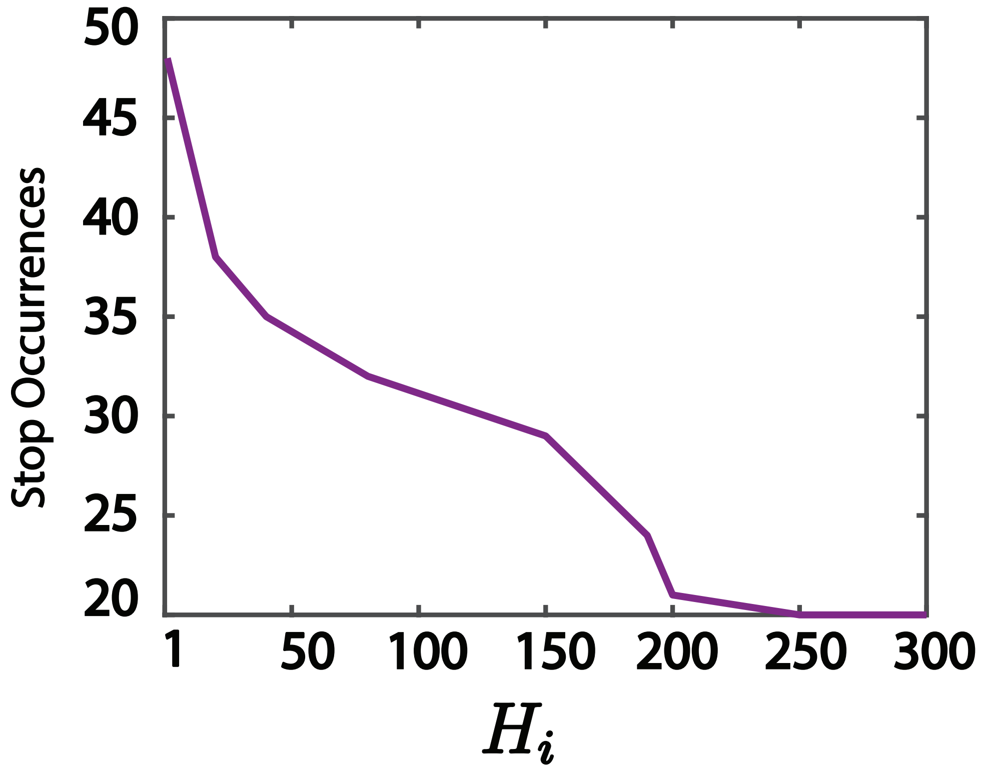

Figure 16: Number of stop occurrences for different prediction horizons.

$\{1, 2\}$. Since the two families or robust one-step controllable set centered in the new waypoint have a non-empty intersection (see orange region), i.e.

$$\{\mathcal{T}^n_{(1,3)}(r_1(150)\}^{18}_{n=0} \bigcap \{\mathcal{T}^n_{(2,3)}(r_2(157)\}^{24}_{n=0} \neq 0$$

the **TM** detects a possibility of collision in the current waypoint, see (51). As a consequence, the (C1)-(C3) procedure is activated and the vehicle UV_1 is stopped, i.e. $\mathcal{I}^{stop} = 1$ and the maximum velocity constraint $\mathcal{V}_3$ is assigned to UV_2 (see *remark* 12).

To show the **TM**'s capabilities to reduce the number of vehicles stop exploiting the waypoint prediction horizon, a simulation campaign involving different horizons, $1 \leq H_i \leq 300$, has been conducted. In Fig.16, the total number of stops (sum of the stops requested to vehicle 1 and 2) in the function of the prediction horizon is reported. It is possible to appreciate that if the waypoint prediction horizon increases, then the total stop time decreases. The latter finds justification because by increasing H_i, the **TM** can anticipate future collisions better and impose velocities

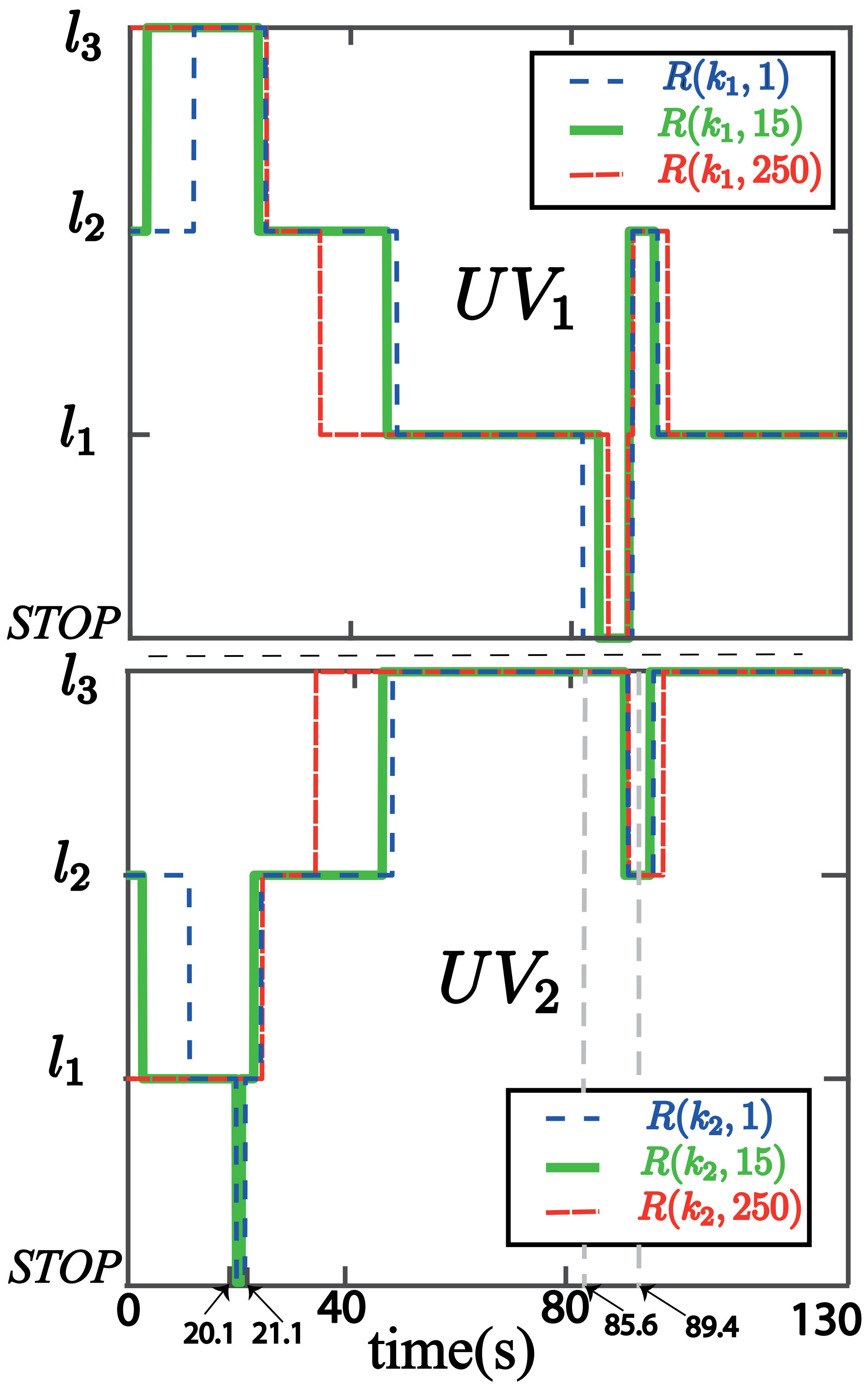

Figure 17: Velocity constraints and stops imposed by the ***TM***.

constraints to minimize their actual occurrence.

To better explain this aspect we shall refer to Fig. 17, where the velocity constraint level l_i and stops assigned to each vehicles are reported for $H_i = 1$ (blue dashed line) $H_i = 15$ (green solid line), and $H_i = 250$ (red dashed line). By referring to the time interval $[20.1, 21.1]$, it is possible to notice that UV_2 is stopped for 10 time instances if $H_2 = 1$, for 8 time instances if $H_2 = 15$, and 0 times if $H_2 = 250$. The latter finds justification in the fact that increasing the prediction horizon the **TM** can earlier impose to UV_2 a lower velocity constraint (e.g. from l_2 to l_1). The same reasoning applies to the time interval $[85.6, 89.4]$ for UV_1.

4.4 Conclusion

In this chapter, we have presented a control architecture to deal with the collision avoidance problem for heterogeneous constrained vehicles moving in a shared environment. In the proposed solution, each vehicle is equipped with a local MPC tracking controller, and a centralized traffic manager solves, in a receding horizon fashion, the collision avoidance problem. In particular, by resorting to set-theoretic arguments and to time-varying velocity constraints, the proposed traffic manager is capable of predicting future potential collisions within a prediction horizon and minimize their occurrence. Moreover, whenever necessary, the traffic manager can impose, in agreement with the tracking controller, a vehicle stop in the robust control invariant region of the current waypoint, so ensuring the absence of collisions. A simulation example in two different scenarios has been used to clarify the capabilities of the proposed solution.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis has extended the ST-MPC paradigm to deal with two different control problems: battery shortage prevention for EUV and collision avoidance strategy for constrained MUVS.

For the battery shortage prevention problem (Chapter. 3), we have designed a novel and simple battery manager for energy shortage prevention. The following two key aspects have been taken into consideration: *(i)* given the EUV's dynamical model and the path to reach the desired reference (goal location), the EUV's behavior is fully described by the underlying tracking controller actions; (ii) the battery discharge rate is approximately proportional to the vehicle's acceleration, see, e.g. [39, 57]. Such considerations are used to design a battery energy shortage avoidance system, which imposes acceleration constraints on the vehicle. The latter is here achieved by exploiting a receding horizon MPC framework known as set-theoretic MPC, see, e.g. [33, 34] and references therein. In particular, starting from the tracking controller developed in [48], we have coupled the design of the tracking controller and battery manager by taking advantage of key features of the set-theoretic paradigm. The main advantages of the proposed energy shortage prevention system can be summarized as follow: *(i)* Given the vehicle's dynamical model, the path to follow, and the initial battery's State-of-Charge (SoC), namely $SoC(0)$, a feasible conservative acceleration profile, assuring energy shortage prevention can be offline defined (if it exists), by solving an integer linear programming problem. *(ii)* In a receding horizon fashion, at each sampling time, given the current $SoC(t)$, the energy shortage avoidance problem can be online re-solved to reduce the offline solution's conservativeness (with respect to a given cost function) and improve the control performance. *(iii)* Different energy profiles can be defined (3 different performance criteria are defined in this paper), and the end-user can arbitrary online switch among them without affecting the controller's recursive feasibility or causing energy shortage.

For the collision avoidance problem (Chapter. 4), we have improved the solution in [48] by proposing a novel collision avoidance strategy that aims to minimize the number of vehicles' stops required to avoid collisions. To this end, assuming that each vehicle follows a sequence of waypoints provided by a local planner (not coordinated among the vehicles), first, the vehicles' tracking controllers have been designed to deal with time-varying velocity constraints. Then, exploiting in a receding horizon fashion a preview of the future vehicle waypoints (waypoint prediction horizon), the traffic manager's collision avoidance algorithm has been enhanced to predict collisions over the available horizon. Finally, given the vehicles' distance from the nearest collision point, and by exploiting the UVs' local controller capabilities, the vehicles' velocity constraints have appropriately been tuned to minimize the change that more than one vehicle reaches the collision points at the same time. It has been formally proved that the overall strategy's recursive feasibility and absence of collisions are guaranteed regardless of each UVs' reference trajectories. The obtained simulation results considering two different scenarios have been presented to show the proposed solution's capability and main features.

5.2 Future Work

Some suggestions and future research in this area are outlined below:

- The battery management solution proposed in Chapter. 3 can be extended to deal with a more complete (nonlinear) model of a EUV or to take into account the collision avoidance problem in a multi-electric vehicles scenario.

- The traffic management solution in Chapter. 4 can be extended to deal with UVs with different priorities (e.g. ambulances, taxi, regular passenger car). Moreover, the proposed traffic manager logic, currently based on the measured distances to the collision points, can be enhanced to take into account more sophisticated collision risk indices taking into account the time to collisions and priorities.

References

[1] Y. Cao, W. Yu, W. Ren, and G. Chen, “An overview of recent progress in the study of distributed multi-agent coordination,” *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2013.

[2] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[3] C. M. Martinez, X. Hu, D. Cao, E. Velenis, B. Gao, and M. Wellers, “Energy management in plug-in hybrid electric vehicles: Recent progress and a connected vehicles perspective,” *IEEE Trans. on Vehicular Technology*, vol. 66, no. 6, pp. 4534–4549, 2016.

[4] B. Sakhdari and N. Azad, “An optimal energy management system for battery electric vehicles,” *IFAC-PapersOnLine*, vol. 48, no. 15, pp. 86–92, 2015.

[5] B. Paden, M. Čáp, S. Z. Yong, and E. Yershov, D.and Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[6] J. Wit, C. D. Crane III, and D. Armstrong, “Autonomous ground vehicle path tracking,” *Journal of Robotic Systems*, vol. 21, no. 8, pp. 439–449, 2004.

[7] A. P. Aguiar and J. P. Hespanha, “Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty,” *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1362–1379, 2007.

[8] G. Dongbing and H. Huosheng, “Receding horizon tracking control of wheeled mobile robots,” *IEEE Trans. Control Syst. Technol*, vol. 14, no. 4, pp. 743–749, 2006.

[9] G. Franzè and W. Lucia, “A receding horizon control strategy for autonomous vehicles in dynamic environments,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 695–702, 2015.

[10] X. Dong, B. Yu, Z. Shi, and Y. Zhong, “Time-varying formation control for unmanned aerial vehicles: Theories and applications,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 340–348, 2015.

[11] K. L. Fetzer, S. Nersesov, and H. Ashrafiuon, “Sliding mode control of underactuated vehicles in three-dimensional space,” in *American Control Conference (ACC)*, 2018, pp. 5344–5349.

[12] M. Egerstedt and X. Hu, “Formation constrained multi-agent control,” *IEEE transactions on robotics and automation*, vol. 17, no. 6, pp. 947–951, 2001.

[13] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, “Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments,” *The International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

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

[15] J. Minguez, F. Lamiraux, and J.-P. Laumond, “Motion planning and obstacle avoidance,” in *Springer handbook of robotics*, 2008, pp. 827–852.

[16] Y. Kodama and K. Nakatani, “Collision avoidance system,” Nov. 27 2018, uS Patent App. 10/140,867.

[17] S. G. Wirasingha and A. Emadi, “Classification and review of control strategies for plug-in hybrid electric vehicles,” *IEEE Transactions on vehicular technology*, vol. 60, no. 1, pp. 111–122, 2010.

[18] B. Geng, J. K. Mills, and D. Sun, “Two-stage energy management control of fuel cell plug-in hybrid electric vehicles considering fuel cell longevity,” *IEEE Trans. on vehicular technology*, vol. 61, no. 2, pp. 498–508, 2011.

[19] S. Cheng, L. Li, H.-Q. Guo, Z.-G. Chen, and P. Song, “Longitudinal collision avoidance and lateral stability adaptive control system based on mpc of autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[20] L. Li, Y. Lu, R. Wang, and J. Chen, “A three-dimensional dynamics control framework of vehicle lateral stability and rollover prevention via active braking with mpc,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3389–3401, 2016.

[21] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, “Predictive active steering control for autonomous vehicle systems,” *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007.

[22] J. Mattingley, Y. Wang, and S. BOYD, “Automatic generation of high-speed solvers,” *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 52–65, 2011.

[23] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.

[24] C. E. Garcia, D. M. Prett, and M. Morari, “Model predictive control: theory and practice—a survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[25] M. Morari and J. H. Lee, “Model predictive control: past, present and future,” *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[26] J. B. Rawlings, “Tutorial overview of model predictive control,” *IEEE control systems magazine*, vol. 20, no. 3, pp. 38–52, 2000.

[27] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. Springer, 1999, pp. 207–226.

[28] P. J. Campo and M. Morari, "Robust model predictive control," in *American control conference (ACC)*, 1987, pp. 1021–1026.

[29] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[30] E. C. Kerrigan and J. M. Maciejowski, "Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 14, no. 4, pp. 395–413, 2004.

[31] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear model predictive control*. Springer, 2009, pp. 345–369.

[32] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on control systems technology*, vol. 18, no. 2, pp. 267–278, 2009.

[33] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008.

[34] D. Angeli, A. Casavola, G. Franzè, and E. Mosca, "An ellipsoidal off-line mpc scheme for uncertain polytopic discrete-time systems," *Automatica*, vol. 44, no. 12, pp. 3113–3119, 2008.

[35] W. Lucia, D. Famularo, and G. Franzè, "A set-theoretic reconfiguration feedback control scheme against simultaneous stuck actuators," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2558–2565, 2017.

[36] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones, "On real-time robust model predictive control," *Automatica*, vol. 50, no. 3, pp. 683–694, 2014.

[37] S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management," *IEEE Trans. on Control Systems Technology*, vol. 22, no. 3, pp. 1018–1031, 2013.

[38] Q. Gong, Y. Li, and Z.-R. Peng, "Trip-based optimal power management of plug-in hybrid electric vehicles," *IEEE Trans. on vehicular technology*, vol. 57, no. 6, pp. 3393–3401, 2008.

[39] F. Aymen and C. Mahmoudi, "A novel energy optimization approach for electrical vehicles in a smart city," *Energies*, vol. 12, no. 5, p. 929, 2019.

[40] B. Geng, J. K. Mills, and D. Sun, "Predictive control for plug-in microturbine powered hybrid electric vehicles using telemetry information," in *Int. Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 1468–1473.

[41] E. J. Rodríguez-Seda, D. M. Stipanović, and M. W. Spong, "Guaranteed collision avoidance for autonomous systems with acceleration constraints and sensing uncertainties," *Journal of Optimization Theory and Applications*, vol. 168, no. 3, pp. 1014–1038, 2016.

[42] X. Wang, M. Kloetzer, C. Mahulea, and M. Silva, "Collision avoidance of mobile robots by using initial time delays," in *IEEE Conference on Decision and Control (CDC)*, 2015, pp. 324–329.

[43] M. Chen, J. C. Shih, and C. J. Tomlin, "Multi-vehicle collision avoidance via hamilton-jacobi reachability and mixed integer programming," in *IEEE Conference on Decision and Control (CDC)*, 2016, pp. 1695–1700.

[44] W. B. Dunbar and D. S. Caveney, "Distributed receding horizon control of vehicle platoons: Stability and string stability," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 620–633, 2012.

[45] H. Fukushima, K. Kon, and F. Matsuno, “Model predictive formation control using branch-and-bound compatible with collision avoidance problems,” *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1308–1317, 2013.

[46] A. Richards and J. How, “Decentralized model predictive control of cooperating uavs,” in *IEEE Conference on Decision and Control (CDC)*, vol. 4, 2004, pp. 4286–4291.

[47] K.-D. Kim and P. R. Kumar, “An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic.” *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3341–3356, 2014.

[48] M. Bagherzadeh and W. Lucia, “Multi-vehicle reference tracking with guaranteed collision avoidance,” in *European Control Conference (ECC)*, 2019, pp. 1574–1579.

[49] D. P. Bertsekas and I. B. Rhodes, “On the minimax reachability of target sets and target tubes,” *Automatica*, vol. 7, no. 2, pp. 233–247, 1971.

[50] S. Savehshemshaki and W. Lucia, “A receding horizon battery shortage prevention control strategy for electric unmanned vehicles,” in *2020 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2020, pp. 108–113.

[51] S. Savehshemshaki and W. Lucia, “A receding-horizon collision avoidance strategy for constrained multi unmanned vehicle systems,” *European Control Conference (ECC) (Under review)*.

[52] M. Bagherzadeh, S. Savehshemshaki, and W. Lucia, “Guaranteed collision-free reference tracking in constrained multi unmanned vehicle systems,” *IEEE Transactions on Automatic Control (Under review)*.

[53] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[54] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, “Invariant approximations of the minimal robust positively invariant set,” *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 406–410, 2005.

[55] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, “Multi-parametric toolbox 3.0,” *European Control Conference (ECC)*, pp. 502–510, 2013.

[56] A. Kurzhanskiĭ and I. Vályi, *Ellipsoidal calculus for estimation and control.* Nelson Thornes, 1997.

[57] A. Flah and C. Mahmoudi, “Design and analysis of a novel power management approach, applied on a connected vehicle as v2v, v2b/i, and v2n,” *International Journal of Energy Research*, vol. 43, no. 13, pp. 6869–6889, 2019.

[58] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, “A generic battery model for the dynamic simulation of hybrid electric vehicles,” in *IEEE Vehicle Power and Propulsion Conference*. Ieee, 2007, pp. 284–289.

[59] O. Tremblay and L.-A. Dessaint, “Experimental validation of a battery dynamic model for ev applications,” *World electric vehicle journal*, vol. 3, no. 2, pp. 289–298, 2009.

[60] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How, “Robust constrained receding horizon control for trajectory planning,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[61] Z. Lin, B. Francis, and M. Maggiore, “Necessary and sufficient graphical conditions for formation control of unicycles,” *IEEE Transactions on Automatic Control*, vol. 50, no. 1, pp. 121–127, 2005.

[62] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks.* Princeton University Press, 2010, vol. 33.