A walk through constrained optimization-based control problems

Ionela Prodan

A walk through constrained optimization-based control problems

Habilitation thesis

Ionela PRODAN

Université Grenoble Alpes

HABILITATION A DIRIGER LA RECHERCHE

Pour obtenir le diplôme

DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : AUTOMATIQUE - PRODUCTIQUE

Arrêté ministériel : 25 mai 2016

Présentée par

Ionela Prodan

préparée au sein du Laboratoire de Conception et d'Intégration des Systèmes dans l' École Doctorale Électronique, Électrotechnique, Automatique, Traitement du Signal EEATS (EEATS)

A walk through constrained optimization-based control problems

Thèse soutenue publiquement le 7 décembre 2020,
devant le jury composé de :

M. Sebastien Gros
Professeur, Norwegian University of Science and Technology, Rapporteure
M. Hugues Mounier
Professeur, CentraleSupélec, Rapporteure
Mme. Angela Schoellig
Professeur, University of Toronto, Rapporteure
M. Mazen Alamir
Directeur de Recherche CNRS, Institut Polytechnique de Grenoble, Examinateur
M. Laurent Lefèvre
Professeur, Institut Polytechnique de Grenoble, Examinateur
M. Silviu-Iulian Niculescu
Directeur de Recherche CNRS, CentraleSupélec, Examinateur
M. Cristian Oara
Professeur, University Politehnica Bucharest, Examinateur
M. Serge Pierfederici
Professeur, Université de Lorraine, Examinateur

Contents

List of Tables

List of Figures

Summary

This manuscript presents the professional, academic and scientific results obtained during the last eight years of my career (after obtaining the PhD title in Automation in 2012) as well as the research directions that I wish to follow in the medium term (based on open questions identified in my area of interest).

In particular, and as will be discovered throughout the manuscript, I am interested in the application of optimization-based control strategies for complex systems. My research contributions offer a geometric viewpoint of various control problems which appear in motion planning for autonomous (aerial, aquatic) vehicles and interconnected systems like DC microgrids or water networks.

Why optimization-based control? Today's systems have complex architectures, are bounded by heterogeneous constraints and have to respect challenging operational costs. Thus, optimization-based approaches, which fully exploit changes in *mission*, *condition* and *environment*, are becoming ubiquitous within the community.

Why a geometric viewpoint? Most control problems can be better understood if seen from a geometrical viewpoint. Such an interpretation may lead to new theoretical insights and streamline their representation. Furthermore, there are many tools which can be readily applied such as set-based methods and combinatorial results.

The first part of the manuscript gives a brief overview of my professional and academic career up to date (post-thesis). The main stages were the follow-up postdoctoral fellowship carried within the EDF chair, École Centrale Paris and the subsequent employment at Univ. Grenoble Alpes, Grenoble INP, Esisar (École Nationale Supérieure in Advanced Systems and Networks), LCIS (Laboratory for Systems Design and Integration) laboratory where I currently hold the position of Associate Professor. Along this path, I participated in various research and teaching activities which allowed me to develop my capacity for independent work.

The second part of the manuscript lists the main scientific results obtained after the end of the thesis. In particular, I consider the following to be relevant research areas:

i) the application of differential flatness for the resolution of various constrained-optimization problems (with auxiliary tools such as port-Hamiltonian modeling and spline representations);

ii) motion planning via model predictive control, for unmanned surface and aerial vehicles (with particular emphasis on multicopter study and experimental validation);

iii) the study and application of hierarchical control strategies for DC microgrids (power balancing, profile generation and tracking, converter control, etc.).

The third part of the manuscript summarizes the problems that I have identified in recent years and the approaches that I wish to apply. A brief list covers:

i) application of spline approximation theory to the flat-based representations of nonlinear dynamics to reduce complexity;

ii) combining a particular class of models with flat representations (in view of automated model reformulation and subsequent resolution of constrained optimization problems);

iii) extensions of modeling and control for larger scale systems (accounting for the multiple time scales, profile generation, reliability and fault propagation) with learning-based methods;

Ce manuscrit présente les résultats professionnels, académiques et scientifiques obtenus au cours des huit dernières années de ma carrière (après la fin de ma thèse en décembre 2012) ainsi que les axes de recherche que je souhaite suivre à moyen terme (sur la base de questions ouvertes identifiées dans mon domaine d'intérêt).

En particulier, et comme on le découvrira tout au long du manuscrit, je m'intéresse à l'application de stratégies de commande basées sur l'optimisation pour des systèmes complexes. Mes contributions de recherche offrent un point de vue géométrique de divers problèmes de contrôle qui apparaissent dans la planification du mouvement pour les véhicules autonomes (aériens, aquatiques) et les systèmes interconnectés comme les micro-réseaux CC ou les réseaux d'eau.

Pourquoi la commande basée sur l'optimisation? Les systèmes d'aujourd'hui ont des architectures complexes, sont limités par des contraintes hétérogènes et doivent respecter des coûts opérationnels difficiles. Ainsi, les approches basées sur l'optimisation, qui exploitent pleinement les changements de mission, de condition et d'environnement, deviennent omniprésentes au sein de la communauté.

Pourquoi un point de vue géométrique? La plupart des problèmes de contrôle peuvent être mieux compris s'ils sont vus d'un point de vue géométrique. Une telle interprétation peut conduire à de nouvelles perspectives théoriques et simplifier leur représentation. En outre, il existe de nombreux outils qui peuvent être facilement appliqués tels que les méthodes basées sur des ensembles et les résultats combinatoires.

La première partie du manuscrit donne un bref aperçu de ma carrière professionnelle et académique à ce jour (post-thèse). Les principales étapes ont été le suivi d'une formation postdoctorale (chaire EDF, École Centrale Paris) et l'emploi au Grenoble INP, Esisar (École Nationale Supérieure en Systèmes Avancés et Réseaux), laboratoire LCIS (Laboratoire de Conception et d'Intégration des Systèmes) où j'occupe actuellement le poste de Maitre de Conférences. Tout au long de ce parcours, j'ai participé à diverses activités de recherche et d'enseignement qui m'ont permis de développer ma capacité de travail autonome.

La seconde partie du manuscrit énumère les principaux résultats scientifiques obtenus après la fin de la thèse. En particulier, je considère comme directions de travail pertinentes les suivantes:

i) l'application de la platitude différentielle pour la résolution de divers problèmes d'optimisation sous contraintes (avec des outils auxiliaires tels que la modélisation port-hamiltonienne et les représentations splines);

ii) planification du mouvement via la commande prédictive pour les drones aquatiques et aériens (avec un accent particulier sur l'étude multicoptère et la validation expérimentale);

iii) l'étude et l'application de stratégies de commande hiérarchiques pour les micro-réseaux CC (équilibrage de puissance, génération et suivi de profils, contrôle de convertisseur, etc.).

La troisième partie du manuscrit résume les problèmes que j'ai identifiés ces dernières années et les approches que je souhaite appliquer. Une brève énumération couvre:

i) application de la théorie de l'approximation splines aux représentations à base plate de la dynamique non linéaire pour réduire la complexité;

ii) combiner une classe particulière de modèles avec des représentations plates (en vue de la reformulation automatisée de modèles et de la résolution ultérieure de problèmes d'optimisation sous contraints);

iii) extensions de la modélisation et du contrôle pour les systèmes à plus grande échelle (tenant compte des échelles de temps multiples, génération de profil, fiabilité et propagation des défauts) avec des méthodes basées sur l'apprentissage.

Part I.

Professional and academic achievements

1. Academic career

I was born on 13^{th} of July 1985 and I did part of my studies in Romania. I have received the B.E. degree from the University "Politehnica" of Bucharest, Romania in 2009. I continued my studies with a PhD in Control Engineering at Supélec, Gif-sur-Yvette, France (2009 – 2012). For one and a half years I was a postdoctoral researcher within the Chair on Systems Science and the Energetic Challenge - EDF, École Centrale Paris, France (2013 – 2014).

I am currently an Associate Professor at University Grenoble Alpes, Grenoble INP, Esisar (École Nationale Supérieure en Systèmes Avancés et Réseaux) within LCIS (Laboratoire de Conception et d'Intégration des Systèmes) laboratory, France.

Therefore, my academic career spans the last 11 years (Oct. 2009 - Mar. 2020) and the main highlights are enumerated below: diplomas, professional experience, a brief recapitulation of my publications and visibility at the national and international level. Since these last academic years were mainly spent in France, this enumeration will be carried out in French.

Je suis née le 13 juillet 1985 et j'ai les nationalités roumaine et française. J'ai obtenu le diplôme d'ingénieur de l'Univ. "Politehnica" de Bucarest, Roumanie en 2009. J'ai poursuivi mes études avec un doctorat en Automatique à Supélec, Gif-sur-Yvette, France (2009 – 2012) et un post-doctorat au sein de la Chaire Science des Systèmes et Défi Énergétique - EDF, École Centrale Paris, France (2013 – 2014). Je suis actuellement Maître de Conférences à l' Univ. Grenoble Alpes, Grenoble INP, Esisar (École Nationale Supérieure en Systèmes Avancés et Réseaux) au sein du LCIS (Laboratoire de Conception et d'Intégration des Systèmes), Valence, France.

Mes intérêts sont multidisciplinaires et centrés sur le contrôle-commande des systèmes dynamiques ainsi que les mathématiques appliquées. Ils englobent la commande sous contraintes fondée sur l'optimisation (via des approches distribuées et hiérarchiques, la commande prédictive non linéaire et des techniques fondées sur les champs de potentiels), la programmation mixte en nombres entiers, la platitude différentielle. Certains systèmes d'application se concentrent sur des robots autonomes multiples et des systèmes d'énergie micro réseaux.

Google scholar: https://scholar.google.com/citations?user=OSiQW5cAAAAJ&hl=en

Publons: https://publons.com/researcher/1582441/ionela-prodan/

ResearchGate: https://www.researchgate.net/profile/Ionela_Prodan2

Web page: http://ionela-prodan.com/

Web of Science ResearcherID: F-9546-2019

ORCID iD: https://orcid.org/0000-0002-3522-5192

CT CPNL: https://www.ct-cpnl.fr/

Part II.

Scientific results

3. Trajectory design via flatness: autonomous vehicles and DC microgrids applications

This chapter addresses the benefits of differential flatness in constrained trajectory design. We present a novel solution for the satisfaction of continuous-time constraints through the use of B-spline curves. Exploiting the B-splines functions properties (positivity, convexity, local support), necessary and sufficient LMI (Linear Matrix Inequalities)-based conditions for the satisfaction of continuous-time constraints are provided. We consider both inclusion and exclusion constraints (in relation with a predefined convex domain) and implement the latter as a MI-SDP (Mixed integer - Semi-definite Programming) problem. We highlight the approach in motion planning and profile generation in microgrid systems: i) off-line trajectory generation for aquatic drones with communication induced constraints; ii) quadcopter trajectory planning with position and angular constraints; iii) dissipative energy minimization through optimal profile generation for an electro-mechanical elevator system.

The chapter is based on several publications:

C51 **Prodan, I.**, F. Stoican, and C. Louembet. "Necessary and sufficient LMI conditions for constraints satisfaction within a B-spline framework", pp. 8061–8066. In: *Proceedings of the 58th IEEE Conference on Decision and Control (CDC'19)*. 2019. Nice, France.

C44 Hervagault, Y., **I. Prodan**, and L. Lefevre. "Trajectory generation and tracking for aquatic drones with communication constraints guarantees", pp. 96–103. In: *Proceedings of the 26th IEEE European Control Conference (ECC'19)*. 2019. Naples, Italy.

C47 Stoican, F., **I. Prodan**, E. I. Grotli, and N. T. Nguyen. "Inspection Trajectory Planning for 3D Structures under a Mixed-Integer Framework", pp. 132–139. In: *Proceedings of the 15th IEEE International Conference on Control & Automation*. 2019. Edinburgh, Scotland.

J9 Nguyen, N. T., **I. Prodan**, and L. Lefèvre. "Flat trajectory design and tracking with saturation guarantees: a nano-drone application". In: *International Journal of Control*, pp. 1–14, 2018. ISSN: 0020-7179. Taylor & Francis.
DOI: 10.1080/00207179.2018.1502474. EID: 2-s2.0-85052058030

C40 Pham, T., **I. Prodan**, D. Genon-Catalot, and L. Lefevre. "Dissipated energy minimization for an electro-mechanical elevator of a DC microgrid", pp. 17–24. In: *Proceedings of the 25th IEEE European Control Conference (ECC'18)*. 2017. Limassol, Cyprus.

C39 Nguyen, N., **I. Prodan**, and L. Lefevre. "Effective angular constrained trajectory generation for thrust-propelled vehicles", pp. 161–168. In: *Proceedings of the 25th IEEE European Control Conference (ECC'18)*. 2017. Limassol, Cyprus.

3.1. Brief literature review

In many control and optimization problems, constraints are considered at discrete instants of time (continuous dynamics are discretized and their behaviour is only analyzed at the sampling times). Furthermore, non-convex constraints are also commonly encountered. The field of motion planning exhibits both these issues (e.g., stall constraints for aerial drones, collision avoidance constraints in robotic applications (Van Parys and Pipeleers 2017), (Mercy, Van Parys, and Pipeleers 2017)). Tackling them often reduces to computing a feasible trajectory, either off-line **J9** or on-line **J8** implementations.

Therefore, the motion planning problem which respects the internal dynamics of autonomous systems (e.g., USVs (Unmanned Surface Vehicles) or VTOL (Vertical Take-Off and Landing) systems) and states and inputs constraints while optimizing certain objectives (e.g., passing through a priori given way-points, state/input constraints satisfaction, consumption minimization, communication constraints) (Herissé, Hamel, Mahony, and Russotto 2012), (Hehn and D'Andrea 2015), (Engelhardt, Konrad, Schäfer, and Abel 2016), **C30**. This is a common problem in the literature and differential flatness properties with appropriate flat output parametrizations are usually employed (Fliess, Lévine, P. Martin, and Rouchon 1995), (Mounier, S.-I. Niculescu, Cela, and Geamanu 2019). For example, the USVs models are highly nonlinear and underactuated, thus the trajectory planning for such systems taking into account communication constraints is non trivial (Pfingsthorn, Birk, and Bülow 2010).

To take into account operating constraints within the trajectory generation problem, different constrained trajectory parametrizations are considered in the literature (H. Lu, C. Liu, L. Guo, and W.-H. Chen 2017), (Hehn and D'Andrea 2011), (Hehn and D'Andrea 2015), (Engelhardt, Konrad, Schäfer, and Abel 2016), e.g., simple monomials (elementary polynomials), Chebyshev polynomials, Laguerre polynomials (Cowling, Yakimenko, Whidborne, and Cooke 2007), Bézier functions (Bipin, Duggal, and Krishna 2014), B-splines functions (H. Lu, C. Liu, L. Guo, and W.-H. Chen 2017), **C30**, etc. In particular, the angles and angular velocities constraints of the aerial systems are usually imposed based on the knowledge of a predefined yaw trajectory, e.g., zero angle (H. Lu, C. Liu, L. Guo, and W.-H. Chen 2017), (Cowling, Yakimenko, Whidborne, and Cooke 2007), **C32** or a spline with specific degree (Engelhardt, Konrad, Schäfer, and Abel 2016). Therefore, any change in the yaw angle trajectory (for example, designedly modified vehicle direction pointing towards the target for camera applications (Engelhardt, Konrad, Schäfer, and Abel 2016) or changes in the direction due to faulty events **C33**), will obviously affect the validation of the above mentioned constraints.

B-splines, in combination with the differential flatness property (Fliess, Levine, P. Martin, and Rouchon 1999) have been intensively used in the literature to solve (static or mobile) obstacle avoidance problems for mobile robots (Van Parys and Pipeleers 2017), (Mercy, Van Parys, and Pipeleers 2017), (Verbandt et al. 2018), (Louembet and Deaconu 2011), **C27**. For instance, the flat outputs of the system are exploited to convert the planning motion problem (usually an optimal control problem) into a static optimization. In fact, using these flat outputs eliminates the differential constraint (dynamic equation) from the optimal control problem by identically satisfying it. The flat outputs are then parametrized on a B-splines basis.

However, significant issues remain: the aforementioned continuous constraint satisfaction but also issues like guaranteeing obstacle avoidance and ensuring performance (Lyche, Manni, and Speleers 2018; Bekcheva, Mounier, and Greco 2017). Each of the approaches encountered in the state of the art comes with particular weaknesses:

i) **J9**, **C21**, (Suryawan, J. De Doná, and M. Seron 2012) exploit the convexity property to provide a finite number of sufficient constraints. Such approaches are conservative, even if the number of the control points is increased (Van Loock, Pipeleers, and Swevers 2015);
ii) (Suryawan, J. De Doná, and M. Seron 2012) provides conditions based on the locality property that can be used in obstacle avoidance problems **C27** through only sufficient formulations;
iii) Necessary and sufficient conditions developed in (Louembet, Cazaurang, and Zolghadri 2010) are based on the sums of squares property. However, LMI formulations can only account for non-convex constraints through simplifications (Louembet and Deaconu 2011).

Notation. A function f is of class $\mathcal{C}^s$ iff its derivatives $f^{(r)}$ are continuous for $r = 0 \dots s$. $\text{ConvHull}\{v_1, \dots, v_n\} = \{x = \sum_i \alpha_i v_i, \sum_i \alpha_i = 1, \alpha_i \geq 0\}$ denotes the convex hull described by vertices $\{v_i\}$.

3.2. Contributions

One of the novelties of this chapter reside in exploiting simultaneously the convexity, positivity and locality properties of B-splines to give necessary and sufficient conditions for the continuous satisfaction of both inclusion and exclusion constraints:

i) Particularize the coefficients description from (Louembet, Cazaurang, and Zolghadri 2010) for each knot sub-interval (by applying both locality and positivity properties);
ii) The above results expand **C27** to necessary and sufficient formulations which ensure collision avoidance in both the single and multi-obstacle case (by applying both positivity and convexity properties).

Noteworthy, inclusion constraints have a standard LMI formulation, whereas exclusion constraints require binary variables thus leading to a mixed-integer LMI formulation.

We extend this approach for off-line trajectory generation for USVs while taking into account communication constraints:

i) using the properties of differential flatness and B-splines parametrization optimal trajectories are generated for the vehicles to follow. The communication constraints are ensured continuously in time through the use of the B-splines convexity properties;
ii) the trajectory tracking controller of the a priori given references is experimentally tested on the SPYBOAT® drone of CT2MC.

Furthermore, by using the differential flatness properties of quadcopter systems:

i) we avoid any requirements on the yaw angle values and we are still able to provide bounds for the angles and their angular velocities. Moreover, the proposed robust constraints can be used with any flat output parametrization in order to generate an optimal trajectory for the thrust-propelled system.
ii) propose a condition for choosing the angle bounds employed in the control design based on the a priori given trajectory and a condition for ensuring the existence of all the related parameters consisting of the reference trajectory, the control design and the limit of system. Thus, we create an unified design scheme for trajectory generation and tracking with bounded thrust and bounded angles while respecting the physical constraints of the system.
iii) validate the control method through simulation and experimental testing over the nano quadcopter Crazyflie 2.0 platform (Giernacki, Skwierczyński, Witwicki, Wroński, and Kozierski 2017).

Finally, in view of dissipative energy minimization for an electro-mechanical elevator of a DC microgrid we use an off-line reference profile generation through flatness and an on-line tracking control problem using MPC:

i) we provide sufficient conditions for the control points describing the B-splines which guarantee the satisfaction of the constraints at all times;
ii) we consider the rotor angle tracking in the MPC formulation by penalizing its discrepancies in the tracking cost;
iii) we provide simulation and comparison results which validate the proposed improvements.

Other applications like flatness-based hierarchical control for DC microgrid systems will be presented in a forthcoming chapter.

3.3. B-splines parametrizations

Hereinafter we make use of the notation presented in Lyche, Manni, and Speleers 2018 to introduce and characterize B-spline functions. These will prove instrumental in characterizing continuous constraints

and costs which will be subsequently used in constrained optimization problems serving varied control requirements (such as motion planning with collision avoidance guarantees for autonomous systems, optimal power flow in microgrids, etc.). Importantly, using these functions to parametrize a flat output will help reduce a continuous optimization problem into a discrete one which considers a finite number of decisions variables (i.e., the control points associated to the B-spline curve describing the flat output).

3.3.1. Definitions and properties

First, we consider a non-decreasing sequence of time instants[1], the so-called *knot vector*,

$$\boldsymbol{\zeta} = \{\tau_1 \leq \tau_2 \leq \cdots \leq \tau_m\}. \tag{3.1}$$

For a $d \leq m-2$, each non-empty (i.e., with $\tau_\ell \neq \tau_{\ell+d+1}$) sub-sequence $\{\tau_\ell \leq \cdots \leq \tau_{\ell+d+1}\} \subset \boldsymbol{\zeta}$ leads recursively to the ℓ-th B-spline function of degree d:

$$B_{\ell,d,\boldsymbol{\zeta}}(t) = \frac{t-\tau_\ell}{\tau_{\ell+d}-\tau_\ell} B_{\ell,d-1,\boldsymbol{\zeta}}(t) + \frac{\tau_{\ell+d+1}-t}{\tau_{\ell+d+1}-\tau_{\ell+1}} B_{\ell+1,d-1,\boldsymbol{\zeta}}(t), \tag{3.2a}$$

$$B_{\ell,0,\boldsymbol{\zeta}}(t) = \begin{cases} 1, & t \in [\tau_\ell, \tau_{\ell+1}), \\ 0, & \text{otherwise.} \end{cases} \tag{3.2b}$$

For further use we make the notations $\mathbf{B}_{d,\boldsymbol{\zeta},[i,j]} = \begin{bmatrix} B_{i,d,\boldsymbol{\zeta}} & \dots & B_{j,d,\boldsymbol{\zeta}} \end{bmatrix}^\top$ and $\mathbf{B}_{d,\boldsymbol{\zeta}} = \mathbf{B}_{d,\boldsymbol{\zeta},[1,n]}$.

Taking $n > d$ such that $m = n + d + 1$ we generate, recursively from (3.2), n B-spline functions $\{B_{1,d,\boldsymbol{\zeta}}, \dots, B_{n,d,\boldsymbol{\zeta}}\}$, which, together with the control points $\{P_1, \dots, P_n\} \subset \mathbb{R}^p$ give the B-spline curve

$$z(t) = \sum_{i=1}^{n} P_i B_{i,d,\boldsymbol{\zeta}}(t), \ \forall t \in [\tau_{d+1}, \tau_{n+1}]. \tag{3.3}$$

The *B-spline functions* $\{B_{\ell,d,\boldsymbol{\zeta}}(t)\}_{\ell=1\dots n}$ resulting from (3.2) and the *B-spline curve* (3.3) are piecewise polynomials of degree d having the properties of:

P1) Local support, for any $\ell = 1 \dots n$:

$$B_{\ell,d,\boldsymbol{\zeta}}(t) = 0, \forall t \notin [\tau_\ell, \tau_{\ell+d+1}]; \tag{3.4}$$

P2) Local partition of unity, for any $\ell = d+1 \dots n$:

$$\sum_{i=\ell-d}^{\ell} B_{i,d,\boldsymbol{\zeta}}(t) = 1, \ \forall t \in [\tau_\ell, \tau_{\ell+1}); \tag{3.5}$$

Enumerating (3.5) for all $\ell = d+1 \dots n$ gives the global equivalent: $\sum_{i=1}^{n} B_{i,d,\boldsymbol{\zeta}}(t) = 1, \ \forall t \in [\tau_{d+1}, \tau_{n+1})$.

P3) Local convexity, for any $\ell = d+1 \dots n$:

$$z(t) = \sum_{i=\ell-d}^{\ell} P_i B_{i,d,\boldsymbol{\zeta}}(t), \forall t \in [\tau_\ell, \tau_{\ell+1}); \tag{3.6}$$

Enumerating (3.6) for all $\ell = d+1 \dots n$ gives the global equivalent: $z(t) = \sum_{i=1}^{n} P_i B_{i,d,\boldsymbol{\zeta}}(t), \forall t \in$

[1] We say that τ_ℓ has multiplicity μ_ℓ if it repeats μ_ℓ times in (3.1).

$[\tau_{d+1}, \tau_{n+1})$.

P4) Smoothness: $B_{i,d,\boldsymbol{\zeta}}(\tau_\ell) \in \mathcal{C}^{d-\mu_\ell}$ at $\tau_\ell \in \boldsymbol{\zeta}$ with multiplicity μ_ℓ and $B_{i,d,\boldsymbol{\zeta}}(t) \in \mathcal{C}^\infty$ at any $t \notin \boldsymbol{\zeta}$.

P5) Clamped B-spline curve. Partitioning the interval $[t_0, t_f]$ into p non-empty sub-intervals:

$$t_0 = t_1 < \cdots < t_{p+1} = t_f, \text{ and} \tag{3.7}$$

taking[2] the knot-vector $\boldsymbol{\zeta} = \{\tau_i\}_{i=1\ldots p+2d+1}$ as

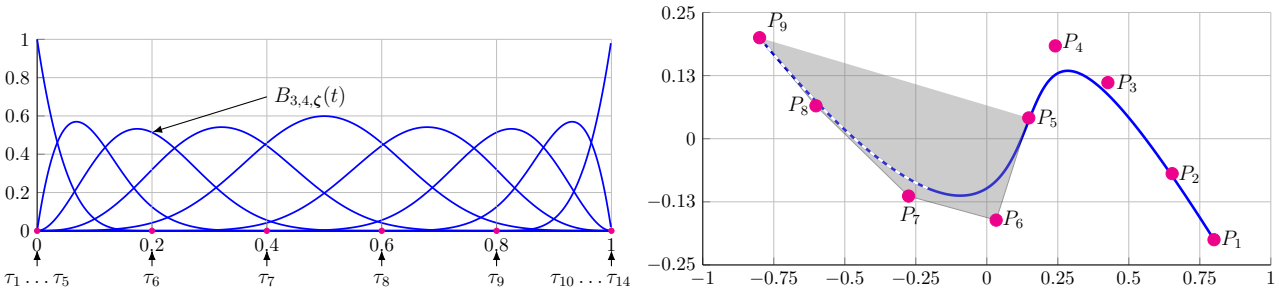
$$\tau_i = \begin{cases} t_1, & 1 \leq i \leq d+1 \\ t_{i-d}, & d+1 < i < p+d+1 \\ t_{p+1}, & p+d+1 \leq i \leq p+2d+1. \end{cases} \tag{3.8}$$

gives a *clamped B-spline* curve where $z(t_0) = P_1$ and $z(t_f) = P_{p+d}$.

We denote with $\mathbb{P}^d_{\boldsymbol{\zeta}}$, the piecewise d-degree polynomial space defined over the knot vector $\boldsymbol{\zeta}$. Then, the B-splines from (3.2) form a basis over the interval $[\tau_{d+1}, \tau_{n+1}]$, thus, any spline from $\mathbb{P}^d_{\boldsymbol{\zeta}}$ can be described by a B-spline curve (3.3) constructed as in the Curry-Schoenberg theorem Lyche, Manni, and Speleers 2018.

Illustrative example

As illustration, we consider the knot-vector (3.1) which partitions the interval $[0, 1]$ into $p = 5$ equidistant sub-intervals and where the first and last knot points have multiplicity $d+1 = 5$. Applying (3.2) for $d = 4$ results the B-spline functions shown in Figure 3.1a.



(a) B-spline basis

(b) B-spline curve and control points

Figure 3.1.: B-spline basis functions and curve

Figure 3.1b shows a B-spline curve (blue line) and its control points (magenta markers). We also illustrate the region (gray-filled) defined by control points $P_5, \ldots, P_9$ to show that it contains the curve on the interval $[\tau_9, \tau_{10})$ (dashed blue line), as guaranteed by **P3)**.

3.3.2. Sufficient and exact validation of continuous constraints

Let us define an obstacle characterized by a finite collection of linear inequalities (a polytope):

$$S = \{z \in \mathbb{R}^q : \ a_k^\top z \geq b_k, \ k = 1 \ldots n_c\} \tag{3.9}$$

[2] To the knot vector (3.8), correspond $n = p + d$ B-spline functions.

where $a_k \in \mathbb{R}^q$, $b_k \in \mathbb{R}$. Defining $z(t)$ as in (3.3) over $\{\nu_\ell(t)\}_{\ell=1\dots n_\nu}$ a B-spline basis function of order d_ν we have that the k-th inequality from (3.9) is equivalent over $[t_s, t_{s+1})$ with:

$$a_k^\top \sum_{\ell=s}^{s+d_\nu} P_\ell \nu_\ell(t) - b_k \geq 0, \tag{3.10a}$$

$$a_k^\top \sum_{\ell=s}^{s+d_\nu} P_\ell \nu_\ell(t) - b_k \sum_{\ell=s}^{s+d_\nu} \nu_\ell(t) \geq 0, \tag{3.10b}$$

$$\sum_{\ell=s}^{s+d_\nu} \left(a_k^\top P_\ell - b_k\right) \nu_\ell(t) \geq 0. \tag{3.10c}$$

(3.10a), (3.10b) are obtained by applying **P3)** and **P2)**, respectively, over the interval $[t_s, t_{s+1})$. (3.10c) gathers the terms in a more compact form which makes clear that the left-side of the k-th constraint is itself a B-spline curve (in the sense of (3.3)) characterized by basis $\{\nu_\ell(t)\}_{\ell=1\dots n_\nu}$ and control points $\{a_k^\top P_\ell - b_k\}_{\ell=1\dots n_\nu}$. For further use we make the notation:

$$\bar{z}_k(t) = \sum_{\ell=1}^{n_\nu} \left(a_k^\top P_\ell - b_k\right) \nu_\ell(t), \ \forall t \in [t_1, t_{n+1}). \tag{3.11}$$

Imposing that $a_k^\top P_\ell - b_k \geq 0, \forall \ell = 1 \dots n_\nu$ means that the curve $\bar{z}_k(t)$ is positive $\forall t \in [t_1, t_{p+1})$ (and thus the inclusion condition $a_k^\top z(t) \geq b_k$ is verified) since the B-spline curve is guaranteed to lie inside the convex hull defined by its control points $a_k^\top P_\ell - b_k$.

Similar discussions may be carried out for exclusion conditions (coming from $z(t) \notin S$) with the caveat that the resulting constraints are disjunctive ("OR" conditions) which requires either a nonlinear solver or the addition of binary variables which in turn leads to mixed-integer formulations.

While through the rest of the chapter I will present mostly results based on the sufficient condition in the rest of this section I will present an LMI-based formulation which guarantees exact constraint validation (i.e., a necessary and sufficient condition). The discussion is based on the sum-of squares link between two families of spline functions. Hence, as a first step, let us consider the interval (3.7) and two B-spline basis of degrees d_ω, d_ν which share, up to the multiplicity of their end-points, the same knot vector ($\boldsymbol{\zeta}_\omega = \{\tau_i^\omega\}_{i=1\dots p+2d_\omega+1}$ and $\boldsymbol{\zeta}_\nu = \{\tau_i^\nu\}_{i=1\dots p+2d_\nu+1}$ defined as in (3.8) with $d \mapsto d_\omega$ and $d \mapsto d_\nu$, respectively):

$$\{\omega_\ell(t)\}_{\ell=1\dots n_\omega} \triangleq \{B_{\ell,d_\omega,\boldsymbol{\zeta}_\omega}(t)\}_{\ell=1\dots p+d_\omega}, \tag{3.12a}$$

$$\{\nu_\ell(t)\}_{\ell=1\dots n_\nu} \triangleq \{B_{\ell,d_\nu,\boldsymbol{\zeta}_\nu}(t)\}_{\ell=1\dots p+d_\nu}. \tag{3.12b}$$

Due to the particularities of the construction (i.e., the choice of $\boldsymbol{\zeta}_\omega, \boldsymbol{\zeta}_\nu$) we have several relationships:

i) the p inner, non-empty sub-intervals coincide:

$$[t_s, t_{s+1}) = [\tau_{d_\omega+s}^\omega, \tau_{d_\omega+s+1}^\omega) = [\tau_{d_\nu+s}^\nu, \tau_{d_\nu+s+1}^\nu), \tag{3.13}$$

with $1 \leq s \leq p$. Consequently, we also have that:

$$[t_1, t_{p+1}) = [\tau_{d_\omega+1}^\omega, \tau_{p+d_\omega+1}^\omega) = [\tau_{d_\nu+1}^\nu, \tau_{p+d_\nu+1}^\nu). \tag{3.14}$$

ii) Applying **P2)** to item (i) we have that to interval $[t_s, t_{s+1})$ correspond $\boldsymbol{\omega}_{[s,s+d_\omega]}$ and $\boldsymbol{\nu}_{[s,s+d_\nu]}$.

In the following we define two operators (which exploit the locality property).

Definition 3.1. *For (3.12a)–(3.12b) we define the operator* $\Lambda^s(\boldsymbol{\nu}_{[s,s+d_\nu]}) : \mathbb{P}^{d_\nu+1} \mapsto \mathbb{P}^{(d_\omega+1)\times(d_\omega+1)}$*:*

$$\Lambda^s(\boldsymbol{\nu}_{[s,s+d_\nu]}(t)) = \boldsymbol{\omega}_{[s,s+d_\omega]}(t)\boldsymbol{\omega}_{[s,s+d_\omega]}(t)^\top = \begin{bmatrix} \lambda_{s,s}^{s,\top}\boldsymbol{\nu}_{[s,s+d_\nu]}(t) & \dots & \lambda_{s,s+d_\omega}^{s,\top}\boldsymbol{\nu}_{[s,s+d_\nu]}(t) \\ \vdots & \ddots & \vdots \\ \lambda_{s+d_\omega,s}^{s,\top}\boldsymbol{\nu}_{[s,s+d_\nu]}(t) & \dots & \lambda_{s+d_\omega,s+d_\omega}^{s,\top}\boldsymbol{\nu}_{[s,s+d_\nu]}(t) \end{bmatrix}, \tag{3.15}$$

with $\{\lambda_{i,j}^s\}_{s\le i,j\le s+d_\omega} \in \mathbb{R}^{d_\nu+1}$ *and* $t \in [t_s, t_{s+1})$. □

Definition 3.2. *For a given operator* $\Lambda^s(\boldsymbol{\nu}_{[s,s+d_\nu]}(t))$ *with* $t \in [\tau_s, \tau_{s+1})$ *there exists the dual operator* $\Lambda^{\star,s}(\mathbf{Y}^s) : \mathbb{R}^{(d_\omega+1)\times(d_\omega+1)} \mapsto \mathbb{R}^{d_\nu+1}$ *defined such that*

$$\left[\Lambda^{\star,s}(\mathbf{Y}^s)\right]^\top \boldsymbol{\nu}_{[s,s+d_\nu]}(t) = \left\langle \mathbf{Y}^s, \Lambda^s(\boldsymbol{\nu}_{[s,s+d_\nu]}(t))\right\rangle, \tag{3.16}$$

holds and is given by

$$\Lambda_\ell^{\star,s}(\mathbf{Y}^s) = \sum_{i,j=1}^{d_\omega+1} Y_{i,j}^s \lambda_{i+s-1,j+s-1,\ell}^s, \tag{3.17}$$

for $1 \le \ell \le d_\nu + 1$. □

We can now provide the exact conditions checking inclusion $z(t) \in S, \forall t \in [t_1, t_{p+1})$:

Proposition 3.1. *For* $\Lambda_\ell^{\star,s,k}(\mathbf{Y}^{s,k})$ *given as in*[3] *(3.16),* S *given as in (3.9),* $z(t)$ *defined as in (3.3) over basis (3.12b), inclusion*

$$z(t) \in S,\ \forall t \in [t_1, t_{p+1}) \tag{3.18}$$

holds for any collection $\{P_\ell\}_{\ell=1\dots n_\nu}$ *which verifies that*

$$\Lambda_\ell^{\star,s,k}(\mathbf{Y}^{s,k}) = a_k^\top P_{\ell+s-1} - b_k,\ \ell = 1 \dots d_\nu + 1, \tag{3.19a}$$

$$\mathbf{Y}^{s,k} \succeq 0, \tag{3.19b}$$

holds for all $1 \le s \le p$ *and* $1 \le k \le n_c$. □

A similar reasoning is applied to ensure the exclusion $z(t) \notin S, \forall t \in [t_1, t_{p+1})$ condition:

Proposition 3.2. *For* $\Lambda_\ell^{\star,s,k}(\mathbf{Y}^{s,k})$ *given as in (3.16),* S *given as in (3.9),* $z(t)$ *defined as in (3.3) over basis (3.12b), exclusion*

$$z(t) \notin S,\ \forall t \in [t_1, t_{p+1}) \tag{3.20}$$

holds for any collection $\{P_\ell\}_{\ell=1\dots n_\nu}$ *which verifies that*

$$\left|\Lambda_\ell^{\star,s,k}(\mathbf{Y}^{s,k}) + \left(a_k^\top P_{\ell+s-1} - b_k\right)\right| \le M\left(1 - \alpha^{s,k}\right),\ \ell = 1 \dots d_\nu + 1, \tag{3.21a}$$

$$\mathbf{Y}^{s,k} \succeq 0, \tag{3.21b}$$

$$\sum_{k=1}^{n_c} \alpha^{s,k} \ge 1, \tag{3.21c}$$

holds for all $1 \le s \le p$ *and* $1 \le k \le n_c$. $\alpha^{s,k} \in \{0,1\}$ *and* $M > 0$ *is a suitably large scalar.* □

The reasoning from Proposition 3.2 is extended easily to the multi-obstacle case. We consider a collection of N hyperplanes $a_k^\top z = b_k$ with $(a_k, b_k) \in \mathbb{R}^q \times \mathbb{R}$. Each of these cuts the space and divides it into two halves (the '+' half, $a_k^\top z \le b_k$ and the '−' half, $-a_k^\top z \le -b_k$). The result is a so-called

[3] We introduce the additional index k in Def. 3.2 and associated variables to account for the multiple constraints considered in (3.9).

hyperplane arrangement Stanley et al. 2004 composed from cells $\mathcal{A}(\sigma_m)$, each defined by a unique sign combination $\sigma_m = (\sigma_1^m, \dots \sigma_N^m) \in \Sigma \subset \{-,+\}^N$:

$$\mathcal{A}(\sigma_m) = \{z : \ \sigma_m^k \cdot a_k^\top z \leq \sigma_m^k \cdot b_k, \forall k = 1 \dots N\}. \tag{3.22}$$

Partitioning the space $\mathbb{R}^q$ becomes a matter of labelling the sign tuples into **[M1]**:

i) interdicted combinations $\sigma_m^\bullet \in \Sigma^\bullet$ characterizing a union of obstacles: $\mathbb{S} = \bigcup_m S_m = \bigcup_m \mathcal{A}(\sigma_m^\bullet)$;

ii) allowed combinations $\sigma_m^\circ \in \Sigma^\circ \triangleq \Sigma \setminus \Sigma^\bullet$ characterizing the feasible space: $\mathbb{R}^p \setminus \mathbb{S} = \bigcup_m \mathcal{A}(\sigma_m^\circ)$.

With these notations we provide the following result.

Corollary 3.1. *For $\Lambda_\ell^{\star,s,k}(\mathbf{Y}^{s,k})$ given as in (3.16), $\mathbb{S}$, $z(t)$ defined as in (3.3) over basis (3.12b), exclusion*

$$z(t) \notin \mathbb{S}, \ \forall t \in [t_1, t_{p+1}) \tag{3.23}$$

holds for any collection $\{P_\ell\}_{\ell=1\dots n_\nu}$ which verifies that

$$\left|\Lambda_\ell^{\star,s,k}(\mathbf{Y}^{s,k}) + \left(a_k^\top P_{\ell+s-1} - b_k\right)\right| \leq M\left(1 - \alpha^{s,k}\right), \tag{3.24a}$$

$$\left|\Lambda_\ell^{\star,s,k}(\mathbf{Y}^{s,k}) - \left(a_k^\top P_{\ell+s-1} - b_k\right)\right| \leq M\alpha^{s,k}, \ \ell = 1 \dots d_\nu + 1, \tag{3.24b}$$

$$\mathbf{Y}^{s,k} \succeq 0, \tag{3.24c}$$

$$\sum_{k:\, \sigma_m^{\bullet,k} = \text{`}-\text{'}} 1 - \alpha^{s,k} + \sum_{k:\, \sigma_m^{\bullet,k} = \text{`}+\text{'}} \alpha^{s,k} \geq 1, \ \forall \sigma_m^\bullet \in \Sigma^\bullet, \tag{3.24d}$$

holds for all $1 \leq s \leq p$ and $1 \leq k \leq N$. $\alpha^{s,k} \in \{0,1\}$ and $M > 0$ is a suitably large scalar. □

Illustrative example

Using the above results and notations we present hereinafter some illustrative examples for the exclusion constraints guarantees (an obstacle avoidance example).

We partition the interval $[0,1]$ into $p = 10$ equidistant sub-intervals, as in (3.7), and use it to construct the B-spline basis (3.12a)–(3.12b) characterized by orders $d_\nu = 4, d_\omega = 2$, knot vectors $\boldsymbol{\zeta}_\nu$, $\boldsymbol{\zeta}_\omega$ given as in (3.8) and with $n_\nu = 14, n_\omega = 12$ elements, respectively. We consider the obstacle S taken from Louembet, Cazaurang, and Zolghadri 2010 and defined as in (3.9), with $n_c = 4$ and a collection of way-points:

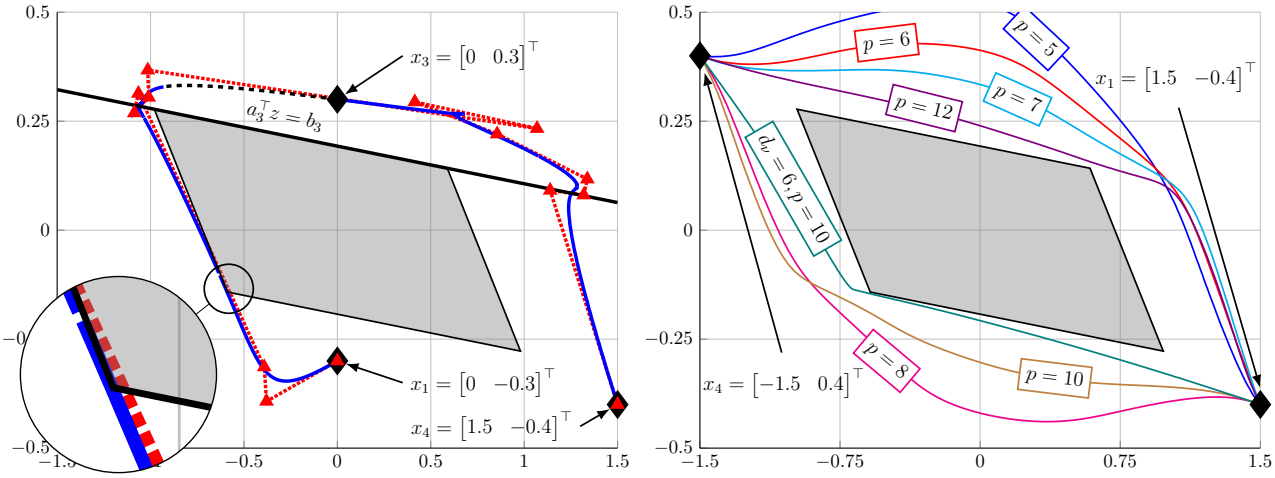
$$S = \left\{ z : \begin{bmatrix} -0.086 & -0.996 \\ 0.731 & 0.683 \\ 0.086 & 0.996 \\ -0.731 & -0.683 \end{bmatrix} z \leq \begin{bmatrix} 0.192 \\ 0.527 \\ 0.192 \\ 0.527 \end{bmatrix} \right\}, \quad x_{1,2,3,4} = \left\{ \begin{bmatrix} 0 \\ -0.3 \end{bmatrix}, \begin{bmatrix} -1.5 \\ 0.4 \end{bmatrix}, \begin{bmatrix} 0 \\ 0.3 \end{bmatrix}, \begin{bmatrix} 1.5 \\ -0.4 \end{bmatrix} \right\}.$$

We define an optimization problem who's goal is to avoid the obstacle and pass through the way-points $x_{1,2,3}$ at times $t_1 = 0$, $t_5 = 0.5$ and $t_{p+1=11} = 1$) while minimizing the energy along the path:

$$\begin{aligned} (P_1^* \dots P_{n_\nu}^*) = \arg \min_{P_1 \dots P_{n_\nu}} & \int_{t_1}^{t_{n+1}} ||z'(t)||^2 dt \\ z(t) \notin S, \ \forall t \in [t_1, t_{n+1}], \quad & z(t_1) = x_1, z(t_5) = x_3, z(t_{n+1}) = x_4. \end{aligned}$$

The obstacle avoidance constraints are implemented as in Prop. 3.2 and the result is illustrated in Figure 3.2a. The way-points $x_{1,3,4}$ are depicted as black, diamond-shaped markers. The $n_\nu = 14$ control points are depicted as red, triangle-shaped markers (the piecewise dotted line they define is

the control polygon of the B-spline curve). The curve is shown as a solid blue curve (except the segment corresponding to the interval $[t_5, t_6] = [0.5, 0.6]$, depicted as a black dashed curve).



(a) Obstacle avoidance by the B-spline curve

(b) B-spline curves with varying degree and basis size

Figure 3.2.: Obstacle avoidance (exact case and variations of order and number of splines)

As expected, the trajectory does indeed avoid intersections with the obstacle (the gray-filled area). The inset from the lower-left corner shows that the control polygon cuts the obstacle and hence, a sufficient method like the ones discussed earlier would not be able to provide this trajectory.

The combination of active constraints may change at each sub-interval $[t_s, t_{s+1})$, $s = 1 \ldots 10$. This is illustrated in Table 3.1 where we observe that the active constraint(s) (the one(s) which ensures the obstacle avoidance, i.e., the k-th inequality is active on the s-th interval iff $\alpha^{s,k} = 0$) may change with each sub-interval. For illustration, in Figure 3.2a we show that during interval $[t_5, t_6)$ the curve lies opposite the obstacle wrt hyperplane $a_3^\top z = b_3$ (i.e., $\alpha^{s=5,k=3} = 0$).

s	1	2	3	4	5	6	7	8	9	10
$\alpha^{s,1}$	1	1	1	1	1	1	1	1	1	1
$\alpha^{s,2}$	0	0	1	1	1	1	1	1	1	1
$\alpha^{s,3}$	1	0	0	0	0	0	0	1	1	1
$\alpha^{s,4}$	1	1	1	1	1	1	1	0	0	0

Table 3.1.: Binary variables denoting the active constraint

To highlight the role of the B-spline basis (3.12b) parameters we solve the path-minimizing problem (this time with $z(t_1) = x_2$ and $z(t_{p+1}) = x_4$) for various combinations of parameters: $p = 4, 5, 6, 7, 8, 9, 10, 12$ for $d_\nu = 4$ and $p = 10$ for $d_\nu = 6$. The resulting trajectories are illustrated in Figure 3.2b. Table 3.2 shows the path lengths (case $p = 4$ is infeasible, there is no trajectory to measure). As expected, increasing the number of sub-intervals p and/or the smoothness of the curve (by increasing its degree d_ν) reduces the path length). The influence of p appears to taper off after $p = 10$ which is again to be expected since the number of constraints ($n_c = 4$) becomes significantly less than the number p of sub-intervals where a change in the combination of active constraints can be done.

p	4	5	6	7	8	9	10	12	10 with $d_\nu = 7$
length	*	3.2889	3.2556	3.2420	3.2403	3.2373	3.2131	3.2183	3.1588

Table 3.2.: Path lengths for various parameters

3.4. Communication constraints for a multi USV team

One of the typical requirements encountered in a multi-agent system is to ensure communication at all times such that information can pass unimpeded among agents and/or reach a ground-based station. This section considers a multi-USV (unmanned surface vehicle) system where the nonlinear dynamics are modeled through a flat-based approach and where the B-spline parametrization of the flat output helps in formulating and enforcing the communication-induced constraints.

3.4.1. Flat-based description of the USV dynamics

The vehicle under test is the SPYBOAT® Goose of CT2MC (France). It is an unmanned vessel of 80 cm length and 47 cm width, designed to fit in a standard car and be manipulated by only one person.

$$\begin{cases} \dot{\boldsymbol{\eta}} & = R(\psi)\boldsymbol{\nu}, \\ M\dot{\boldsymbol{\nu}} & +C(\boldsymbol{\nu})\boldsymbol{\nu} + D(\boldsymbol{\nu})\boldsymbol{\nu} + g(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_E, \end{cases} \tag{3.25}$$

The elements from (3.25) are not repeated here and may be found, e.g., in **C44**. Classical assumptions are applied to obtain the following simplified model:

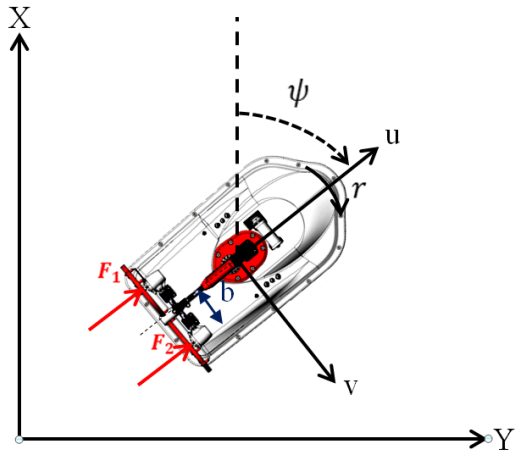


Figure 3.3.: Underactuated USV model coordinates [4]

$$\dot{x} = u\cos\psi - v\sin\psi, \tag{3.26a}$$
$$\dot{y} = u\sin\psi + v\cos\psi, \tag{3.26b}$$
$$\dot{\psi} = r, \tag{3.26c}$$
$$m\dot{u} = \tau_u + mvr + X_u u, \tag{3.26d}$$
$$m\dot{v} = -mur + Y_v v, \tag{3.26e}$$
$$I_z\dot{r} = \tau_r + N_r r, \tag{3.26f}$$

where we denote by

$$\bar{\mathbf{x}} = [x \quad y \quad \psi \quad u \quad v \quad r]^\top \text{ and } \bar{\mathbf{u}} = [\tau_u \quad \tau_r]^\top, \tag{3.27}$$

the state and input vectors respectively[5], thus allowing to write the compact form:

$$\dot{\bar{\mathbf{x}}} = f(\bar{\mathbf{x}}, \bar{\mathbf{u}}), \tag{3.28}$$

[4] The x and y axis are defined such that ψ has the same orientation as the reading of a standard compass.
[5] Whenever the notations are cumbersome we discard the time dependence of the variables.

where $f(.,.) : \mathbb{R}^6 \times \mathbb{R}^2 \to \mathbb{R}^6$ captures the description of the dynamical system (3.26).

System (3.28) is controllable, hence it admits a flat output representation. Choosing the flat output as the position coordinates in Earth frame, the states and the inputs of the system can be algebraically expressed in terms of $\mathbf{z}(t)$ and a finite number of its higher-order derivatives:

$$\begin{aligned} \bar{\mathbf{x}}(t) &= \phi_0(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t), \mathbf{z}^{(3)}(t)), \\ \bar{\mathbf{u}}(t) &= \phi_1(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t), \mathbf{z}^{(3)}(t)), \mathbf{z}^{(4)}(t)), \end{aligned} \tag{3.29}$$

where $\mathbf{z}(t) = [x(t) \quad y(t)]^\top$ is the flat output.

3.4.2. Multi-USVs trajectory generation with communication constraints

Let us first define the list of waypoints $\mathcal{W} = \{w_1 \dots w_n\}$ through which a leader agent will have to pass. Next, the number of vehicles n_a needed to maintain the communication link for the entire duration of the mission is calculated, by considering the radio range limitations:

$$n_a = \max_j \left\lceil \frac{\|w_j - \mathbf{p}_g\|}{\rho} \right\rceil, \forall j \in \{1 \dots n\}, \tag{3.30}$$

where $\mathbf{p}_g$ is the position of the ground station, ρ is the radio range and w_j is the j-th waypoint.

These mission requirements are written into an optimization problem whose goal is to:

$$\text{Minimize the total course length} \tag{3.31}$$

such that:

$$\begin{cases} \text{the leader reaches the waypoints in time,} & \text{(3.32a)} \\ \text{each agent preserves a communication link,} & \text{(3.32b)} \\ \text{each agent respects the maximal surge speed.} & \text{(3.32c)} \end{cases}$$

Remark 3.1. Note that we implicitly take into account the system model and constraints by considering the representation in (3.26), hence providing a feasible trajectory.

The cost defined in (3.31) is represented by the the sum of all the courses lengths.

$$J = \sum_{i=1}^{n_a} \int_{c_1}^{c_n} \left\| \dot{\mathbf{z}}_i(t)^\top \right\| dt. \tag{3.33}$$

In terms of the B-Splines parametrization (3.3), the cost function (3.33) becomes:

$$J = \sum_{i=1}^{n_a} \sum_{j,k} \left([\mathbf{P}_i \mathbf{M}_1]_j \right)^\top \left(\int_{c_0}^{c_n} \mathbf{B}_{j,d-1}(t) \mathbf{B}_{k,d-1}(t) dt \right) ([\mathbf{P}_i \mathbf{M}_1]_k), \tag{3.34}$$

where $[.]_j$ extracts the j-th column of the argument.

To relax (3.32a) we consider a so called "acceptance region" centered in the waypoint and through which the USV has to pass (i.e., the initial equality is relaxed to an inequality). Therefore, we define for each waypoint a regular polytopic region with a circumradius equal to ρ_w:

$$S_j = \{[x \ y]^\top \in \mathbb{R}^2, \ A_w \left([x \ y]^\top - w_j \right) \leqslant b_w, \ A_w \in \mathbb{R}^{(n,2)}, b_w \in \mathbb{R}^n\}, \tag{3.35}$$

Hence, the relaxed waypoint passing constraints $\mathbf{z}_1(c_j) \in S_j$, through the B-Splines parametrization (3.3), become:

$$\mathbf{P}_1 \mathbf{B}_d(c_j) \in S_j. \tag{3.36}$$

Since some waypoints are placed outside the ground station range, communication constraints need to be enforced (3.32b). We consider that the network is fixed (i.e., the links between agents do not change) with a simply-connected tree structure, and constant radio range ρ, for all the agents.

To ensure the radio coverage, we define a regular polytopic region with a circumradius equals to ρ:

$$S_r = \{[x\ y]^\top \in \mathbb{R}^2,\ A_r[x\ y]^\top \leqslant b_r,\ A_r \in \mathbb{R}^{(m,2)},\ b_r \in \mathbb{R}^m\}. \tag{3.37}$$

Proposition 3.3. *With the flat output defined as in (3.3), the communication constraint (3.32b) is verified if the collection of control points $\mathbf{P}_i$ verifies:*

$$\begin{aligned} &((\mathbf{P}_i - \mathbf{P}_{i-1})\mathbf{B}_d(t)) \in S_r, \quad \forall i \in \{2 \dots n_a\} \\ &((\mathbf{P}_{n_a} - \mathbf{p}_g)\mathbf{B}_d(t)) \in S_r. \end{aligned} \tag{3.38}$$

To ensure that the trajectories are feasible for each agent, a constraint as in (3.32c) is applied on the maximal speed. Since the flat output is chosen as the position of the vehicle, it is simple to express the speed in terms of the flat output:

$$\left\| \dot{\mathbf{z}}_i^\top \right\| \leqslant v_{max}, \quad \forall i \in \{1 \dots n_a\}, \tag{3.39}$$

where v_{max} is the maximal available speed. Next, the speed constraint (3.39) is expressed using the B-Splines parametrization.

$$\left\| (\mathbf{P}_i \mathbf{M}_1 \mathbf{B}_{d-1}(t))^\top \right\| \leqslant v_{max}, \quad \forall i \in \{1 \dots n_a\}. \tag{3.40}$$

Applying a maximum speed constraint implies a maximum input constraint equals to $\tau_{max} = -X_u v_{max}$.

To summarize, the optimization problem we consider is the following:

$$\min_{\mathbf{P}_i} \sum_{i=1}^{n_a} \sum_{j,k} \left([\mathbf{P}_i \mathbf{M}_1]_j \right)^\top \left(\int_{c_0}^{c_n} \mathbf{B}_{j,d-1}(t) \mathbf{B}_{k,d-1}(t) dt \right) ([\mathbf{P}_i \mathbf{M}_1]_k) \tag{3.41}$$

such that:

$$\begin{cases} \mathbf{P}_1 \mathbf{B}_d(c_j) \in S_j, & \forall j \in \{1 \dots n\} & \text{(3.42a)} \\ ((\mathbf{P}_i - \mathbf{P}_{i-1})\mathbf{B}_d(t)) \in S_r, & \forall i \in \{2 \dots n_a\} & \text{(3.42b)} \\ ((\mathbf{P}_{n_a} - \mathbf{p}_g)\mathbf{B}_d(t)) \in S_r, & & \text{(3.42c)} \\ \left\| (\mathbf{P}_i \mathbf{M}_1 \mathbf{B}_{d-1}(t))^\top \right\| \leqslant \dfrac{-\tau_{max}}{X_u}, \forall i \in \{1 \dots n_a\}. & & \text{(3.42d)} \end{cases}$$

3.4.3. Simulation and experimental results for offline trajectory generation

For testing the offline constrained trajectory generation problem in (3.31)-(3.40) we consider the real parameters of a SPYBOAT® Goose vessel obtained through identification experiments. The a priori given list of waypoints to follow is:

$$\mathcal{W} = \{(0,0), (1,-2), (3,-3), (4,-4), (7,-3), (6,3), (4,1), (0,4), (-4,6), (-7,2), (-5,-4)\} \cdot 10^2 m.$$

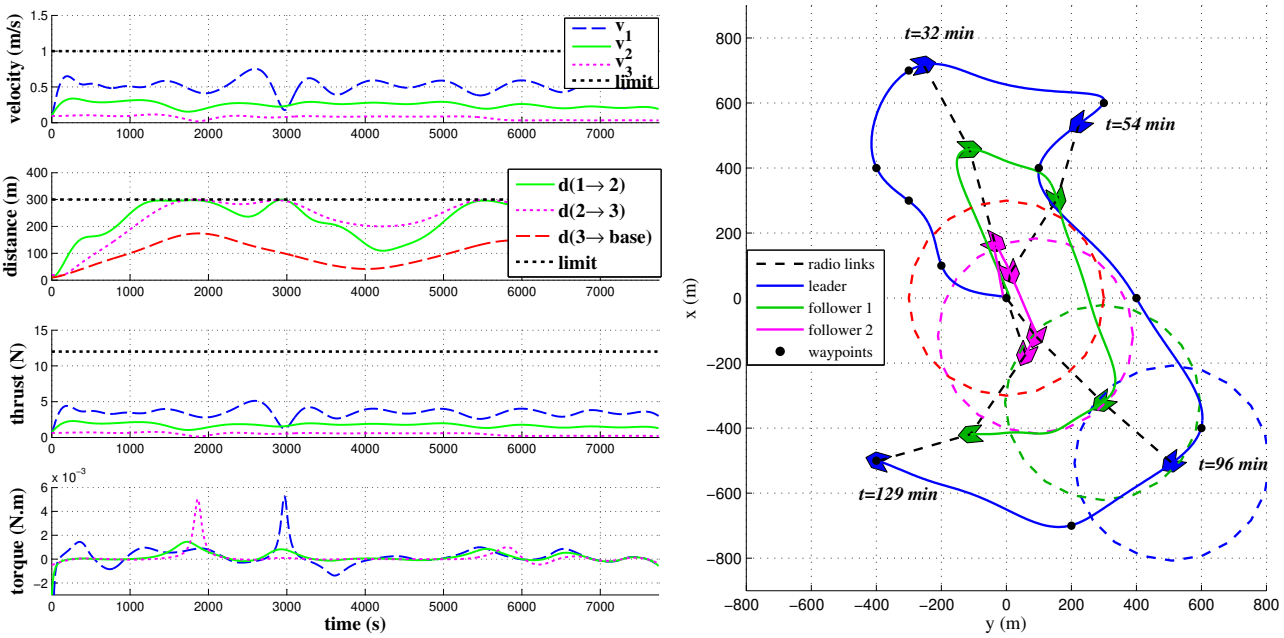
The radio range is $\rho = 300m$ and the radius used to construct A_w and b_w from (3.35) is fixed to $\rho_w = 1m$. The maximal velocity is fixed to $v_{max} = 1m/s$ and we obtain the time instants set $\mathcal{T}$ depending on the distance between each waypoint:

$$\mathcal{T} = \{0, 447, 894, 1177, 1810, 3026, 3592, 4592, 5486, 6486, 7751\}s.$$

As in (3.30), we define $n_a = 3$, the minimum number of USVs necessary to maintain the communication link for the duration of the mission. Next, we proceed with parametrizing the trajectories in

function of the B-splines curves (3.2) and the associated control points (3.36), (3.38) and (3.40).

We choose 20 control points and B-Splines functions of order d=5 to characterize the agents' trajectories. These are illustrated in Fig. 4.3. We show the relative positions of each agent at four time instants to highlight the communications links. The circles plotted at $t = 96\ min$ show the scale of the communication ranges for each vessel. The red circle represents the ground station radio coverage. Furthermore, in Fig. 3.4b we show that the maximal values in terms of speed, thrust forces, torque and communication ranges are within the required limits.



(a) USV trajectories representation in the x/y plane

(b) constraints on states and inputs

Figure 3.4.: 3 USVs with waypoint passing and communication guarantees

Given feasible reference trajectories which ensure communication constraints we attach a classical tracking controller (LQR-based gain scheduling through successive linearizations) and test experimentally the SPYBOAT® Goose USV.

The experimental trials have been carried out on the Bourget lake (France), close to the Charpignat harbor between the 18th and 21st September 2018. For security reasons, the communication range is reduced to 60m and the waypoints are placed in a smaller area, as we want to be able to maintain a clear line-of-sight with the USV and abort the course if other users approach the test region.

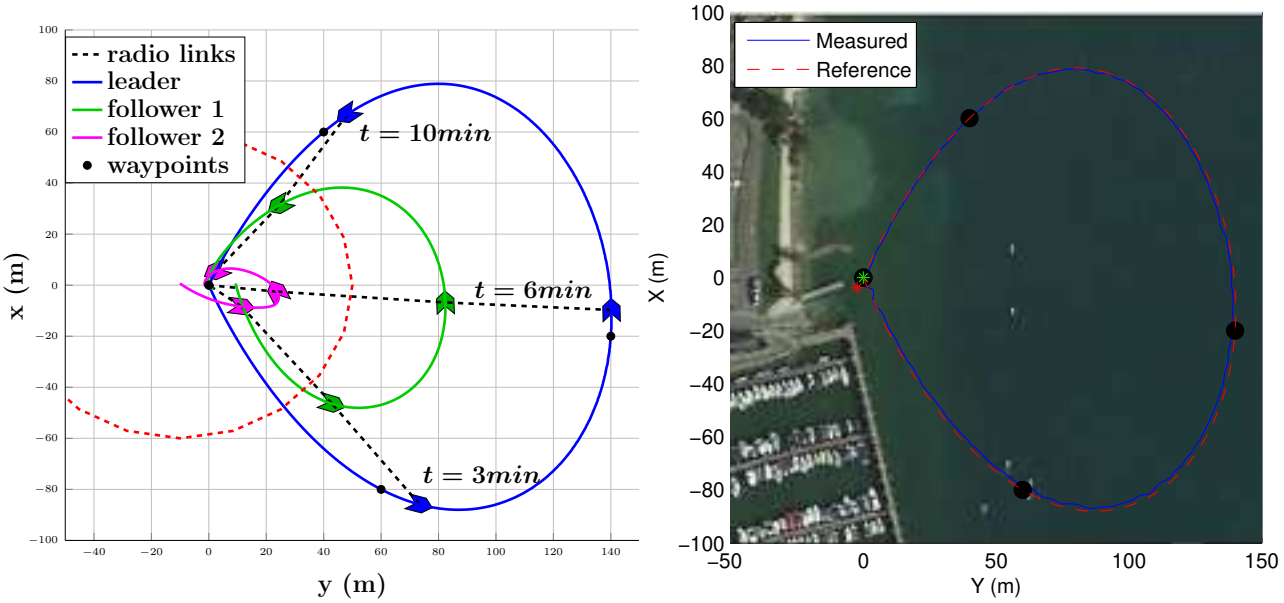
Fig. 3.5a shows the leader and followers trajectories in x/y plane. The leader has to reach five waypoints, the distance of its course is 470m long and lasts 800s:

$$\mathcal{W} = \left\{(0,0)^\top, (-80,60)^\top, (-20,140)^\top, (60,40)^\top, (0,0)^\top\right\} m, \quad \mathcal{T} = \{0, 200, 400, 656, 800\} s.$$

Due to the unavailability of supplementary vehicles, the experiments are done only for the leader, the other agents are represented virtually, with only their reference trajectories. The experimental data (of the leader) shows that a smooth loop trajectory can be achieved by the implemented control law.

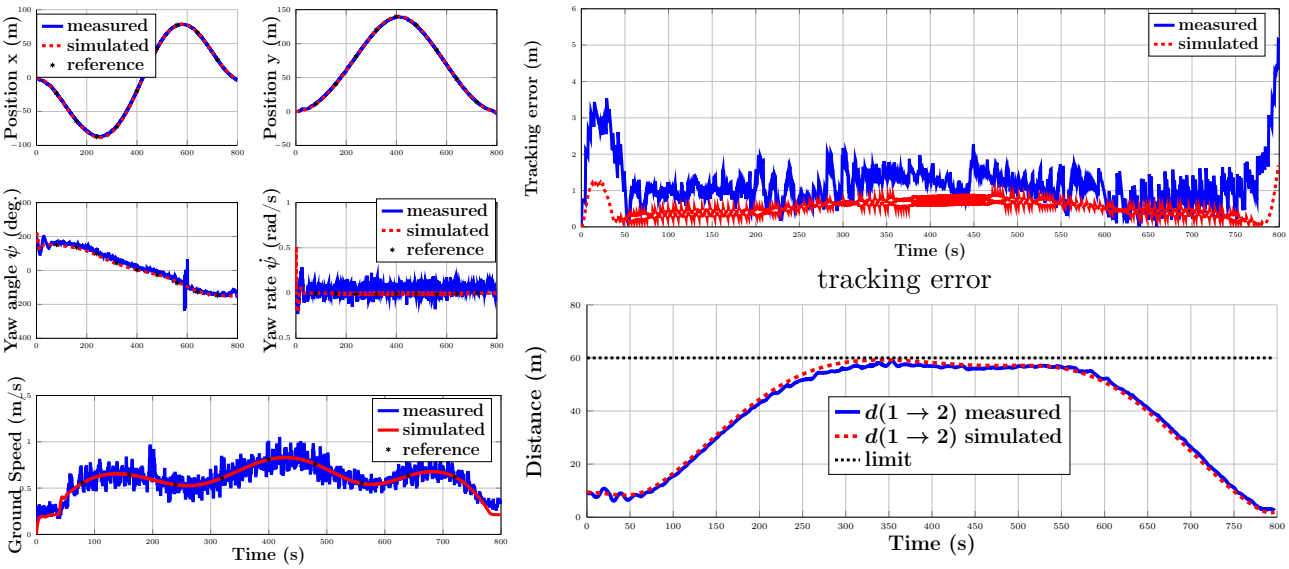
We detail the results obtained for the loop trajectory (Fig. 3.5b) in Fig. 3.6.

The states x and y present a good similarity between the measured and the reference data. The spikes observed in the ψ measurements around 600s are most probably a measuring error. The yaw velocity measurements are quite noisy, both due to the sensors but also due to the influence of the medium (the slightest wave or wind gust changes significantly the value). The simulated and measured tracking error are plotted in Fig. 3.6.



(a) experimental (b) experimental

Figure 3.5.: Trajectories with communication constraints in x/y plane



reference versus experimental values inter-distance between leader and first virtual follower

Figure 3.6.: Experimental trial results

The high error values in the first 50s are due to initial state mismatch. In the last 30s, the error comes most probably from the difficulties to reduce speed and stop over water effectively. Outside these cases, the trajectory tracking stays below 2m, which is acceptable compared to the standard GPS position accuracy (0.5m). Furthermore, this error magnitude is small enough to consider that communication constraints are fulfilled. Lastly, the communication constraint between the leader and the reference trajectory of the first virtual follower (in green in Fig. 3.5a) is verified and displayed in Fig. 3.6.

3.5. Dissipated energy minimization for an elevator integrated in a DC microgrid

Another typical example of nonlinear dynamics which stands to benefit from flat-based representations are DC microgrids. Beyond their nonlinearities, of interest are the constraints which apply for stable functioning. Among these, we were interested in power flow equilibrium conditions; saturation conditions for currents, voltages and powers and minimization of dissipated energy. In this section we consider an electro-mechanical elevator which exhibits all of the aforementioned elements.

3.5.1. Flat-based description of the DC microgrid

The elevator system shown in Figure 3.7 is represented by the combination of the AC/DC converter, the PMSM (Permanent Magnet Synchronous Machine) and the mechanical elevator with the following specifications: linear regime of machine, symmetry in the machine construction, non-friction in mechanical system and constant length of the elevator rope. To explicitly describe the power-preserving

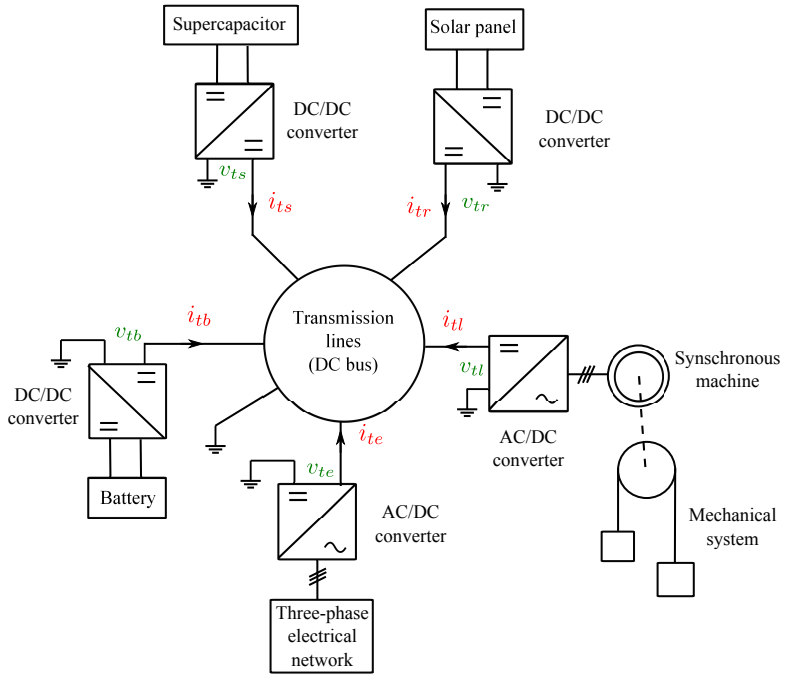


Figure 3.7.: DC microgrid elevator system.

interconnection and the energy conservation of the elevator system, we express its dynamics using the PCH formalism as[6]:

$$\begin{cases} \dot{\mathbf{x}}(t) &= [\mathbf{J}(\mathbf{x}) - \mathbf{R}(\mathbf{x})]\,\nabla H(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{G}^T(\mathbf{x})\nabla H(\mathbf{x}), \end{cases} \tag{3.43}$$

where the state vector, $\mathbf{x}(t) = [x_1(t) \quad x_2(t) \quad x_3(t)]^T \in \mathbb{R}^3$, represents the direct and quadrature stator fluxes and the mechanical momentum. The input vector, $\mathbf{u}(t) = [u_1(t) \quad u_2(t) \quad u_3(t)]^T \in \mathbb{R}^3$, describes the direct and quadrature stator voltages and the mechanical torque caused by the gravity. The output vector, $\mathbf{y}(t) = [y_1(t) \quad y_2(t) \quad y_3(t)]^T \in \mathbb{R}^3$, denotes the direct and quadrature stator currents and the motor speed. The interconnection matrix, $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{3\times3}$, the resistive matrix, $\mathbf{R}(\mathbf{x}) \in \mathbb{R}^{3\times3}$ and the

[6]Note that in **C34** we provide the Port-Hamiltonian-based modelling of the global microgrid system which takes into account all the physical properties of the grid. Since the modelling part is not the main goal of this chapter, we do not enter into details.

input matrix, $\mathbf{G}(\mathbf{x}) \in \mathbb{R}^{3\times 3}$, are given by:

$$\begin{cases} \mathbf{J}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & x_2(t) \\ 0 & 0 & -x_1(t) \\ -x_2(t) & x_1(t) & 0 \end{bmatrix}, \\ \mathbf{R}(\mathbf{x}) = \text{diag}\,\{R_l, R_l, 0\},\ \mathbf{G}(\mathbf{x}) = \mathbf{I}_3, \end{cases} \tag{3.44}$$

where $x_1(t),\ x_2(t) \in \mathbb{R}$ are the 1st and 2nd coordinates of the state vector, $\mathbf{x}(t)$, describing the stator fluxes, R_l is the phase resistance of the machine stator. The Hamiltonian, which has a quadratic form, describes the magnetic energy in PMSM stator and the kinematic energy in the mechanical elevator:

$$H(\mathbf{x}) = \mathbf{Q}_0 + \mathbf{Q}_1^T\mathbf{x}(t) + \frac{1}{2}\mathbf{x}^T(t)\mathbf{Q}_2\mathbf{x}(t), \tag{3.45}$$

where $\mathbf{Q}_0 \in \mathbb{R}$, $\mathbf{Q}_1 \in \mathbb{R}^{3\times 1}$, $\mathbf{Q}_2 \in \mathbb{R}^{3\times 3}$ are the weight matrices.

We consider typical constraints for the elevator system on the motor currents, $y_1(t)$, $y_2(t)$, the motor speed $y_3(t)$, the motor voltages, $u_1(t)$, $u_2(t)$, and the mechanical torque, $u_3(t)$, such as:

$$\mathbf{y}(t) \in \mathbb{G}_y = \left\{ \mathbf{y}(t) \in \mathbb{R}^3 \;\middle|\; \begin{array}{l} y_1^2(t) + y_2^2(t) \le \dfrac{I_{max}^2}{2}, \\ \omega_{min} \le y_3(t) \le \omega_{max} \end{array} \right\},\ \forall t \in [t_0, t_f], \tag{3.46a}$$

$$\mathbf{u}(t) \in \mathbb{G}_u = \left\{ \mathbf{u}(t) \in \mathbb{R}^3 \;\middle|\; \begin{array}{l} u_1^2(t) + u_2^2(t) \le \dfrac{v_{ref}^2}{2}, \\ u_3(t) = \Gamma_{res} \end{array} \right\},\ \forall t \in [t_0, t_f], \tag{3.46b}$$

where: I_{max} is the maximal current magnitude; ω_{min}, ω_{max} are the minimal and maximal rotor speeds; v_{ref} is the DC bus voltage at one side of the converter; Γ_{res} is the gravity torque; t_0, t_f denote the initial and final time instants; and $\mathbb{G}_y, \mathbb{G}_u \subset \mathbb{R}^3$ are convex sets.

The initial and final elevator speeds fulfill the following constraints:

$$y_3(t_0) = y_3(t_f) = 0. \tag{3.47}$$

The rotor angle respects the constraint (θ_f is the required rotor angle during an elevator travel):

$$\theta(t_f) = \theta_f,\ \text{with}\ \ \theta(t) = \int_{t_0}^{t} y_3(t)dt. \tag{3.48}$$

Via (3.43), we derive the dissipated energy within the system during the elevator travel as:

$$V(\mathbf{x}) = \int_{t_0}^{t_f} \nabla^T H(\mathbf{x})\mathbf{R}(\mathbf{x})\nabla H(\mathbf{x})dt \tag{3.49}$$

From the presented dynamics (3.43)-(7.14), the constraints and the cost function, $V(\mathbf{x})$, in (3.49), we formulate the optimal control problem for the elevator system as:

$$\overline{\mathbf{u}}(t) = \underset{\mathbf{u}(t)}{\text{argmin}}\ V(\mathbf{x}), \tag{3.50}$$

$$\text{subject to:} \begin{cases} \text{dynamics (3.43)-(7.14)}, \\ \text{constraints (3.46)-(3.48)}. \end{cases} \tag{3.51}$$

Note that the optimization problem (3.50)-(3.51) is in continuous-time. In **C26**, it is approximated by a combination of off-line reference profile generation and on-line reference tracking using differential flatness with B-splines parameterization and MPC. We proceed in a similar fashion hereafter.

Proposition 3.4. *The dynamical PCH system given in (3.43)-(7.14) is differentially flat with the flat output, $\mathbf{z}(t) \in \mathbb{R}^3$, given by:*

$$\mathbf{z}(t) = \mathbf{y}(t). \tag{3.52}$$

Furthermore, in (3.44)-(7.14), we notice two following characteristics: i) the input matrix, $\mathbf{G}(\mathbf{x})$, is constant and invertible; ii) the Hamiltonian, $H(\mathbf{x})$, is quadratic and positive. Thus, using the definition of the flatness **levine2009**, *we can easily verify that the elevator system (3.43)-(7.14) is flat with the flat output $\mathbf{z}(t)$ in (3.52) which (together with its derivatives) describes the system's states and inputs:*

$$\mathbf{x}(\mathbf{z}(t)) = \mathbf{Q}_2^{-1}\left(\mathbf{z}(t) - \mathbf{Q}_1\right), \tag{3.53a}$$

$$\mathbf{u}(\mathbf{z}(t), \dot{\mathbf{z}}(t)) = \mathbf{A}_1\dot{\mathbf{z}}(t) + \mathbf{A}_2\mathbf{z}(t) + \sum_{k=1}^{3} z_k(t)\mathbf{A}_{3,k}\mathbf{z}(t), \tag{3.53b}$$

where $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_{3,k} \in \mathbb{R}^{3\times 3}$ appropriately chosen.

From (3.43)-(7.14) and (3.52), we obtain the expression of the cost in function of the flat output:

$$V(\mathbf{z}) = \int_{t_0}^{t_f} \mathbf{z}^T(t)\mathbf{R}\mathbf{z}(t)dt, \tag{3.54}$$

3.5.2. Continuous constraint validation through a B-spline parametrization

Taking the B-spline parametrization of $\mathbf{z}(t)$ as in (3.3) and substituting (3.52)-(3.53) in the continuous-time optimization problem (3.50)-(3.51), we obtain a new optimization problem with finite-dimensional arguments, i.e., the control points $\mathbf{P}_i$ characterizing the B-spline curve (3.3).

As a fist step in solving (3.50)-(3.51) to obtain the reference profiles, we rewrite it as

$$\overline{\mathbf{P}} = \underset{\mathbf{P}}{\operatorname{argmin}}\, V(\mathbf{P}) = \underset{\mathbf{P}}{\operatorname{argmin}} \int_{t_0}^{t_f} \left(\sum_{i=1}^{N} \mathbf{P}_i B_{i,d}(t)\right)^T \mathbf{R} \left(\sum_{j=1}^{N} \mathbf{P}_j B_{j,d}(t)\right) dt, \tag{3.55}$$

subject to the constraints (3.51) with formulation (3.53), which we also rewrite as:

$$\sum_{j=1}^{N} \mathbf{P}_j B_{j,d}(t) \in \mathbb{G}_y, \tag{3.56a}$$

$$\mathbf{u}\left(\sum_{j=1}^{N} \mathbf{P}_j B_{j,d}(t), \sum_{j=1}^{N} \mathbf{P}_j \dot{B}_{j,d}(t)\right) \in \mathbb{G}_u, \tag{3.56b}$$

$$\sum_{j=1}^{N} P_{3,j} B_{j,d}(t_0) = \sum_{j=1}^{N} P_{3,j} B_{j,d}(t_f) = 0, \tag{3.56c}$$

$$\int_{t_0}^{t_f} \sum_{j=1}^{N} P_{3,j} B_{j,d}(t) = \theta_f, \tag{3.56d}$$

Reorganizing expression (3.55), we obtain a quadratic cost function:

$$V(\mathbf{P}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{p}_i^T \mathbf{R}_{i,j} \mathbf{p}_j, \text{ with } \mathbf{R}_{i,j} = \int_{t_0}^{t_f} B_{i,d}(t)\mathbf{R}B_{j,d}(t)dt. \tag{3.57}$$

Note that, in (3.46b) and (3.56b), the control points must satisfy an equality constraint with non-

linear terms. This is not easy to implement and leads to a numerically cumbersome formulation. Hereinafter, we rewrite the optimization problem above (3.55)-(3.56) as a soft constrained optimization problem where we penalize in the cost a slack variable ϵ:

$$\overline{\mathbf{P}} = \underset{\mathbf{P},\epsilon}{\operatorname{argmin}}\ V(\mathbf{P}) + Q_\epsilon \epsilon, \tag{3.58a}$$

$$\text{subject to constraints (3.56a), (3.56c)-(3.56d)}, \tag{3.58b}$$

$$\mathbf{u}\left(\sum_{j=1}^{N}\mathbf{P}_j B_{j,d}(t), \sum_{j=1}^{N}\mathbf{P}_j \dot{B}_{j,d}(t)\right) \in \mathbb{G}_{u\epsilon}, \tag{3.58c}$$

$$\epsilon \geq 0, \tag{3.58d}$$

where the set $\mathbb{G}_u$ in (3.46b) is rewritten as $\mathbb{G}_{u\epsilon} = \left\{ \mathbf{u}(t) \in \mathbb{R}^3 \;\middle|\; \begin{array}{l} u_1^2(t) + u_2^2(t) \leq \dfrac{v_{ref}^2}{2}, \\ |u_3(t) - \Gamma_{res}| \leq \epsilon \end{array} \right\}$ where $Q_\epsilon \in \mathbb{R}$ is a positive coefficient, $\epsilon \in \mathbb{R}$ is the relaxation factor. In the following, sufficient conditions for the control points are formulated which guarantee the satisfaction of the system constraints (3.44)-(3.48).

Proposition 3.5. *Let the following conditions be satisfied:*

$$P_{3,1} = P_{3,N} = 0, \tag{3.59a}$$

$$\sum_{j=1}^{N} P_{3,j} \int_{t_0}^{t_f} B_{j,d}(t)dt = \theta_f, \tag{3.59b}$$

$$\mathbf{P}_j \in \mathbb{G}_y, \tag{3.59c}$$

$$\mathbf{P}_{l,i,j} \in \mathbb{G}_{u\epsilon}, \tag{3.59d}$$

with $d-1 \leq l \leq N+d$ the B-spline time interval index $[\tau_{l-1}, \tau_l]$, $l-d+1 \leq i,j \leq l$. Also, $\mathbf{M}_{d,d-1} \in \mathbb{R}^{N\times(N-1)}$ describes the relations between the first derivatives of the B-splines of order d and the B-splines of order $d-1$. $\mathbf{S}_{l,d-1,d,i} \in \mathbb{R}^{N-1}$ describes the relations of the B-splines of order $d-1$ and the i^{th} B-spline of order d over the time interval $[\tau_{l-1}, \tau_l]$. $\mathbb{G}_y$, $\mathbb{G}_{u\epsilon}$ are the convex sets defined in (3.46) and (3.58c). The short notation $\tilde{\mathbf{P}}_{l,i,j} \in \mathbb{R}^3$ is defined as:

$$\tilde{\mathbf{P}}_{l,i,j} = \mathbf{A}_1 \mathbf{P} \mathbf{M}_{d,d-1} \mathbf{S}_{l,d-1,d,i} + \mathbf{A}_2 \mathbf{P}_i + \sum_{k=1}^{3} P_{k,i} \mathbf{A}_{3,k} \mathbf{P}_j. \tag{3.60}$$

The proposition deserves a sketch of its proof. Exploiting the locality property **P1)**we have that, among the N B-splines, only the first B-spline, $B_{1,d}(t)$, is non-zero at the initial instant, t_0, i.e., $B_{1,d}(t_0) = 1$. Similarly, $B_{N,d}(t_f) = 1$. Thus, (3.59a) implies the constraint (3.56c).

Since, (3.3) is linear, we can easily derive the condition (3.59b) for the control points from the angle constraint (3.56d). Using the sufficient inclusion conditions discussed around (3.11), $\mathbf{z}(t)$ is guaranteed to belong to $\mathbb{G}_y$ if its control points also lie inside. Now, consider the time interval $[\tau_{l-1}, \tau_l]$ and let $\beta_{i,j,d}(t) = B_{i,d}(t)B_{j,d}(t)$ with $1 \leq i,j \leq N$. From (3.3) and property **P2)** we obtain:

$$\mathbf{z}(t) = \sum_{i=1}^{N} \mathbf{p}_i B_{i,d}(t) = \sum_{i=1}^{N} \mathbf{p}_i B_{i,d}(t) \sum_{j=1}^{N} B_{j,d}(t) = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{p}_i \beta_{i,j,d}(t). \tag{3.61}$$

Using (3.3) and the derivation properties of B-spline functions we arrive at:

$$\dot{\mathbf{z}}(t) = \sum_{i=1}^{N} \mathbf{P}\mathbf{M}_{d,d-1}\mathbf{S}_{l,d-1,d,i} B_{i,d}(t). \tag{3.62}$$

Multiplying the two sides of (3.62) by $\sum_{j=1}^{N} B_{j,d}(t)$ we rewrite (3.62) as $\dot{\mathbf{z}}(t) = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{PM}_{d,d-1}\mathbf{S}_{l,d-1,d,i}\beta_{i,j,d}(t)$. Substituting (3.3) to the third term of the input, $\mathbf{u}(t)$, in (3.53), we derive:

$$\sum_{k=1}^{3} z_k(t)\mathbf{A}_{3,k}\mathbf{z}(t) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{3} p_{k,i}\mathbf{A}_{3,k}\mathbf{p}_j\beta_{i,j,d}(t). \tag{3.63}$$

Using (3.61) we express the input, $\mathbf{u}(t)$, in (3.53) as $\mathbf{u}(t) = \sum_{i=1}^{N}\sum_{j=1}^{N} \tilde{\mathbf{p}}_{l,i,j}\beta_{i,j,d}(t)$.

Since $\beta_{i,j,d}(t)$ with $l-d+1 \leq i,j \leq l$ also partition the unit, $\mathbf{u}(t)$ remains in the convex hull of $\{p_{l,i,j}\}$ with $l-d+1 \leq i,j \leq l$. Thus, if $\{p_{l,i,j}\}$ with $l-d+1 \leq i,j \leq l$ belong to the convex set $\mathbb{G}_{u\epsilon}$ as in (3.59d), the input $\mathbf{u}(t)$ remains in $\mathbb{G}_{u\epsilon}$. This implies that the continuous-time constraint (3.58c) is satisfied. Consequently, (3.58) are satisfied if (3.59) are satisfied.

3.5.3. Simulation results

Formulating the optimization with the control points as decision variables,

$$\overline{\mathbf{P}} = \underset{\mathbf{P}}{\operatorname{argmin}} \ \left(V_\epsilon(\mathbf{P}) := V(\mathbf{P}) + Q_\epsilon|\epsilon|\right), \text{ subject to constraints (3.59)}, \tag{3.64}$$

with the cost $V_\epsilon(\mathbf{P})$ described in (3.57) allows to generate the reference profiles for the system outputs, $\overline{\mathbf{y}}(t)$, representing the motor currents and the elevator speed, thanks to (3.52) and (3.3). Similarly, we derive the input reference profiles, $\overline{\mathbf{u}}(t)$, representing the motor voltages thanks to (3.52) and (3.3). Lastly, we determine the reference profile of the rotor angle, $\overline{\theta}(t)$, thanks to (3.48), (3.52) and (3.3).

We consider a MPC law for tracking the motor angle reference of the elevator system with the elevator dynamics (3.43) linearized successively. We can now proceed and present simulation and comparison results for the reference profile generation and their tracking. The model parameters, given by the industrial partner SODIMAS (an elevator company from France), may be retrieved from **C40** and are not introduced here. Table 3.3 presents the simulation settings. Two simulation

Table 3.3.: Simulation setting.

Name	Notation	Unit	Value
Time interval	$[t_0, t_f]$	$[s]$	$[0, 30]$
B-spline number	N		10
B-spline order	d		5
Soft constraint weight	Q_ϵ	$[(Nm)^{-1}]$	10^5
Discrete time step	h	$[s]$	0.001
Prediction horizon	N_p		1
Input weight matrix	$\mathbf{Q}_u$		$\mathbf{I}_3$

scenarios are considered for perturbation-affected dynamics with: (a) feedforward control, (b) MPC. These perturbations represent strong, of short duration, interactions between the system and the environment (much shorter than the time step). The perturbation are summarized in Table 3.4. *Reference profile generation:* Table 3.5-3.6 present values of the relaxation factor, ϵ in (3.58a), and

Table 3.4.: Perturbations values for the two scenarios.

Case	Current	Speed	Angle
(a)	9 $[A]$	-1.2 $[rad/s]$	1 rad
(b)	0 $[A]$	-1.2 $[rad/s]$	0 rad

the computation time with different B-splines parametrizations (i.e., by B-spline order and number). The relaxation weight, Q_ϵ (3.58a), is described in Table 3.3. We see that with B-splines of order 5 the soft constraint technique gives the smallest torque error and, consequently, will be used to generate the reference profiles in the following.

Table 3.5.: Tuning for the soft constraint, ϵ in (3.58a).

	N=8	N=9	N=10	N=11	N=12
d=4	0.1216	0.1851	0.0263	0.0533	0.0084
d=5	4×10^{-8}	5×10^{-8}	1×10^{-7}	7×10^{-8}	1×10^{-7}
d=6	3×10^{-5}	1×10^{-5}	7×10^{-5}	28×10^{-5}	19×10^{-5}

Table 3.6.: Computation time in seconds of the off-line reference profile generation.

	N=8	N=9	N=10	N=11	N=12
d=4	6.0800	7.0780	7.9130	8.8400	9.9960
d=5	8.3550	10.0300	11.9450	13.6260	15.2460
d=6	11.0610	13.8170	16.8470	20.5200	23.8510

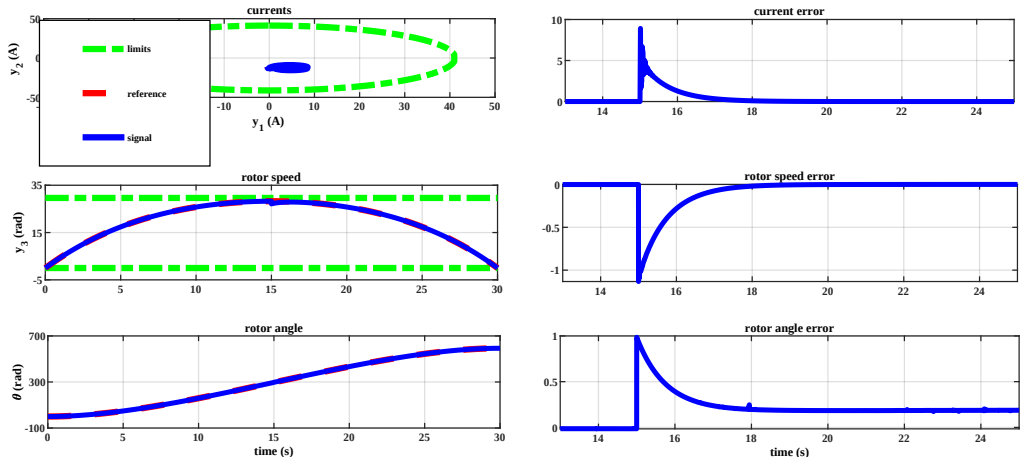


Figure 3.8.: Time evolutions and errors of the output variables and of the rotor angle in case (a).

The reference profiles of the output currents, $\overline{y_1}(t), \overline{y_2}(t)$, the rotor speed, $\overline{y_3}(t)$, and the rotor angle, $\overline{\theta}(t)$, are illustrated in Fig. 3.8 by the dashed lines. We show here that the constraints on the currents, voltages and rotor speed (the dash-dot lines) are respected at all times.

Comparison results: In Table 3.7 we present both the results obtained for a trapezoidal speed profile of the elevator Vittek, Butko, Ftorek, Makys, and Gorel 2017, and our flatness-based approach. Our method respect the elevator dynamics and provide a lower dissipated energy, which is not the case for the MTPA method.

Reference tracking: Next, the feasible references obtained above are tracked using MPC. The generated control references are first applied to the perturbation-affected system. The obtained motor currents, speed and angle are illustrated by the solid lines in Fig. 3.8. The dynamics of the two currents and the speed are asymptotically stable around the corresponding reference profiles, as shown empirically by the convergences to zero of their tracking errors. This is not the case for the rotor's angle tracking error.

To handle the rotor's angle tracking error we have to account for the speed limitation constraint. This problem is especially important at around $t = 15\ s$ when the motor speed and the voltages are

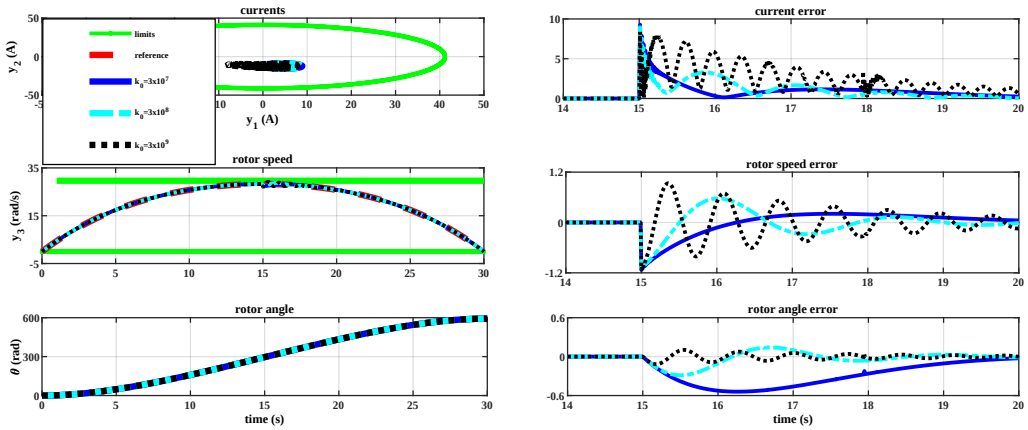
Table 3.7.: Nominal dissipated energy.

Case	Flatness-based approach	Trapezoidal speed profile
Energy [J]	2647	2709

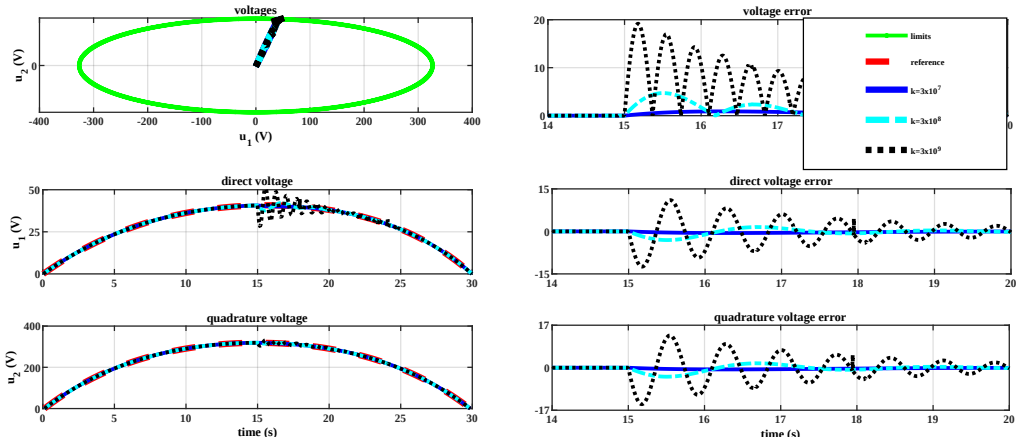
closer to their saturation limits (illustrated in Fig.3.8). Based on the simulations for the open-loop system, we consider some simplifications for the tracking control problem such as:

- only the discrepancies of the rotor angle and stator voltages are penalized in the MPC cost;
- only the perturbation on the motor speed is considered;
- the perturbation-affected state variables are assumed to always respect the constraints;
- one-step MPC is considered, i.e., $N_p = 1$.

The controller parameters are enumerated in the Table 3.3.



(a) errors of the output variables and of the rotor angle



(b) errors of the input variables

Figure 3.9.: Time evolutions in case (b).

Figures 3.9a-3.9b describe the time evolution and the discrepancies of the output, input variables and the rotor angle for the case of the perturbation-affected dynamics with an MPC-computed feedback. We observe reasonable tracking performance for the a priori generated reference profiles. Furthermore,

Table 3.8.: Dissipated energy.

Case	(a)	$k_0 = 3.10^7$	$k_0 = 3.10^8$	$k_0 = 3.10^9$
Energy [J]	2647.1	2647.3	2648.5	2669.1

increasing the angle weight parameter, k_0, increases the vibration of the state discrepancies without reducing the convergence time. The dissipated energy also increases with k_0, as seen in Table 3.8.

3.6. Position and angular constraints for a multicopter

3.6.1. Flat-based model for the multicopter dynamics

According to **C33**, the quadcopter system is differentially flat with the associated flat output $\mathbf{z} = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^\top$ defined as follows:

$$\mathbf{z} = \begin{bmatrix} x & y & z & \tan\left(\frac{\psi}{2}\right) \end{bmatrix}^\top. \tag{3.65}$$

Thus, the three angles, ϕ, θ and ψ, and the thrust, T, are expressed in terms of $\mathbf{z}$ as:

$$\phi = \arcsin\left(\frac{2z_4\ddot{z_1} - (1-z_4^2)\ddot{z_2}}{(1+z_4^2)\sqrt{\ddot{z_1}^2 + \ddot{z_2}^2 + (\ddot{z_3}+g)^2}}\right), \tag{3.66}$$

$$\theta = \arctan\left(\frac{(1-z_4^2)\ddot{z_1} + 2z_4\ddot{z_2}}{(1+z_4^2)(\ddot{z_3}+g)}\right), \tag{3.67}$$

$$\psi = 2\arctan(z_4), \tag{3.68}$$

$$T = \sqrt{\ddot{z_1}^2 + \ddot{z_2}^2 + (\ddot{z_3}+g)^2}. \tag{3.69}$$

This section presents a novel approach for imposing angle constraints for the trajectory generation of the thrust-propelled underactuated vehicle without any required knowledge on the yaw angle.

Proposition 3.6. *Let us consider the trajectory generation for the thrust-propelled dynamics described in **C33** with the flat output (3.65), and define the angle $\epsilon_{max} \in [0, \frac{\pi}{2})$ which verifies:*

$$\max_t \left(\frac{k_1^2(t) + k_2^2(t)}{k_3^2(t)}\right) \leq \tan^2(\epsilon_{max}), \tag{3.70}$$

where $k_1(t) = \ddot{z}_1(t)$, $k_2(t) = \ddot{z}_2(t)$ and $k_3(t) = \ddot{z}_3(t) + g$.
Then, the following inclusions:

$$\phi(t) \in [-\epsilon_{max}, \epsilon_{max}], \tag{3.71a}$$

$$\theta(t) \in [-\epsilon_{max}, \epsilon_{max}], \tag{3.71b}$$

*hold regardless the yaw angle. See the proof in **C39**.*

Let us illustrate the Proposition 3.6 by using some simple trajectories of k_1, k_2 and k_3 given as follows:

$$k_1(t) = 2\sin(t), \quad k_2(t) = \sin(2t), \quad k_3 = 0.5\sin(0.5t) + g. \tag{3.72}$$

The evaluations of the roll, $\phi = \Phi(k_1, k_2, k_3, z_4)$, and pitch, $\Theta(k_1, k_2, k_3, z_4)$, angles under different trajectories of the yaw angle represented by $z_4(t)$ are illustrated in Figure 3.10. We can observe that the values of the roll (red lines) and pitch (green lines) angles of the specific 3D trajectory vary according to the yaw angle values but they are always bounded by $-\epsilon(t)$ and $\epsilon(t)$ (blue lines).

This property is very useful and will be employed later for imposing constraints on the roll and pitch angles for the offline trajectory generation and online-tracking control design subject to saturation inputs.

Corollary 3.2. *Combining (3.70) with the standard bounds for thrust (3.69) (see e.g., (Bipin, Duggal, and Krishna 2014; Hehn and D'Andrea 2011)) which is given by:*

$$T_{min} \leq m\sqrt{k_1^2(t) + k_2^2(t) + k_3^2(t)} \leq T_{max}, \tag{3.73}$$

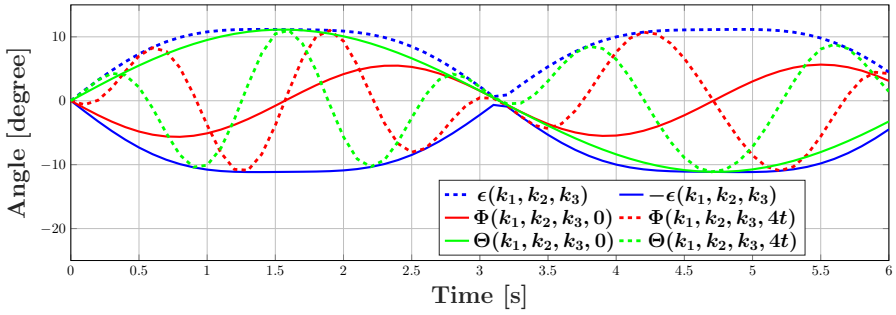


Figure 3.10.: Illustrations of the roll and pitch angles bounded by the angle boundary ϵ.

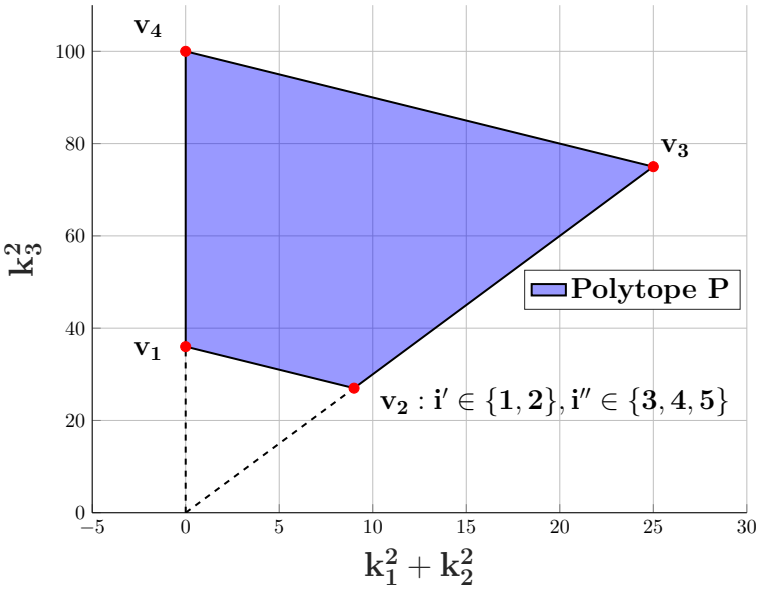


Figure 3.11.: Illustration of the polyhedral set P and its vertices as in (3.75).

it is derived that $\mathcal{K} \triangleq [k_1^2 + k_2^2 \quad k_3^2]^\top$ *lies inside a polytope* P *defined as:*

$$P = \Big\{\mathcal{K} \in \mathbb{R}^2 : \mathcal{K} = \sum_j \alpha_j v_j, \forall \alpha_j \geq 0, \\ \sum_j \alpha_j = 1, j = 1, \cdots, 4\Big\}, \tag{3.74}$$

where the extreme points $v_j \in V, j = 1, \cdots, 4$ *are parameterized after the three parameters appearing in* (3.70), (3.81)*:* ϵ_{max}, T_{min} *and* T_{max} *as follows:*

$$V = \Bigg\{ \left(0, \frac{T_{min}^2}{m^2}\right), \left(\frac{T_{min}^2}{m^2}\sin^2\epsilon_{max}, \frac{T_{min}^2}{m^2}\cos^2\epsilon_{max}\right), \\ \left(\frac{T_{max}^2}{m^2}\sin^2\epsilon_{max}, \frac{T_{max}^2}{m^2}\cos^2\epsilon_{max}\right), \left(0, \frac{T_{max}^2}{m^2}\right) \Bigg\}. \tag{3.75}$$

Illustrative example: Let us illustrate the Corollary 3.2 by using the multicopter dynamical system described in **C39** with the parameters $m = 0.5kg$, $T_{min} = 3N$, $T_{max} = 5N$ and $\epsilon_{max} = \pi/6$. The polytope P (3.74) and its vertices are depicted in Fig. 3.11.

Conditions for satisfying constraints on roll and pitch angular velocities are given in **C39**.

In the literature, various states and inputs constraints of the aerial systems are usually imposed based on a predefined yaw angle trajectory, e.g., zero angle (H. Lu, C. Liu, L. Guo, and W.-H. Chen 2017; M. W. Mueller and D'Andrea 2013; Cowling, Yakimenko, Whidborne, and Cooke 2007) or a spline with specific degree (Engelhardt, Konrad, Schäfer, and Abel 2016). I.e. introducing a predefined yaw angle trajectory into the multicopter dynamics, then, imposing constraints on the resulted system. However, the built-in controller of the CF quadcopter controls only the yaw rate, $\dot{\psi}_r$ and it does not provide very good tracking results, even for maintaining a constant direction (i.e. $\dot{\psi}_r(t) = 0$). Therefore, the unavoidable change in the yaw angle trajectory may cause violation of the constraints if we follow these foregoing approaches.

In order to overcome this restriction, we decouple the position and the yaw angle trajectory generation procedure. Firstly, we define the yaw angle rate reference trajectory as $\dot{\psi}_r(t) = 0$ since it is straightforward to be implemented and further verified. However, we will not use this information for the position trajectory generation (hereinafter, we call it shortly, trajectory generation).

The trajectory generation will take into account various states and inputs constraints such as boundary constraints, waypoints constraints, constraints on thrust, roll and pitch angles without requiring knowledge on a predefined yaw angle trajectory.

State constraints

Assume that the CF quadcopter has a known initial state composed of the position, and its higher derivatives up to the acceleration, i.e. $\{\xi_0,\ \dot{\xi}_0,\ \ddot{\xi}_0\}$. The trajectory is considered to satisfy the initial state constraints and also drive the CF quadcopter to a defined final state consisting of position, ξ_f, velocity $\dot{\xi}_f$ and acceleration $\ddot{\xi}_f$. By employing the the properties of the B-splines curve the boundary constraints are constructed as:

$$p_0 = \xi_0, \qquad \mathbf{P}M_1L_1\mathbf{B}_d(t=t_0) = \dot{\xi}_0, \qquad \mathbf{P}M_2L_2\mathbf{B}_d(t=t_0) = \ddot{\xi}_0, \tag{3.76}$$

$$p_n = \xi_f, \qquad \mathbf{P}M_1L_1\mathbf{B}_d(t=t_f) = \dot{\xi}_f, \qquad \mathbf{P}M_2L_2\mathbf{B}_d(t=t_f) = \ddot{\xi}_f. \tag{3.77}$$

Note that, $\mathbf{B}_d(t=t_0) = \begin{bmatrix}1,0,\ldots,0\end{bmatrix}^\top$ and $\mathbf{B}_d(t=t_f) = \begin{bmatrix}0,\ldots,0,1\end{bmatrix}^\top$.

Moreover, we consider a collection of $N+1$ waypoints[7] and the time instances associated to them (there must be no conflict with the boundary conditions (3.76), (3.77)):

$$\mathbb{W} = \{w_k\} \text{ and } \mathbb{T}_\mathbb{W} = \{t_k\}, \quad k \in \{0,\ldots,N\}. \tag{3.78}$$

The trajectory has to pass through each waypoint w_k at the time instant t_k, i.e.:

$$\mathbf{P}\mathbf{B}_d(t_k) = w_k. \tag{3.79}$$

Input constraints

Since the built-in controller of the CF quadcopter is composed of linear PID controllers constructed around the hovering states (roll and pitch angles equal zero), the ideal operating conditions are small values of these two angles. Thus, the reference trajectory is subject to saturation constraints on roll and pitch angles with desired maximum value $\epsilon_d \in \left(0, \frac{\pi}{2}\right)$ given as follows:

$$\left|\bar{\phi}\right| \leq \epsilon_d,\ \left|\bar{\theta}\right| \leq \epsilon_d. \tag{3.80}$$

Moreover, since we do not want the CF to have an aggressive altitude variation, the normalized thrust reference, $\bar{T}$, is also limited by its lower bound, $g - \Delta_g > 0$ and upper bound, $g + \Delta_g$, given as follows:

$$g - \Delta_g \leq \bar{T} \leq g + \Delta_g, \tag{3.81}$$

where $\Delta_g > 0$ is a desired parameter. Note that, the desired parameter Δ_g only needs to satisfy

[7] Note that, considering waypoints at the trajectory generation level is coherent with typical software-hardware UAV configurations which use waypoints in the communication protocol.

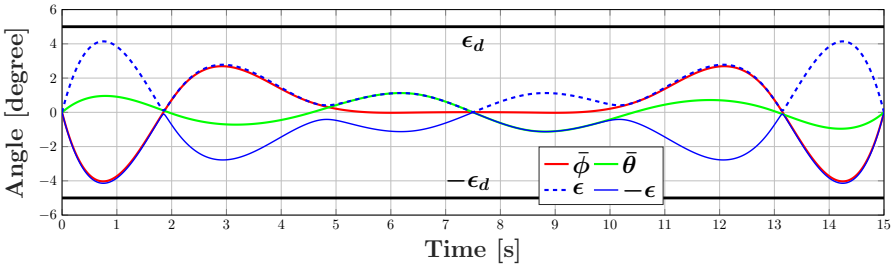


Figure 3.12.: Roll, pitch angles references $\bar{\phi}, \bar{\theta}$, and the angle boundary ϵ compared with the desired angle limit ϵ_d.

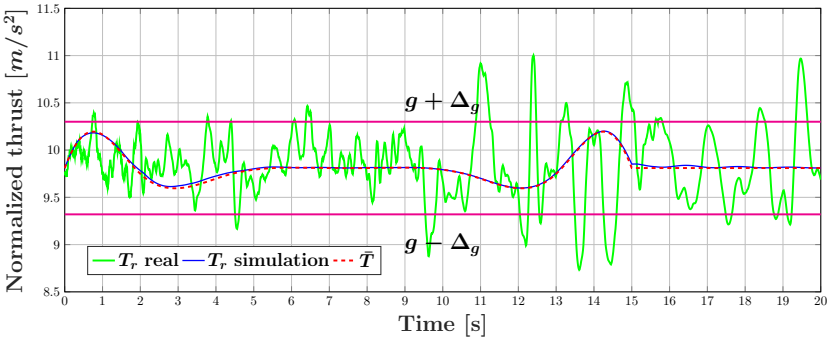


Figure 3.13.: The thrust input T_r under simulation and experiment compared with the thrust reference $\bar{T}$.

$g - \Delta_g > 0$. Consequently, the upper bound of the thrust $g + \Delta_g$ is always smaller than $2g$, thus, the thrust reference, $\bar{T}$, always respects the real thrust limit, $T_{limit} = 20.18 > 2g$.

3.6.2. Simulation and experimental results

The trajectory constraints include passing through the a priori given waypoints $\mathbb{W} = \left\{ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top, \begin{bmatrix} 0.5 & 2 & 1 \end{bmatrix}^\top, \begin{bmatrix} 1.5 & 2 & 1 \end{bmatrix}^\top, \begin{bmatrix} 2 & 0 & 0 \end{bmatrix}^\top \right\}$ with the associated time instants $\{0, 5, 10, 15\}$ seconds, respecting all the derivatives of $\{x, y, z\}$ up to 2^{nd} order equal 0 at the initial and final time instants. Furthermore, the roll and pitch angles are constrained to lie inside $[-\epsilon_d, \epsilon_d]$ with $\epsilon_d = 5^\circ$ and the normalized thrust $\bar{T}$ is also bounded by $g - \Delta_g$ and $g + \Delta_g$ with $\Delta_g = 0.05g$. We emphasize that the desired parameters $\epsilon_d = 5^\circ$ and $\Delta_g = 0.05g$ are validated with the thrust limit, $T_{limit} = 20.18 m/s^2$, thus ensuring the existence of the control parameters. The trajectory generation algorithms are implemented using Yalmip (Löfberg 2004) in Matlab 2015a with a total processing time of 7.81 sec. In Figure 3.12, we provide results of the roll, pitch angles $\bar{\phi}$ and $\bar{\theta}$ (plotted in red and green lines) which are actually bounded by the angle boundary ϵ (plotted in blue lines). Moreover, the angle boundary ϵ does not exceed the desired angle limit ϵ_d. Furthermore, in Figure 3.13, the thrust reference $\bar{T}$ (plotted in dashed red line) stays within the desired thrust limits, $[g - \Delta_g, g + \Delta_g]$. The 3D reference trajectory, the waypoints and the tracking results is presented in Figure 3.14.

After having the reference trajectory, the tracking controller is constructed by using a feedback linearization law facilitated by the nested control design **J9**.

3.7. Conclusions and discussions

This chapter presented a novel approach based on B-splines and their properties for continuous-times constraint satisfaction. More precisely, locality, positivity and convexity properties were merged together to obtain continuous time guarantees for set inclusion and exclusion. Mixed integer formulations

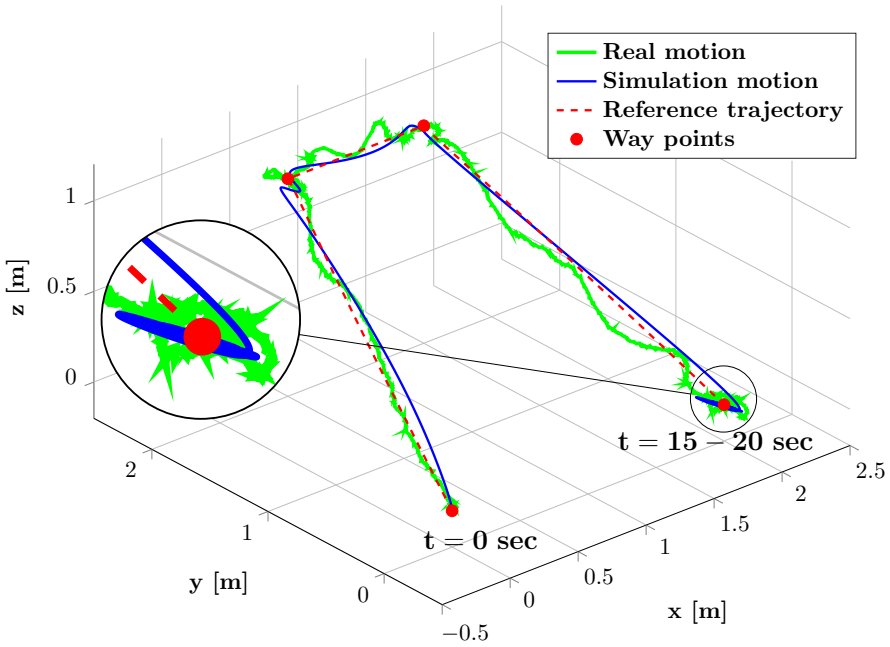


Figure 3.14.: Quadcopter motions under simulation and experiment.

were derived for exclusion constraints.

We next concentrated on the use of B-splines in conjunction with differential flat systems and their application to motion planning problems where continuous time constraint satisfaction on the states and inputs is paramount. For example, we discussed the trajectory generation for multicopter systems which took into account boundary constraints, waypoints constraints, constraints on thrust, and roll, pitch angles without requiring knowledge on a predefined yaw angle trajectory. Thus, possible tracking errors in yaw angle do not affect the validation of the proposed constraints. The trajectory generation problem with communication constrains has been implemented and verified through experimental results for USVs (Unmanned Surface Vehicle). Finally, we proposed an optimization-based control design for the dissipated energy minimization of an elevator system of a DC microgrid while respecting the system dynamics and constraints. An efficient combination among differential flatness, B-spline parameterization and tracking Model Predictive Control was employed and validated through simulations and comparisons.

While differential flatness is a well established method in control, there is still place for other applications. For example, data-based (i.e., learning procedures) are often performing better than classical, model-based, approaches but lack formal guarantees of stability and performance. Hence, approaches which thread the two worlds are promising from both practical and theoretical viewpoints. E.g., (Greeff and Schoellig 2020) provides a robust learning mechanism through the use of differential flatness theory.

4. Mixed-integer representations in motion planning

This chapter addresses the problem of collision avoidance in a multi-obstacle environment and focuses on its representation in optimization-based control problems. We make use of hyperplane arrangements to characterize these regions, provide zonotopic over-approximations and highlight their benefits when introduced in the obstacle avoidance problem. Further, we consider an adaptation of the generic MPC (Model Predictive Control) trajectory tracking problem, aiming to guarantee feasibility and the convergence.

The chapter is based on several publications:

J10 Ioan, D., S. Olaru, **I. Prodan**, F. Stoican, and S.-I. Niculescu. "From Obstacle-Based Space Partitioning to Corridors and Path Planning. A Convex Lifting Approach". In: *IEEE Control Systems Letters*, pp. 79–84, 2019. IEEE.
DOI: 10.1109/LCSYS.2019.2922414.

C57 Ioan, D., **I. Prodan**, S. Olaru, F. Stoican, and S. Niculescu. "Navigation in cluttered environments with feasibility guarantees", pp. 1–6. In: *Proceedings of the 21th IFAC World Congress*. 2020. Berlin, Germany.

C50 Ioan, D., **I. Prodan**, F. Stoican, S. Olaru, and S. Niculescu. "Complexity bounds for obstacle avoidance within a zonotopic framework", pp. 1556–1563. In: *Proceedings of the IEEE American Control Conference (ACC'19)*. 2017. Philadelphia, USA.

C49 Ioan, D., S. Olaru, **I. Prodan**, F. Stoican, and S. Niculescu. "Zonotopic-based description of a multi-obstacle environment with feasibility guarantees", pp. 1772–1777. In: *Proceedings of the 26th IEEE European Control Conference (ECC'19)*. 2019. Naples, Italy.

C48 Ioan, D., S. Olaru, **I. Prodan**, F. Stoican, and S.-I. Niculescu. "Parametrized Hyperplane Arrangements for Control Design with Collision Avoidance Constraints", pp. 52–59. In: *Proceedings of the 15th IEEE International Conference on Control & Automation*. 2019. Edinburgh, Scotland.

4.1. Brief literature review

The recent proliferation of real-world applications involving unmanned vehicles has led to a growing interest in both the robotics and control communities on developing reliable collision-free control strategies within a multi-obstacle environment (Janecek, Klauco, Kaluz, and Kvasnica 2017), (Manyam, Casbeer, and Sundar 2016). Among such applications we may mention monitoring or surveillance (Puri 2005), autonomous overtaking (Ballesteros-Tolosana, Olaru, Rodriguez-Ayerbe, Pita-Gil, and Deborne 2017) or precision agriculture (H. Jawad, Nordin, Gharghan, A. Jawad, and Ismail 2017). The main challenge resides in the non-convexity of the feasible regions in the motion space.

There exist in the literature two main approaches dealing with the path/trajectory planning in a non-convex feasible space. The optimization-based methods state the problem as a constrained optimization one either over a non-convex domain (e.g. mixed-integer formulation (Richards and How 2002), **[M1]**) or with a non-linear cost (e.g. potential field formulation (Y.-b. Chen, Luo, Mei, J.-q. Yu, and X.-l. Su 2016; Olfati-Saber and Murray 2002)). The second class of approaches, the sampled (or graph) -based methods, replaces the search of feasible paths with the search of the shortest (w.r.t. a predefined criterion) path within a graph whose nodes are randomly selected samples (Latombe 2012), the most popular ones: PRM (probabilistic roadmaps) (Karaman and Frazzoli 2011), RRT (rapidly-exploring random tree) (LaValle 1998) and their variants. Moreover, there are some techniques, like tunnel MILP (Vitus, Waslander, and Tomlin 2008), which combine these two classes of methods.

Generally, regardless of the approach, a polytopic representation of the obstacles is used. This allows, especially for the first type of methods, an efficient characterization of the non-convex and non-connected feasible domain as a union of convex regions defined by a hyperplane arrangement in a mixed-integer formalism. The main drawback is that the complexity of the control problem is strongly dependent on the number of obstacles. Nonetheless, various technical procedures have been developed to manage the complexity of the formulation: cell merging **J3**, logarithmic formulations (Vielma and Nemhauser 2011) or the use of zonotopic approximations **J10**.

Zonotopes are becoming ubiquitous (here, but also in other control areas like: collision detection (Guibas, A. Nguyen, and L. Zhang 2003), reachability analysis (Althoff, Stursberg, and Buss 2010), fault diagnosis (Puig 2010) or guaranteed state estimation (Alamo, Bravo, and Camacho n.d.)). As well, there are toolboxes like CORA (Althoff 2015) able to manage their representation. Noteworthy, the approximation may change the topology of the environment (e.g., passages can be blocked).

From the viewpoint of motion planning we may consider two distinct sub-tasks: path/trajectory generation and trajectory tracking. Moreover, these (sub-)tasks are usually viewed as independent or decoupled problems (Latombe 2012). For instance, the classical sample-based[1] approaches, (Karaman and Frazzoli 2011), (LaValle 1998), are prone to focus on the first task, neglecting the second one and, thus, simplifying the problem. Next, the optimization-based strategies, e.g. mixed-integer formulations (Richards and How 2002), **[M1]**, convexification techniques (Szmuk, Pascucci, Dueri, and Acikmese 2017), potential field methods (Y.-b. Chen, Luo, Mei, J.-q. Yu, and X.-l. Su 2016) or set-theoretic approaches (Franzè and Lucia 2015) merge the planning and tracking tasks at the expense of a higher computational complexity, especially in the case of cluttered/congested multi-obstacle environments. In addition, there are a few works, e.g., (A. Weiss, Danielson, Berntorp, I. Kolmanovsky, and Cairano 2017), that have a more intrinsic approach, considering planning and tracking as distinct but interacting tasks (Berntorp, A. Weiss, Danielson, I. V. Kolmanovsky, and Di Cairano 2017) and employing appropriate methods from the previously mentioned categories. For instance, in this last category, we may include (Berntorp, Danielson, A. Weiss, and Di Cairano 2018), where the authors proposed an extension of the RRT (rapidly-exploring random tree), using feedback control and positively invariant sets to guarantee collision-free tracking.

The main drawback of the methods involving sample-based techniques is their *probabilistic completeness* (Barraquand et al. 1997). Specifically, the probability that the algorithm returns a feasible solution tends to 1 if the number of sample points is sufficiently large (approaching ∞), as it was em-

[1] Sample-based approaches are generally related to the construction of a graph structure.

pirically shown in (Hsu, Latombe, and Kurniawati 2007). Moreover, these probabilistic completeness proofs do not provide any guarantee on the time in which the algorithm finds the optimal path (if there exists one). Therefore, offering *global feasibility guarantees* is a worthy endeavor.

4.2. Contributions

The main contributions of this chapter are in the areas of:

Zonotopic over-approximations Owing to the geometric properties of zonotopic sets (due to their generator representation), we propose over-approximating a collection of polyhedral obstacles to reduce the problem's complexity. Taking zonotopes with a common generator seed and which differ through their scaling factors and center allows to provide bounds for the complexity of the representation (using the hyperplane arrangement notions) and to optimize the over-approximation performance. The details are provided in Subsection 4.3.3 with emphasis on the $mathbbR^2, \mathbb{R}^3$ cases, which have particular relevance in the motion planning setting.

Mixed-integer formulations In a sense, the mixed-based approaches are the most generic as, at each step, an optimization problem such as (4.37)–(4.37d) is solved. The main difficulty here comes from the non-convex formulation of the feasible domain. This is mitigated to some extent by methods which exploit the problem's structure (e.g., a partition induced by a hyperplane arrangement, dynamic obstacle recognition, etc.). Still, the unavoidable fact is that the shape of the obstacles is directly linked with the problem's complexity. The idea, explored in Subsection 4.4.1 and based on the results from Subsection 4.3.3, is to provide zonotopic over-approximations of the polytopic obstacles. This allows to decouple the complexity of the environment from the complexity of the problem passed to the solver.

Corridor-based formulations In some circumstances, solving a mixed-integer problem is not possible (limited resources, large number of obstacles, etc.). The popular idea is to provide a corridor through which the agent has to pass. By limiting the feasible domain to a union of convex shapes it is sometimes possible to fall-back to a classical quadratic-cost problem. Even if discrete decisions (which imply the use of binary variables) still have to be taken, these are reduced in number: instead of deciding at each simulation step which separating hyperplane to consider, we only have to pay attention when passing from one piece of the corridor to the next one. Subsection 4.4.2 provides a corridor design procedure which again takes into account the zonotopic over-approximations from Subsection 4.3.3 and Subsection 4.4.3 builds upon the preceding results to offer recursive feasibility guarantees.

4.3. Geometric tools for obstacle characterizations

Obstacles, targets, safety regions and other regions of interest are often modeled through ellipsoidal or polyhedral sets in motion planning (LaValle 2006). Even if typical applications are in $\mathbb{R}^2$ or $\mathbb{R}^3$ there are still significant numerical or theoretical issues to be considered. Hence, in this chapter we recapitulate and investigate implementations based on zonotopic sets and the hyperplane arrangement that they span.

In effect we are interested in efficient characterizations (e.g., through zonotopic over-approximations) of the feasible domain $\mathbb{R}^n \setminus \mathbb{P}$ with $\mathbb{P} \subset \mathbb{R}^n$ denoting a collection of N_o disjoint regions:

$$\mathbb{P} = \bigcup_{j=1}^{N_o} P_j, \quad P_i \cap P_j = \emptyset, \forall i \neq j. \tag{4.1}$$

4.3.1. Zonotopic sets

Recall the definition of a support function (I. Kolmanovsky and Gilbert 1998), $h_Q(\eta) = \sup_{q\in Q} \eta^T q$, evaluated at $\eta \in \mathbb{R}^d$ for a given set $Q \subset \mathbb{R}^d$. Further, let us consider two sets X and Y, then the inclusion $X \subseteq Y$ holds only iff $h_X(\eta) \leq h_Y(\eta), \forall \eta \in \mathbb{R}^d$. If, moreover, Y is a polytope in the half-space form:

$$Y = \{x \in \mathbb{R}^d : s_i^\top x \leq r_i, i = 1 \dots n_v\}, \tag{4.2}$$

the inclusion condition becomes $h_X(s_i) \leq r_i, \forall i$. If X is a polytope described by its extreme points:

$$X = \{x \in \mathbb{R}^d : x = \sum \alpha_j v_j, \sum \alpha_j = 1, \alpha_j \geq 0\}, \tag{4.3}$$

the inclusion condition is rewritten as $s_i^\top v_j \leq r_i, \forall i, j$.

We consider zonotopes, a class of polytopes, endowed with a third representation due to their symmetry which provide a good balance between numerical complexity and fidelity of representation.

Definition 4.1 (Zonotopes-(Kuhn 1998))**.** *A zonotope is a centrally symmetric polytope, which can be described as a Minkowski sum of line segments. In its generator representation a zonotope $\mathcal{Z}(G, c)$ is described by center $c \in \mathbb{R}^d$ and generator matrix $G = \begin{bmatrix} g_1 & \dots & g_m \end{bmatrix} \in \mathbb{R}^{d\times m}$:*

$$\mathcal{Z}(G, c) = \{c + \sum_{k=1}^{m} \xi_k g_k : \|\xi\|_\infty \leq 1\}. \tag{4.4}$$

Zonotopes own several properties of practical interest (Fukuda 2004):

i) are closed under linear transformation:

$$R\mathcal{Z}(G_1, c_1) = \mathcal{Z}(RG_1, Rc_1); \tag{4.5}$$

ii) are closed under Minkowski sum:

$$\mathcal{Z}(G_1, c_1) \oplus \mathcal{Z}(G_2, c_2) = \mathcal{Z}(\begin{bmatrix} G_1 & G_2 \end{bmatrix}, c_1 + c_2); \tag{4.6}$$

iii) their volume has an explicit formulation (Gover and Krikorian 2010):

$$\mathrm{Vol}(\mathcal{Z}(G, c)) = \sum_{1\leq j_1<j_2\dots j_d\leq m} \left|\det(G^{j_1\dots j_d})\right|, \tag{4.7}$$

where $G^{j_1\dots j_d}$ denotes the matrix composed from columns of indices $j_1 \dots j_d$ from G.

From the generators of a zonotope, its half-space representation can be constructed: to each sequence of $d-1$ generators $1 \leq j_1 < j_2 \dots j_{d-1} \leq m$ corresponds the pair $(h_i, k_i) \in \mathbb{R}^d \times \mathbb{R}$, where:

$$h_i \perp g_{j_l}, \forall j_l \in \{j_1 \dots j_{d-1}\}, \qquad k_i = \sum_{j_l \notin \{j_1\dots j_{d-1}\}} \left|h_i^\top g_{j_l}\right|. \tag{4.8}$$

Adding the center from generator representation, the halfspace description (4.8) becomes:

$$\mathcal{Z}(G, c) = \bigcap_{1\leq j_1<\dots j_{d-1}\leq m} \{x \in \mathbb{R}^d : |h_i(x - c)| \leq k_i\}. \tag{4.9}$$

Using the support function notion, the inclusion $\mathcal{Z}(G, c) \subseteq Y$, with Y given as in (4.2), is valid iff:

$$s_i^\top c + \sum_j \left|s_i^\top g_j\right| \leq r_i \quad \forall i, j. \tag{4.10}$$

The inclusion of a polytopic set X, given as in (4.3), into a zonotope $X \subseteq \mathcal{Z}(G, c)$ holds iff:

$$\left|h_i^\top (v_j - c)\right| \leq k_i \quad \forall i, j. \tag{4.11}$$

4.3.2. Hyperplane arrangements

Consider a finite collection of hyperplanes from $\mathbb{R}^d$:

$$\mathbb{H} = \{\mathcal{H}_i\}_{i \in \mathcal{I}}, \tag{4.12}$$

with $\mathcal{H}_i = \{x \in \mathbb{R}^d : s_i^\top x = r_i\}$. Each of these hyperplanes divides the space in two disjoint regions:

$$\mathcal{R}_i^{\pm} = \left\{x \in \mathbb{R}^d : \pm s_i^\top x \leq \pm r_i\right\}. \tag{4.13}$$

Next, the space can be partitioned into cells using the hyperplane arrangement notion.

Definition 4.2 (Hyperplane arrangements – (Ziegler 2012))**.** *The collection $\mathbb{H}$ partitions the space into a union of disjoint cells $\mathcal{A}(\sigma)$, characterized by a sign tuple $\sigma \in \{-,+\}^N$:*

$$\mathcal{A}(\sigma) = \bigcap_{i \in \mathcal{I}} \mathcal{R}_i^{\sigma(i)}. \tag{4.14}$$

The hyperplane arrangement of cells covering the entire space is the union of all feasible sign tuples:

$$\mathcal{A}(\mathbb{H}) = \bigcup_{l=1 \ldots \gamma(N)} \mathcal{A}(\sigma_l), \tag{4.15}$$

where $\sigma_l \in \{-,+\}^N$ is the sign tuple resulting from a non-empty intersection of half-spaces and $\gamma(N)$ is the number of feasible cells. ♦

A (sub-)arrangement $\mathcal{B} \subseteq \mathbb{H}$ is called *central* if $\bigcap_{\mathcal{H}_i \in \mathcal{B}} \mathcal{H}_i \neq \emptyset$. Notation $\#\mathcal{B}$ denotes the number of hyperplanes and $\text{rank}(\mathcal{B})$ the rank of the intersection.

Theorem 4.1 (Whitney's theorem – (Stanley 2004))**.** *Let $\mathcal{A}$ be an arrangement in an d-dimensional space. Then the characteristic polynomial of $\mathcal{A}$ is defined as*

$$\chi_{\mathcal{A}}(t) = \sum_{\mathcal{B} \subseteq \mathcal{A}, \mathcal{B} \text{ central}} (-1)^{\#\mathcal{B}} t^{d - rank(\mathcal{B})}. \tag{4.16}$$

The total number of regions and the number of bounded regions characterizing the arrangement are:

$$r(\mathcal{A}) = (-1)^d \chi_{\mathcal{A}}(-1), \quad b(\mathcal{A}) = (-1)^{rank(\mathcal{A})} \chi_{\mathcal{A}}(1). \tag{4.17}$$

For a hyperplane arrangement with n hyperplanes in general position[2], the bounds are therefore:

$$r(\mathcal{A}) = \sum_{k=0}^{d} \binom{n}{k}, \quad b(\mathcal{A}) = \binom{n}{d-1}. \tag{4.18}$$

4.3.3. Zonotopic over-approximations with fixed generator seed

Considering Definition 4.1 we refer to a family of zonotopes parametrized after their centers $c_\ell \in \mathbb{R}^d$ and scaling factors $\Delta_\ell \in \mathbb{R}^{m \times m}$ applied to a common generator "seed" (an a priori given matrix $G \in \mathbb{R}^{d \times m}$):

$$\mathcal{Z}(G\Delta_j, c_j), \quad j = 1 \ldots N_o. \tag{4.19}$$

[2] An arrangement is considered to be in *general position* iff no two hyperplanes are parallel (i.e., share the same normal).

Δ_ℓ is a diagonal matrix. The k-th diagonal element is noted as δ_{j_k}[3].

Having a common generator seed:

i) provides linear inclusion constraints in term of the scaling and center parameters;

ii) allows measures for the obstacle over-approximation in terms of ℓ_1, ℓ_∞ norms and the volume;

iii) reduces the representation complexity (i.e., the number of cells of the associated arrangement).

Replacing g_k with $g_k \cdot \delta_{j_k}$ in (4.8), the half-space representation of the j-th zonotope from (4.19) becomes:

$$h_i \text{ s.t. } h_i \perp g_k, \forall k \in \{k_1 \dots k_{d-1}\}, \tag{4.20a}$$

$$k_i(\Delta_j) = \sum_{k \notin \{k_1 \dots k_{d-1}\}} \left| h_i^\top g_k \right| \delta_{j_k}, \tag{4.20b}$$

where i enumerates the

$$p(d, m) = \binom{m}{d-1} \tag{4.21}$$

combinations of $d-1$ distinct generators selected from the m available ones (i.e., $1 \leq k_1 < \dots < k_{d-1} \leq m$).

Remark 4.1. Note that h_i remains unchanged w.r.t. (4.8) because the subspace perpendicular on $\{g_k\}_{k \in \{k_1, \dots k_{d-1}\}}$ is identical with the one perpendicular on $\{g_k \cdot \delta_{j_k}\}_{k \in \{k_1, \dots k_{d-1}\}}$, regardless of δ_{j_k}. ♦

We gather the support hyperplanes (4.20) into the collection:

$$\tilde{\mathbb{H}} = \{\mathcal{H}(h_i, \pm k_i(\Delta_j))\}. \tag{4.22}$$

Using the parametrization (4.20), with sets X, Y defined in (4.2)-(4.3), allows to reformulate the inclusion conditions (4.10), (4.11) into a linear form w.r.t. parameters c_j, Δ_j:

$$\mathcal{Z}(G\Delta_j, c_j) \subseteq Y : s_i^\top c_j + \sum_{k=1}^{m} \left| s_i^\top g_k \right| \cdot \delta_{j_k} \leq r_i, \ \forall i, \tag{4.23a}$$

$$X \subseteq \mathcal{Z}(G\Delta_j, c_j) : \left| h_i^\top (v_k - c_j) \right| \leq k_i(\Delta_j), \ \forall k. \tag{4.23b}$$

The overall goal is to provide adequate over-approximations (4.19) for the collection (4.1). That is, seek a (symmetric) zonotope $\mathcal{Z}(G\Delta_j, c_j)$ enclosing the (usually non-symmetric) polytope P_j such that a specific measure parametrized after c_j, Δ_j is minimized:

$$(\Delta_j, c_j)^* = \arg\min_{\Delta_j, c_j} \mathcal{C}(\Delta_j, c_j) \text{ s.t. } P_j \subseteq \mathcal{Z}(G\Delta_j, c_j). \tag{4.24}$$

For the cost in (4.24), we consider the following measures:

i) zonotope volume $\text{Vol}(\mathcal{Z}(G\Delta_j, c_j))$:

$$\mathcal{C}(\Delta_j, c_j) = \sum_{1 \leq k_1 < \dots < k_d \leq m} \left(\left| \det(G^{k_1 \dots k_d}) \right| \cdot \prod_{k \in \{k_1, \dots, k_d\}} \delta_{j_k} \right); \tag{4.25}$$

ii) generator sum $\sum_{k=1}^{m} g_k \delta_{j_k}$:

$$\mathcal{C}(\Delta_j, c_j) = ||\Delta_j||_1 = \sum_{k=1}^{m} \delta_{j_k}; \tag{4.26}$$

[3]If not explicitly stated otherwise, we consider that the diagonal entries are distinct since having the same diagonal entry is only a simplification.

iii) largest generator $\max\limits_{k=1\ldots m} g_k\delta_{j_k}$:

$$\mathcal{C}(\Delta_j, c_j) = ||\Delta_j||_\infty = \max_{k=1\ldots m} \delta_{j_k}. \tag{4.27}$$

Remark 4.2. Volume (4.25) is a sum of polynomial terms $\prod \delta_{j_k}$, thus, non-linear. Imposing equality among the scaling factors ($\delta_{j_1} = \ldots = \delta_{j_m} = \tilde{\delta}_j$), leads to a simplified volume formulation which can be used instead of (4.25). Yet, solving (4.24) with (4.25) is burdensome in higher dimensions and the simpler (but less accurate) measures, i.e., the norms (4.26) and (4.27) need to be considered. ♦

As shown in (4.22), the $2 \cdot p(d,m) \cdot N_o$ hyperplanes are partitioned in $2N_o$ families: in each family there is a hyperplane with normal h_i but with a different offset $\pm k_i(\Delta_j)$. This particularity allows the following proposition.

Proposition 4.1. *$\mathcal{A}(\tilde{\mathbb{H}})$ has the characteristic polynomial:*

$$\chi(\mathcal{A}) = \sum_{k=0}^{d} (-1)^k t^{d-k} \cdot \binom{p(d,m)}{k} \cdot (2N_o)^k, \tag{4.28}$$

to which correspond the following bounds:

$$r(\mathcal{A}) = \sum_{k=0}^{d} \binom{p(d,m)}{k} \cdot (2N_o)^k, \quad b(\mathcal{A}) = \left| \sum_{k=0}^{d} (-1)^k \binom{p(d,m)}{k} \cdot (2N_o)^k \right|, \tag{4.29}$$

the total number and respectively the bounded number of regions.

Proof. To apply Whitney's theorem and obtain the bounds (4.18), the key is to count the central arrangements of rank k. Hence, we need to avoid selecting parallel hyperplanes from the available $p(d,m) \cdot 2N_o$. We consider each of the $2N_o$ families of $p(d,m)$ hyperplanes and select as follows: from the 1st family we select j_1^-; in the 2nd family there remain $m - j_1^-$ available (the others are parallel to those already selected and thus unsuitable) from which we select j_1^+; the procedure continues up to the $2N_0$-th family from which we select $j_{N_o}^+$. Enumerating all sequences $j_1^-, \ldots, j_{N_o}^+$ which add to k and taking all the possible combinations for a fixed sequence we obtain the total number of central arrangements of rank k as a sum of products of combinatorial terms.

$$\sum_{j_1^- + j_1^+ + \cdots + j_{n_o}^- + j_{n_o}^+ = k} \binom{p}{j_1^-} \cdot \binom{p - j_1^-}{j_1^+} \cdots \binom{p - j_1^- - j_1^+ - \cdots - j_{N_o}^-}{j_{N_o}^+}. \tag{4.30}$$

Writing explicitly these terms for a sequence $j_1^-, j_1^+ \ldots, j_{N_o}^-, j_{N_o}^+$

$$\frac{p!}{j_1^-! \cdot (p - j_1^-)!} \cdot \ldots \cdot \frac{(p - j_1^- \cdots - j_{n_o}^-)!}{j_{n_o}^+! \cdot (p - j_1^- \ldots j_{N_o}^- - j_{N_o}^+)!}$$
$$= \frac{k!}{(p-k)! \cdot k!} \cdot \frac{k!}{j_1^-! \cdot j_1^+! \cdot \ldots \cdot j_{N_o}^-! \cdot j_{N_o}^+!} = \binom{p}{k} \cdot \frac{k!}{j_1^-! \cdot j_1^+! \cdot \ldots \cdot j_{N_o}^-! \cdot j_{N_o}^+!}$$

and identifying each as the coefficient from the *multinomial theorem* (Hazewinkel 2001) allows to rewrite the sum of products in a simplified form.

$$\sum_{j_1^- + \ldots j_{N_o}^+ = k} \binom{p}{k} \cdot \frac{k!}{j_1^-! \cdot j_1^+! \ldots j_{N_o}^-! \cdot j_{N_o}^+!} = \binom{p}{k} \cdot (2N_o)^k. \tag{4.31}$$

Introducing (4.31) in (4.16) leads directly to (4.28) which, applied as in (4.17), leads to (4.29). □

These results allow to further derive a bound for the number of generators in the zonotopic representation and to provide a rigorous bound for $d = 3$.

Corollary 4.1. *Assuming n_o^* support hyperplanes in (4.1), for any $m \in \mathbb{N}^+$ which verifies:*

$$\sum_{k=0}^{d} \binom{p(d,m)}{k} \cdot (2N_o)^k \leq \sum_{k=0}^{d} \binom{n_o^*}{k}, \tag{4.32}$$

the arrangement $\mathcal{A}(\tilde{\mathbb{H}})$ has fewer cells than $\mathcal{A}(\mathbb{H})$.

Proof. The left-side of (4.32) comes from (4.29) and the right side from (4.18) with $n \mapsto n_o^*$. □

Corollary 4.2. *For the case $d = 3$, we have that:*

i) the total and the bounded number of cells for $\mathcal{A}(\tilde{\mathbb{H}})$ is:

$$r(\mathcal{A}) = 1 + \eta \cdot 2N_o + \eta(\eta-1) \cdot 2N_o^2 + \frac{1}{6}\eta(\eta-1)(\eta-2)8N_o^3, \tag{4.33a}$$

$$b(\mathcal{A}) = \left| 1 - \eta \cdot 2N_o + \eta(\eta-1) \cdot 2N_o^2 - \frac{\eta(\eta-1)(\eta-2)}{6} 8N_o^3 \right|; \tag{4.33b}$$

where $\eta = \frac{1}{2}m(m-1)$;[4]

ii) the largest m s.t. $\mathcal{A}(\tilde{\mathbb{H}})$ has fewer cells than $\mathcal{A}(\mathbb{H})$ is:

$$m = \left\lfloor \frac{1 + \sqrt{1 + 8\eta^*}}{2} \right\rfloor. \tag{4.34}$$

where η^* represents the real solution of a solvable third order equation.

Proof. The detailed proof can be found in **C50**. □

Remark 4.3. Likewise, the particularization can be done for the case $d = 2$ and we obtain similar results. The difference consist of: $\eta \to m$ (because of $p(2,m) = \binom{m}{1} = m$) and, in both $r(\mathcal{A})$ and $b(\mathcal{A})$ the last term is eliminated:

$$m = \left\lfloor \frac{(N_o - 1) + \sqrt{(N_o-1)^2 + n_o^*(n_o^*+1)}}{2N_o} \right\rfloor. \tag{4.35}$$

4.4. Navigation in a cluttered environment with feasibility guarantees

In what follows, our aim is to obtain a collision-free trajectory which avoids a collection of obstacles (4.1) for an agent described by the following LTI dynamics:

$$x_{k+1} = Ax_k + Bu_k, \, \forall k, \tag{4.36}$$

with $x_k \in \mathbb{R}^d$ the state vector, $u_k \in \mathbb{R}^{d_u}$ the input vector and the matrices A, B of appropriate dimension. Also, the physical constraints lead to the compact sets $\mathcal{X}$ and $\mathcal{U}$ from $\mathbb{R}^d$ and $\mathbb{R}^{d_u}$, respectively. Dynamics (6.19) are further embedded into a MPC (Model Predictive Control) strategy:

[4] Also, the particularization can be done for the case $d = 2$. In (4.33) η is replaced by m (because of $p(2,m) = \binom{m}{1} = m$) and, in both $r(\mathcal{A})$ and $b(\mathcal{A})$ the last term is eliminated.

$$\min_{\mathbf{u}} |x^T_{k+N_p|k}\|^2_P + \sum_{l=1}^{N_p-1} \|x^T_{k+l|k}\|^2_Q + \sum_{l=1}^{N_p-1} \|u^T_{k+l|k}\|^2_R \tag{4.37a}$$

$$\text{s.t.} \quad x_{k+l|k} = Ax_{k+l-1|k} + Bu_{k+l-1|k}, \tag{4.37b}$$

$$x_{k+l|k} \in \mathcal{X}, u_{k+l|k} \in \mathcal{U}, \tag{4.37c}$$

$$x_{k+l|k} \in \mathcal{X} \setminus \mathbb{P}. \tag{4.37d}$$

In (4.37a), N_p is the prediction horizon, the matrices P (terminal cost penalty), Q (output error penalty) and R (control move penalty) are positive semi-definite and of appropriate dimensions.

Unless otherwise stated, the dynamics considered will be those of the double integrator where the state is composed from position (also the system's output) and velocity components and the input is the acceleration to the system (we will consider $d = 2$ or $d = 3$, as is appropriate for each case):

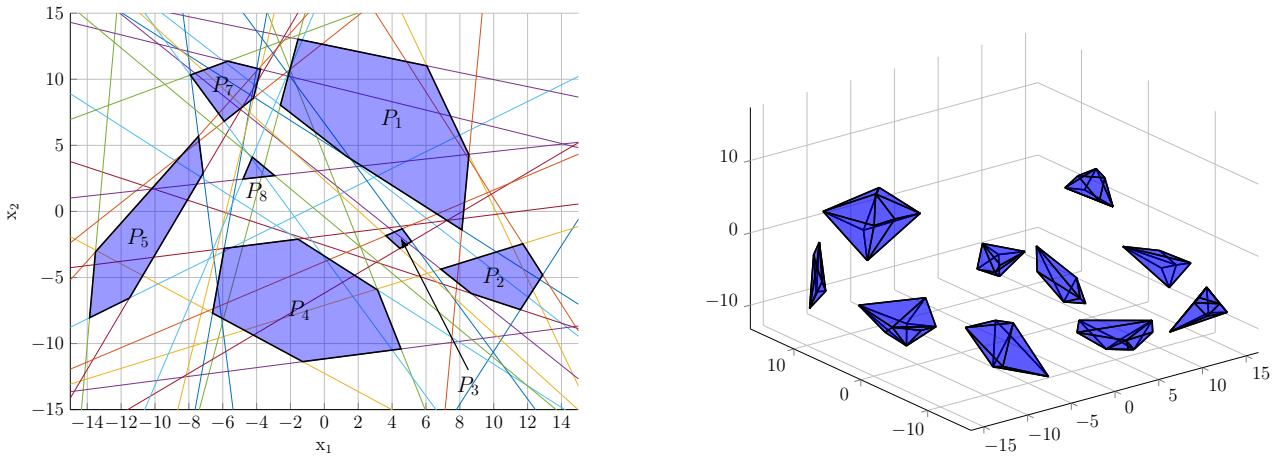
$$A = \begin{bmatrix} O_d & I_d \\ O_d & -\frac{\mu}{M} I_d \end{bmatrix}, \quad B = \begin{bmatrix} O_d \\ M I_d \end{bmatrix},$$

where $\mu = 3$ and $M = 60$. The agent's state and input are constrained: $\mathcal{X} = \{x : -15 \leq x_i \leq 15, \forall i = 1 \dots 2d\}$ and $\mathcal{U} = \{u : -1 \leq u_i \leq 1, \forall i = 1 \dots 2d\}$.

The main problem for the navigation in the cluttered environment is the obstacle avoidance. Assuming that the obstacles (their position and shape are known) we may consider two major approaches: mixed-integer, explored in Subsection 4.4.1 and corridor-based, detailed in Subsection 4.4.2.

4.4.1. Mixed-integer MPC with zonotopic over-approximations

Consider in Figure 4.1 two collections[5] of obstacles (in $\mathbb{R}^2$ and $\mathbb{R}^3$, respectively) as in (4.1) and delineate in Table 4.1 the number of: support hyperplanes ($\#\mathbb{H}$); feasible cells[6] ($\gamma^*(N)$) with their offline constructing time ($t_{\gamma^*(N)}$ with $N = \#\mathbb{H}$); and interdicted cells ($\#\Sigma_{\mathbb{P}}$).



(a) $d = 2$ and $N_o = 7$ (b) $d = 3$ and $N_o = 10$

Figure 4.1.: Hyperplane arrangement for polytopic obstacles.

Several issues become apparent:

i) the problem complexity is strongly correlated with the number of arrangement cells ($\gamma^*(N)$);

ii) the cell number depends on the number of hyperplanes from the obstacle description ($\#\mathbb{H}$).

[5] To avoid cluttering the image, we do not illustrate the hyperplanes for the 3D case.

[6] We take into account only the cells from the bounded domain $\mathcal{X}$.

	$\#\mathbb{P}$	$\#\mathbb{H}$	$\gamma^*(N)$	$t_{\gamma^*(N)}$ [s]	$\#\Sigma_{\mathbb{P}}$
$d=2$	7	34	419	9.22	75
$d=3$	10	106	57185	1255,88	1378

Table 4.1.: Parameters and computing time of the hyperplanes arrangements given in Figure 4.1.

These issues serve as justification for the use of zonotopic sets to over-approximate (4.1):

i) they preserve a good approximation of the original interdicted domain;

ii) have a clear complexity evaluation (in terms of total number of cells).

We caution again that a zonotope is centrally symmetric. Hence, not any convex shape can be rigorously approximated by an enclosing zonotope and this will be the main source of conservatism hereafter (and a potential direction of improvement).

We proceed to find the zonotopic over-approximations of the obstacles in Figure 4.1. The simulations were done for different variants of the generator seed G in (4.19) for both $d=2$ and $d=3$ dimensions:

$$G_{1,3} \in \left\{ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix} \right\}, \quad G_{4,5} \in \left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \right\}. \tag{4.38}$$

Maintaining the structure of Table 4.1, we delineate in Table 4.2 some parameters of interest. Thus, we indicate the total computing time corresponding to each considered method (t_{sol}), the relative modification of the number of cells ($\frac{\Delta\gamma(N)}{\gamma(N)}$), the volume of the over-approximation (V) and the relative error with respect to the volume of the polytopic obstacles ($\frac{\Delta V}{V}$).

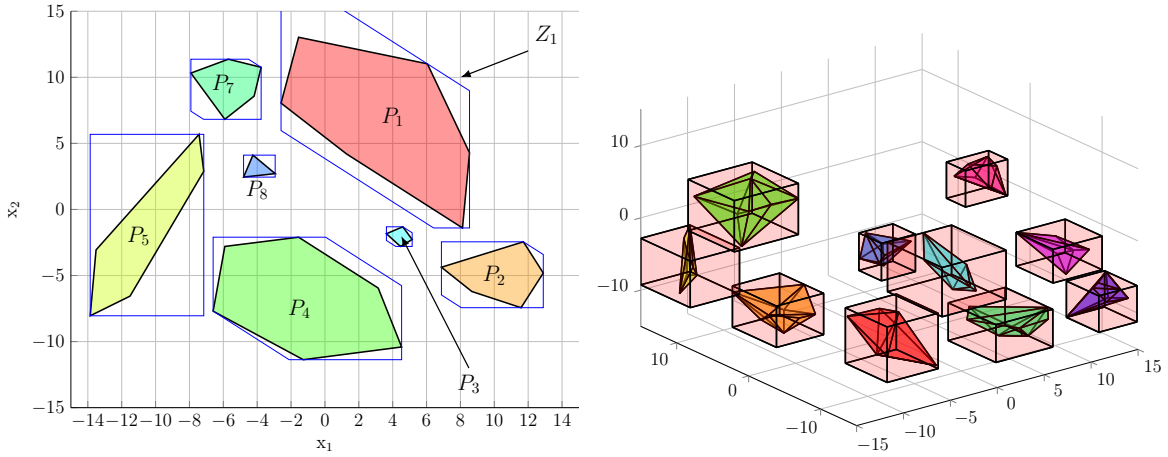
	Measure	Constraints	G	t_{sol}	$\#\mathbb{H}$	$\gamma^*(N)$	$\frac{\Delta\gamma(N)}{\gamma(N)}(\%)$	$t_{\gamma^*(N)}$	$\#\Sigma_{\mathbb{P}}$	V	$\frac{\Delta V}{V}(\%)$
$d=2$	$\|\delta\|_1$	(4.23)	G_1	8,13	42	505	20,53	9,53	197	**376,98**	71,7
			G_3	8,27	42	534	27,45	10,09	167	368,93	68,04
	$\|\delta\|_\infty$	(4.23)	G_1	8,19	42	441	5,25	8,19	374	897,92	308,98
			G_3	8,19	42	441	5,25	8,19	374	897,92	308,98
	Vol	(4.23)	G_1	9,40	42	510	21,72	9,66	199	**368,41**	67,8
			G_3	9,30	40	530	26,49	10,27	169	374,99	70,8
$d=3$	$\|\delta\|_1$	(4.23)	G_4	9,82	60	8400	-85,31	105,98	934	**1857,46**	323,56
			G_5	10,50	120	62480	9,26	1145,09	4952	2019,34	360,47
	$\|\delta\|_\infty$	(4.23)	G_4	9,71	60	8000	-86,01	100,22	1127	2623,3	498,19
			G_5	10,59	120	51396	-10,12	932,12	9432	5852,55	1234,56
	Vol	(4.23)	G_4	11,02	60	8400	-85,31	105,07	934	**1857,46**	323,56
			G_5	11,79	84	24528	-57,11	413,07	2218	1908,2	335,13

Table 4.2.: Performance of the zonotopic over-approximation techniques.

Note that $\|\cdot\|_1$ has a better behavior than $\|\cdot\|_\infty$ (i.e., its relative volume error is smaller) and that the volume criterion gives the most accurate approximation (at the price of a large computational effort). We conclude thus that $\|\cdot\|_1$ provides the best compromise between over-approximation error and computation time. Additionally, we observe an inverse proportionality between the volume of the approximation and number of hyperplanes (a factor which can be tweaked by a suitable choice of G).

The results are illustrated in Figure 4.2. While for $d=2$ the results are somewhat ambiguous (the decrease of the complexity is not marked), for the $d=3$ case the impact is substantial (even though the choice of the matrix G was empirical and not the result of an optimization process).

The main use (in our papers at least) for the zonotopic over-approximations described earlier was



(a) Volume with $G = G_3$.

(b) Volume with $G = G_4$.

Figure 4.2.: Zonotopic approximation for the multi-obstacle environment from Figure 4.1.

their use in characterizing the feasible domain in which an agent moves (in order to avoid obstacle collisions). The feasible domain is characterized through a mixed-integer formulation **[M1]**.

The constrained optimization problem (4.37a) has to be solved over a non-convex domain $\mathcal{X} \setminus \mathbb{P}$ in (4.37d). As stated in **[M1]** the use hyperplane arrangement with a mixed-integer formalism leads to effective control strategies. The quadratic MIP (4.37a) was solved via YALMIP (Löfberg 2004) using the CPLEX solver. We compare the performances of the control strategy corresponding to a polytopic representation (Figure 4.1) to a zonotopic over-approximation (we assume no overlaps). We depict in Figure 4.3 the agent motion having the same initial and final position in the both topologies. In Table 4.3 we delineate some noteworthy computational characteristics: N_{goal} - the number of steps to attain a neighborhood of the final point, t_{goal} - the total time to compute the trajectory, and t_{worst} - the maximum time to solve (4.37a). We note that for the $d = 2$ case the computational performances are quasi-similar for both topologies. We have a longer trajectory for the zonotopic one, because of the additional "obstacle" $P_j \setminus \mathcal{Z}(G\Delta_j, c_j)$ which becomes an interdicted region. However, t_{goal} is smaller due to the symmetry properties while t_{worst} has a similar value. The last aspect is directly caused by the values presented in Table 4.2 - the number of cells. Furthermore, for $d = 3$ the differences are noticeable, having a decrease of 70% for t_{goal}, and of 50% for t_{worst} in the zonotopes framework.

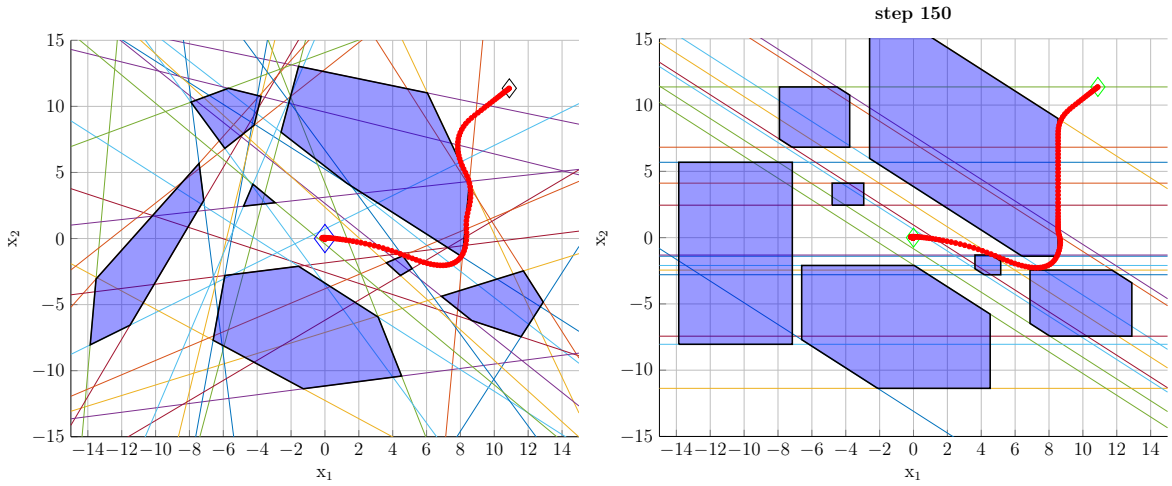
	Topology	N_{goal}	t_{goal}(sec)	t_{worst} (sec)
$d = 2$	$\mathbb{P}$	143	11.64	0.22
	$\mathbb{Z}$	146	10.07	0.18
$d = 3$	$\mathbb{P}$	98	83.87	0.81
	$\mathbb{Z}$	132	57.07	0.42

Table 4.3.: Simulation results for (4.37a) with $N_p = 10$, $P = 10I_{2d}$, $Q = I_{2d}$, $R = I_d$.

In contrast to the MIP approach where the discrete decisions are encoded in a mathematical formalism, the graph-based approaches reduce these decisions to the search of the shortest path between nodes in a graph. The PRM (probabilistic roadmaps) is an useful method if an awareness map of the environment is available (Karaman and Frazzoli 2011). Hence, a comparison with the MIP-based approach is pertinent.

Remark 4.4. An accentuation of the differences can be done either by increasing the prediction horizon N_p (at the expense of higher computational effort) or by optimized selection of the common "seed" G.

Employing the classical PRM, we randomly select a number of samples within the non-convex feasible domain and we connect them based on a visibility (Latombe 2012) criterion (if there is a line



(a) Polytopes

(b) Zonotopes

(c) Comparison MIP/PRM

Figure 4.3.: Illustration of obstacle avoidance for $d = 2$.

connecting the points without intersecting any obstacle, then the points are visible from one another). Having this visibility graph (offline computed), we link the start and the final position to the nearest graph nodes and find the shortest path through the graph, e.g., using Dijkstra's algorithm (Karaman and Frazzoli 2011). To follow the path (the black lines from Figure 4.3c), we opted to use MPC, (4.37a) without the constraints (4.37d).

	Approach	N_{goal}	t_{goal}(sec)	ℓ_t(m)
$d = 2$	MIP ($\mathbb{Z}$)	146	10.07	21.36
	PRM	172	9.81	24.41
$d = 3$	MIP ($\mathbb{Z}$)	132	57.07	25.57
	PRM	192	12.15	29.96

Table 4.4.: Results for the agent trajectories in Figure 4.3c.

As noted from Table 4.4, the trajectory length (ℓ_t) obtained with PRM is longer than the one resulting from the MIP ($\mathbb{Z}$) formulation. As well we observe that t_{goal} corresponding to PRM is independent of the value of d, whereas the zonotopic representation is strongly impacted by the space dimension. Within these two examples, the MIP-based approach performs better in the sense of

number of steps and trajectory length. Nevertheless, the choice of the approach always needs an application-dependent analysis.

4.4.2. Corridor design

Recalling the obstacle collection $\mathbb{P}$ defined in (4.1) we introduce several ancillay notions.

Definition 4.3. *Given the obstacles $\mathbb{P}$, a corridor between two points $x_0, x_f \in \text{int}(\mathcal{C}_{\mathbb{X}}(\mathbb{P}))$ is enabled by the existence of two continuous functions:*

$$\gamma : [0,1] \rightarrow \mathcal{C}_{\mathbb{X}}(\mathbb{P}) \tag{4.39}$$
$$\rho : [0,1] \rightarrow \mathbb{R}_{>0} \tag{4.40}$$

satisfying

$$\gamma(0) = x_0,\ \gamma(1) = x_f \tag{4.41}$$
$$\gamma(\theta) \oplus \mathbb{B}_{0,\rho(\theta)} \subset \mathcal{C}_{\mathbb{X}}(\mathbb{P}), \forall \theta \in [0,1] \tag{4.42}$$

Based on (4.39)-(4.42) the corridor is defined as:

$$\Pi = \{x \in \mathbb{R}^d :\ \exists \theta \in [0,1]\ s.t.\ x \in \gamma(\theta) \oplus \mathbb{B}_{0,\rho(\theta)}\}. \tag{4.43}$$

In this context, the objective is the construction of corridors in a *cluttered environment*.

Remark 4.5. Corridor construction encompasses classical path planning. Indeed, the corridor can be understood as a compact family of feasible paths between the initial and final point. Obviously, the selection of a path within a corridor is conceptually simpler in comparison with the direct search of a path avoiding the obstacles (provided that a corridor has already been found). Additionally, given a path within a corridor, the distance to the boundary of the corridor represents a robustness margin for the trajectory. ⧫

The problem of navigation in cluttered environments $\mathbb{X}$ can be divided in three sub-problems:

P1) Starting from the set of forbidden regions $\mathbb{P}$, determine a partition of the *cluttered environment* around them.

P2) Considering any two points $x_0, x_f \in \text{int}(\mathcal{C}_{\mathbb{X}}(\mathbb{P}))$, construct a corridor linking them or provide a certificate of infeasibility.

P3) Given a non-empty corridor, select/generate a continuous trajectory $\tau : [t_0, t_f] \rightarrow \Pi$ guaranteeing collision avoidance, i.e. $\tau(\theta) \cap \mathbb{P} = \emptyset, \forall \theta \in [0,1]$.

The notion of distance in this context is related to the characterization of the nearest obstacles and indirectly leads to the partitioning of the *cluttered environment* according to the distribution of obstacles.

Definition 4.4. *A family of sets $\{X_i\}_{i \in \mathcal{I}}$ verifying:*

i) $\mathbb{X} = \bigcup_{i=1}^{N_o} X_i$,

ii) $\text{int}(X_i) \bigcap \text{int}(X_j) = \emptyset, \forall i \neq j \in \mathcal{I}$,

iii) $P_i \subset \text{int}(X_i), \forall i \in \{1 \ldots, N_o\}$

is called a partition of $\mathbb{X}$ induced by the obstacles $\mathbb{P}$. ⧫

Definition 4.5. *If the sets $\mathbb{X}$ and $X_i, \forall i$ in Definition 4.4 are polyhedral, then $\mathbb{X} = \bigcup_{i=1}^{N_o} X_i$ is called a polyhedral partition.* ⧫

The partitioning introduced in Definition 4.5 induces a graph structure which allows to compute an obstacle-avoiding path.

Definition 4.6. *A weighted graph* $\Gamma(\mathcal{N}, \mathcal{E}, f)$ *is defined by the triple* $(\mathcal{N}, \mathcal{E}, f)$ *with* $\mathcal{N}$ *the set of nodes,* $\mathcal{E}$ *the set of edges, and* $f : \mathcal{E} \to \mathbb{R}$ *a function which associates to each edge a real positive value (its weight).*

Remark 4.6. A sometimes encountered variation is for the weight function $f(.)$ to associate weights to the nodes ($f : \mathcal{N} \to \mathbb{R}$), not to the edges, as is the case in Definition 4.6. Consequently, the classical shortest-path algorithms are slightly modified but can still be applied. ♦

Having the partition $\{X_i\}_{i=1:N_o}$ of the workspace $\mathbb{X}$, the goal is to construct a graph in order to generate feasible paths through $\mathbb{X}$. Therefore, we have to select the nodes, the edges and the associated weights from the constructive parameters of the compact sets X_i (vertices and faces). In Table 4.5 we delineate the existing possibilities for selecting graph components. While for the $d = 2$ the differences among the alternatives are negligible, for higher dimensions the choice of the graph representation may have a significant effect on the characteristics of the corridor (4.43).

	Domain of f	$\mathcal{N}$	$\mathcal{E}$
1	$\mathcal{E}$	$\mathcal{F}^0(X_i)$	$\mathcal{F}^1(X_i)$
2		$\mathcal{F}^{d-2}(X_i)$	$\mathcal{F}^{d-1}(X_i)$
3	$\mathcal{N}$	$\mathcal{F}^1(X_i)$	$\mathcal{F}^0(X_i)$
4		$\mathcal{F}^{d-1}(X_i)$	$\mathcal{F}^{d-2}(X_i)$

Table 4.5.: Alternative selection of graph triple $(\mathcal{N}, \mathcal{E}, f)$.

Proposition 4.2. $\Gamma\left(\{\mathcal{F}^{d-2}(X_i)\}_{X_i \in \mathbb{X}}, \{\mathcal{F}^{d-1}(X_i)\}_{X_i \in \mathbb{X}}, f\right)$ *is a connected planar graph.*

Proof. As $\{X_i\}_{i=1:N_o}$ is a partition of $\mathbb{X}$ satisfying ii) from Definition 4.4, there exists no intersection among the facets of regions X_i ($\mathcal{F}^{d-1}(X_i)$), except the faces of dimension $d-2$, and, by consequence, among edges of the graph (i.e. Γ is planar). Also, ii) from Definition 4.4 means that for any pair of "$d-2$"-faces from those of partition $\{X_i\}_{i=1:N_o}$ there exists a sequence of facets which connect these two faces. Hence, the existence of a path between any two nodes through the graph is guaranteed (i.e. Γ is connected). □

To illustrate, we present a constructive method for the graph $\Gamma_1(\mathcal{N}_1, \mathcal{E}_1, f_1)$, in the case $d = 2$, where:

- the nodes are the vertices of the polyhedral regions X_i:

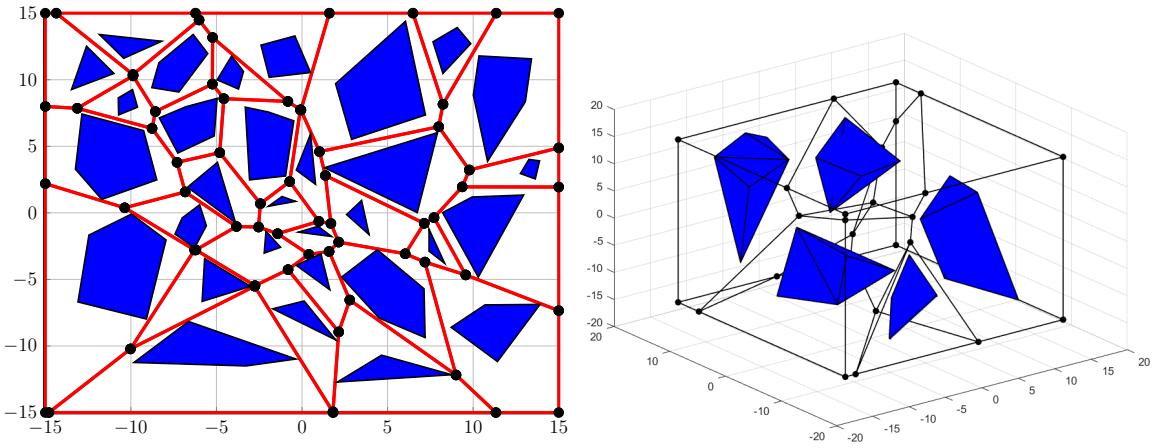
$$\mathcal{N}_1 = \bigcup_{i=1}^{N_o} \mathcal{F}^0(X_i) = \bigcup_{i=1}^{N_o} \mathcal{V}(X_i);$$

- the edges are the facets of the partition regions X_i (i.e., their support hyperplanes):

$$\mathcal{E}_1 = \mathcal{F}^1(X_i);$$

- the function f_1 gives the Euclidean distance between the incident nodes of the edge.

We depict in Figure 4.4a, the graph corresponding to the polyhedral partition from Figure 4.1a and in Figure 4.4b, the graph associated with the 3D partitioning shown in Figure 4.1b. Note that the partitioning $\{\mathcal{X}_i\}$ is obtained through a lifting procedure detailed in, e.g., (Ioan, Olaru, **Prodan**, Stoican, and S.-I. Niculescu 2019a).



(a) $d = 2$ (b) $d = 3$

Figure 4.4.: Graph $\Gamma(\mathcal{N}, \mathcal{E}, f)$.

To determine a corridor as in (4.43), the graph Γ needs to be slightly modified. Specifically, we have to add the nodes/edges corresponding to the start and final points $x_0, x_f \in \text{int}(\mathcal{C}_{\mathbb{X}}(\mathbb{P}))$. Usually, this augmentation of the graph involves some auxiliary components as well. For instance, in the 2D case of Γ_1 a first step consist in finding the closest edge such that the connection to it does not intersect any obstacles. Therefore, we compute the closest projection on the facets $\mathcal{F}^1(X_i)$ which satisfies (4.44b):

$$(j^{\perp}, x^{\perp})(x_0) = \arg\min_{j \in \mathcal{I}_{f_i}} \min_{x \in \mathcal{F}^1_j(X_i)} \|x - x_i\| \tag{4.44a}$$

$$\text{s.t.} \quad \alpha x + (1-\alpha)x_0 \notin P_i, \forall \alpha \in [0,1], \tag{4.44b}$$

$$x_0 \in X_i, \tag{4.44c}$$

where $\mathcal{I}_{f_i} = 1, \ldots, N_{f_i}$ with N_{f_i} the number of facets of X_i. Similarly, we obtain $(j^{\perp}, x^{\perp})(x_f)$. These two auxiliary nodes are added to the graph by linking them with the incident nodes of the containing edge and removing this edge. A new graph $\tilde{\Gamma}_1(x_0, x_f)$ which preserves the properties of Γ_1 is thus obtained. A graph search algorithm (e.g. Dijkstra's Algorithm (Karaman and Frazzoli 2011)) is employed and the shortest path between the nodes induced by x_0 and x_f is obtained.

Remark 4.7. For further use, we denote the shortest path through the graph between x_i and x_f as $\text{Path}(x_0, x_f) = (\bar{x}_0 = x_i, \bar{x}_1, \ldots, \bar{x}_n, \bar{x}_{n+1} = x_f)$. This represents an ordered set of points where no segment defined by a pair of consecutive points cuts any of the obstacles. It is not a path in the sense stated in problem P3), but is a sufficient condition for the existence of a corridor (4.43). ♦

Algorithm 1: Continuous path generation

Require: $\tilde{\Gamma}(x_0, x_f)$

Ensure: a path $\pi : [0,1] \to \mathbb{X}$ with $\pi(\theta) \cap \mathbb{P} = \emptyset, \forall \theta$

1: Find $\text{Path}(x_0, x_f)$ using Proposition 4.2.

2: Determine $\gamma(\theta)$ from Definition 4.3 for $\text{Path}(x_0, x_f)$.

3: Find $\rho(\theta)$ such that:

$$\rho(\theta) \leq \min_{P_i \in \mathbb{P}} d_H(P_i, \gamma(\theta)), \ \forall \theta \in [0,1]; \tag{4.45}$$

4: Find a path $\pi(\theta)$ inside the corridor characterized by $\gamma(\theta), \rho(\theta)$ as:

$$\pi(\theta) = \arg\min_{\nu} \mathcal{C}(\nu) \text{ s.t } \nu(\theta) \in \gamma(\theta) \oplus \mathbb{B}_{0,\rho(\theta)}, \ \forall \theta \in [0,1].$$

Each step of Algorithm 1 represents a further refinement along the trajectory generation procedure.

Step 1 provides the graph structure of the workspace, from which, in Step 2, $\gamma(\theta)$, a PWA (continuous) function is given (under the construction proposed in this section: a collection of edges). To these, in Step 3 is attached, $\rho(\theta)$, a continuous *width* function which provides a measure of the acceptable deviation from the nominal $\gamma(\theta)$. Step 4 replaces the feasible solution $\gamma(\theta)$ by an *optimization-based* selection $\pi(\theta)$. The cost $\mathcal{C}(\nu)$ is often taken as the path length but may be replaced/augmented by a cost describing energy/performance/smoothness.

Remark 4.8. Solving a continuous optimization problem, as required by Steps 3 and 4, is difficult if not downright infeasible. Standard approaches are to sample the continuous variables (θ in our case) or to project onto a basis function and reformulate the problem in terms of the associated weights. ♦

Proposition 4.3. *Any polyhedral partition* $\{X_i\}_{i=1:N_o}$ *provides a corridor (7) for a given pair* $x_0, x_f \in \text{int}(\mathcal{C}_{\mathbb{X}}(\mathbb{P}))$.

Proof. As Proposition 4.2 states, there exists a connected graph Γ induced by the partition $\{X_i\}_{i=1:N_o}$. By connecting any two points to the graph the connectivity is preserved. Thus, by using a search algorithm for the augmented graph we identify a continuous piecewise affine function γ generated by the edges composing $\text{Path}(x_0, x_f)$. Moreover, we consider ρ as a PW constant function defined for each edge of $\text{Path}(x_0, x_f)$ as the minimum Hausdorff distance between the edge and the obstacles around it. The set Π, described by functions γ and ρ is a corridor in the sense of (4.43). □

For illustration purposes we revisit the obstacle collection from Figure 4.4a to which we apply Algorithm 1.

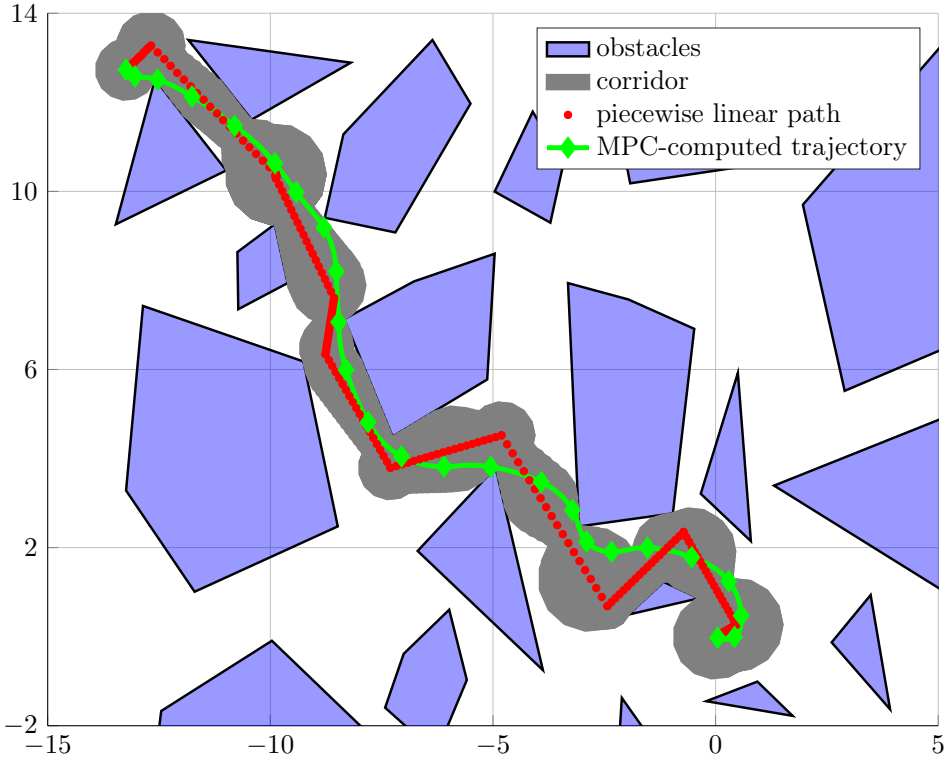


Figure 4.5.: The shortest path $\text{Path}(x_i, x_f)$ and a feasible trajectory within the corridor.

First, we construct the associated graph (depicted in Figure 4.4a) and find a path γ, as shown in Figure 4.5. Next, we provide an approximation of the corridor width ρ (gray area in Figure 4.5 is the corridor, as defined in (4.43)). To compute the corridor width we sampled the continuous parameter θ and introduced it in (4.45). We sidestep Step 4 of the algorithm by choosing $\pi = \gamma$. This path is provided as reference to a standard path tracking mechanism which (green line with diamond markers)

is shown to respect the constraints (there is no intersection with the obstacles and the destination is successfully reached). For illustration of the ultimate path tracking task we considered a standard double integrator dynamic and applied an MPC (Model Predictive Control) strategy.

4.4.3. A relay-MPC implementation

This section analyzes the formulation of a MPC controller and its recursive feasibility for navigation in a corridor (by exploiting the existence of feasible corridors resulted from the partitioning of the environment, as carried in Subsection 4.4.2). First we recall some concepts and definitions and next we present the formulation associated to a single compact corridor (a segment from (4.43)). Then the extension to the entire corridor (4.43) is tackled.

Our aim is to obtain a collision-free trajectory inside a corridor Π_i for an agent described by the LTI dynamics (6.19). The constraints corresponding to the corridor $\Pi_i \subset \mathbb{R}^{d_1}$ are usually only on a subset of the state variables $\mathcal{X} \subset \mathbb{R}^d$ (without loss of generality we consider these to be the first d_1 elements):

$$\tilde{\Pi}_i = \{x \in \mathcal{X} \mid \begin{bmatrix} I_{d_1} & 0_{d-d_1+1} \end{bmatrix} x \in \Pi_i\}. \tag{4.46}$$

For the MPC problem we consider a standard formulation with a quadratic cost (similar with the one shown in (4.37)–(4.37d)) but enhanced through the explicit variation of the prediction horizon and the presence of the target $\bar{x}_i$:

$$\mathcal{J}(N_p, \bar{x}_i) = \left(\overbrace{\|x_{k+N_p|k} - \bar{x}_i\|^2_{\mathbf{P}}}^{\mathbf{V}_f(x_k,u_k)} + \underbrace{\sum_{l=1}^{N_p-1} \|x_{k+l|k} - \bar{x}_i\|^2_{\mathbf{Q}} + \sum_{l=1}^{N_p-1} \|\Delta u_{k+l|k}\|^2_{\mathbf{R}}}_{\mathbf{V}(x_k,u_k)} \right) \tag{4.47}$$

where N_p is the prediction horizon, the weight matrices $\mathbf{P}$ (terminal cost penalty), $\mathbf{Q}$ (output error penalty) and $\mathbf{R}$ (control action penalty) are positive semi-definite and of appropriate dimensions. For further implications, the value of $\mathbf{P}$ from the terminal cost $\mathbf{V}_f(x_k, u_k)$ is selected such that the function $V(x) = x^\top Px$ is a Lyapunov function for a pre-stabilizing state-feedback law $u = Kx$ with $K \in \mathbb{R}^{d \times d_u}$ in a vicinity of $\bar{x}_i$ (Mayne, Rawlings, Rao, and Scokaert 2000). As well, the values of $\mathbf{Q}$ and $\mathbf{R}$ from the cost per stage $\mathbf{V}(x_k, u_k)$ are chosen in order to enhance the tracking performances.

Therefore, the MPC problem to be solved at each time step throughout a corridor $\tilde{\Pi}_i$ is:

$$\mathcal{P}(\tilde{\Pi}_i, N_p, \mathcal{X}_f, \bar{x}_i, \mathcal{X}) : \min_{\mathbf{u}} \ \mathcal{J}(N_p, \bar{x}_i) \tag{4.48a}$$

$$\text{s.t. } x_{k+l+1|k} = Ax_{k+l|k} + Bu_{k+l|k}, \tag{4.48b}$$

$$u_{k+l|k} \in \mathcal{U}, \ \forall \ell = 1 : N_p - 1 \tag{4.48c}$$

$$x_{k+l|k} \in \tilde{\Pi}_i \tag{4.48d}$$

$$x_{k+N_p|k} \in \mathcal{X}_f, \tag{4.48e}$$

Having (4.48), the question to be addressed is *how to select the parameters* $\tilde{\Pi}_i, N_p, \mathcal{X}_f, \bar{x}_i, \mathcal{X}$ *in order to ensure the recursive feasibility.* The recursive feasibility is one of the fundamental challenges in MPC literature. Basically, it represents the property that for all initial feasible states and for all optimal sequences of control inputs the MPC optimization problem remains feasible (Mayne, Rawlings, Rao, and Scokaert 2000). (Löfberg 2012) offers a broad overview on this topics, especially from the computational perspective.

Essentially, the selection of the parameters in (4.48) can be viewed as an additional analysis step, which builds on the, so-called, backward reachable set (BRS).

Definition 4.7 (N-step BRS)**.** *The N-step (BRS) is the set of all states that can reach a final position* $\bar{x}_i$ *in* N *steps- associated to the system described by LTI dynamics* (4.37b)*:*

$$R_N^i = A^{-N}\left(\bar{x}_i \oplus \bigoplus_{j=0}^{N-1} -A^j BU\right). \tag{4.49}$$

Since we have "hard" constraints on state (the ones given by the corridor), we have to compute a modified, $\tilde{R}_N^i$, N-step BRS taking into account these constraints. Sets $\tilde{R}_N^i$ remain the same as long as they check the corridor constraints. From that point, the computation can be done iteratively as in Algorithm 2.

Algorithm 2: Computing N-step BRS for (4.37b) taking into account the state constraints $\tilde{\Pi}_i$

1: Find first $R_{N_{uc}}^i$ such that $R_{N_{uc}}^i \not\subset \tilde{\Pi}_i$
2: For $N < N_{uc}$: $\tilde{R}_N^i = R_N^i$
3: $\tilde{R}_{N_{uc}}^i = R_{N_{uc}}^i \cap \tilde{\Pi}_i$
4: For $N > N_{uc}$ the computation is iterative and relies on: $\tilde{R}_N^i = A^{-1}\left(\tilde{R}_{N-1}^i \oplus (-BU) \cap \tilde{\Pi}_i\right)$.

The N-step BRSs have an instrumental role in providing the parameters of the MPC problem (4.48). For instance, the theoretical minimal value of N_p is given by the condition $x_0 \in R_{N_p}^i$.

Proposition 4.4. *$\mathcal{P}(\Pi_i, N_p, \mathcal{X}_f, \bar{x}_i, \mathcal{X})$ is feasible for all feasible initial states if*

$$N_p \geq \arg\min_N \; N \text{ s.t. } x_0 \in \tilde{R}_N^i. \tag{4.50}$$

Proof. It is straightforward that $x_0 \in \tilde{R}_{N_p}^i$. That is, there exists at least one sequence of N_p inputs so that the predicted final state is exactly the reference $\bar{x}_i$. As well, the existence of a terminal cost as in (4.47) ensures the convergence. Moreover, since $\bar{x}_i \in \mathcal{X}_f$, it is obvious that (4.48e) holds. Therefore, the feasibility of $\mathcal{P}(\Pi_i, N_p, \mathcal{X}_f, \bar{x}_i, \mathcal{X})$ is guaranteed. □

As a consequence of Proposition 4.4, the recursive feasibility of a MPC controller based on $\mathcal{P}(\cdot)$ follows straightforwardly. Since the recursive feasibility of an MPC strategy based on (4.48) is guaranteed for certain parameters, we aim to exploit these degrees of freedom for the resolution of the problem P3). To this end, an intuitive solution is to iteratively compute the parameters ensuring feasibility for each corridor segment. Specifically, for each initial position ($\gamma_i(0)$) within the corridor segment, we compute the N-step BRS centered in x_f^i. The computational effort corresponding to this may be substantial relative to a consequent real-time implementation. To overcome this drawback, we can split the strategy since we are aware of the corridor characteristics.

1. (OFF-LINE) For each $\tilde{\Pi}_i$ and x_f^i
   - compute the BRSs (as inAlgorithm 2) and $N_{p_{min}}^i$:

$$N_{p_{min}}^i = \arg\min_N \; N \text{ s.t. } \tilde{\Pi}_i \subseteq \tilde{R}_N^i, \tag{4.51}$$

   - $\mathcal{X}_f^i = (\tilde{\Pi}_i \cap \tilde{\Pi}_{i+1})$ excepting the last segment, for which we have $\mathcal{X}_f^i = \tilde{R}_1^i$
2. (ON-LINE) Apply Algorithm 3

As its name suggests, the idea behind the "Relay MPC" strategy is to ensure the transitions from a segment of corridor ($\tilde{\Pi}_i$) to the next ($\tilde{\Pi}_{i+1}$). For that reason, we choose the terminal sets $\mathcal{X}_{f_i}$ as the

intersections between two consecutive segments.

Algorithm 3: "Relay MPC"

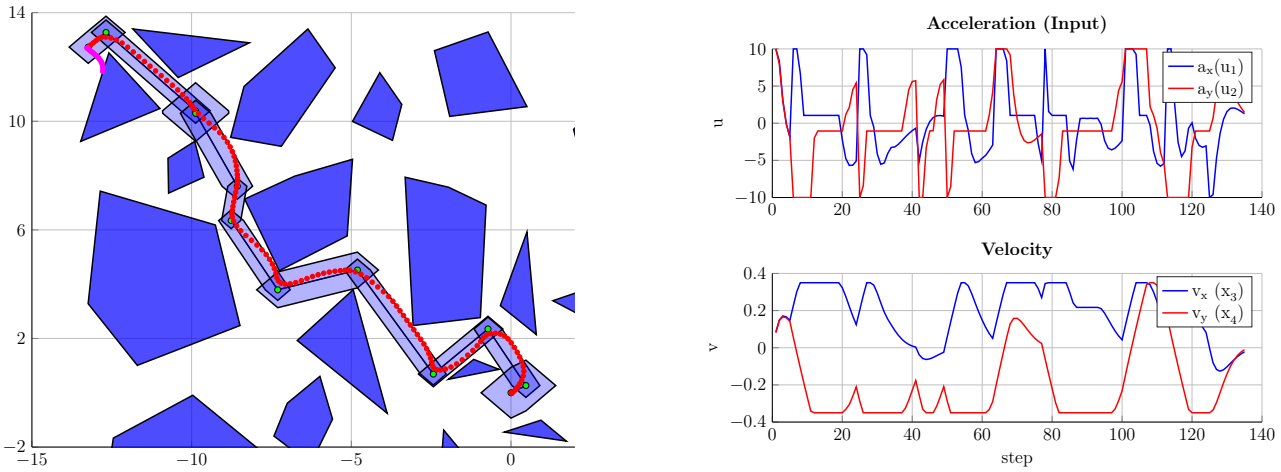
Require: $\Pi = \bigcup_{i=1}^{N_c} \Pi_i$, x_0, $N^i_{p_{min}}$, $\mathcal{X}^i_f$

1: **for each** $\tilde{\Pi}_i \subset \tilde{\Pi}$ **do**
2: $N_p = N^i_{p_{min}}$;
3: **repeat**
4: Apply MPC strategy solving $\mathcal{P}(\tilde{\Pi}_i, N_p, \mathcal{X}^i_f, \bar{x}_i, \mathcal{X})$
5: **until** $x_{k+1|k} \in \mathcal{X}^i_f$
6: $x_0 = x_{k+1|k}$
7: Update the parameters of $\mathcal{P}$
8: **end for**

Proposition 4.5. *If* $\text{Path}(x_0, x_f)$ *exists and the control law based on* $\mathcal{P}(\cdot)$ (4.48) *is recursively feasible then the convergence* $x_0 \to x_f$ *is guaranteed.*

Proof. The existence of $\text{Path}(x_0, x_f)$ leads to the construction of the corridor $\tilde{\Pi} = \bigcup_i \tilde{\Pi}_i$. The recursive feasibility and the selection of terminal sets as in (1) ensure the transition from $\tilde{\Pi}_i$ to $\tilde{\Pi}_{i+1}$. Moreover, for the last segment we have $\mathcal{X}^i_f = \tilde{R}^i_1$ and the initial state in $\tilde{R}_{N_p}$ which directly leads to x_f being reachable, and, by consequence, it proves the convergence of the scheme. □

For illustration purposes, using dynamics (6.19) we revisit the obstacle collection depicted in Figure 4.1a to which we apply Algorithm 3. As well, in Figure 4.6b the values of the acceleration and velocity are plotted along the simulation horizon.



(a) path through corridor

(b) input and velocity

Figure 4.6.: Algorithm 3 over the example in Figure 4.1a

As a side remark, for the same initial and final position an MIP based MPC strategy can be employed, considering the entire set of obstacles and imposing a large prediction horizon ($N_p = 40$). Despite the large prediction horizon, the resulting trajectory does not converge to the final position, the agent remaining on the boundary of one of the obstacles (the pink trajectory in Figure 4.6a).

Regarding the offline part, as stated above, the computational effort is substantial, e.g., for the trajectory in Figure 4.6a is around 3 minutes, but this can be improved by replacing the polyhedral representation of the sets with a zonotopic one (Althoff 2015). Intuitively, there is a connection between the length of a corridor segment ($\Delta\gamma_i = \|\gamma_i(1) - \gamma_i(0)\|$) and the length of the minimal prediction horizon($N_{p_{min}}$). Therefore, we depict in Table 4.6 the values corresponding to (4.51).

i	1	2	3	5	8	10	11
$\|\Delta\gamma_i\|$	0.77	4.02	0.09	1.27	4.50	2.39	0.53
$N_{p_{min}}$	11	24	13	14	29	19	13

Table 4.6.: Illustration of the correlation between corridor length and prediction horizon size

Remark 4.9. In practice, whenever the length of a segment of the corridor is greater than a user-defined value, that segment can be split. In this way, one can manage the trajectory tracking with similar length of the prediction horizons in concordance with the available computational constraints.

To emphasize the importance of the corridor and how it reflects in a real-world application, we can assume that the dynamics (6.19) are the nominal ones and any disturbance impacts, in fact, the dimension of the corridor. That is, the function $\rho(\cdot)$, as in (4.43), becomes $\tilde{\rho}(\cdot) = \rho(\cdot) + \Delta\rho$. In Table 4.7 we delineate: N_{goal} - the number of steps to attain a neighborhood of the final point, t_{goal} - the total time to compute the trajectory and ℓ_t - the trajectory length. As it can be seen in Table 4.7, the computing time t_{goal} has similar values (slight modifications), while N_{goal} and ℓ_t increase with the values of $\Delta\rho$. This behavior is counter-intuitive, but it can be explained by the fact that the decreasing in terms of steps (or distance) gained along of the corridor are wasted with the maneuvers associated to changing the segments of the corridor. However, by moving all complex operations from on-line to off-line, we note that the computing time is kept reasonable, allowing a comparison with the classical sampled-based methods.

$\Delta\rho$	t_{goal}(sec)	N_{goal}	ℓ_t(m)
0	1,200	135	23.920
3e-3	1,219	138	24,068
7e-3	1,227	139	24,095
1e-2	1,231	139	24,089
5e-2	1.254	141	24,066
7e-2	1.277	142	24,059

Table 4.7.: Analysis of the scheme's performance under disturbance-induced corridor tightening

4.5. Conclusions and discussions

This chapter presented a geometric view for the collision avoidance problem using zonotopic over-approximations of the obstacles. We emphasized the benefits of choosing a particular family of sets (parametrized zonotopes with a common generator seed) regarding the complexity of a non-convex feasible domain representation. While the results hold for $\mathbb{R}^d$ we detailed the cases $d = 2, 3$ as these have relevance for the subsequent motion planning procedures.

Furthermore, we presented a constructive solution for the generation of collision-free trajectories between two points in an environment containing multiple obstacles in a d-dimensional space. First we considered a mixed-integer implementation to solve at each step the non-convex optimization problem (within a simplified environment characterized by zonotopic over-approximation). Second, we proposed a corridor-based approach which restricts the agent's movement but greatly simplifies the problem complexity. The corridors themselves are the result of a graph-based procedure and the trajectory tracking within the corridors is the result of a relay-MPC which employs reachable sets to guarantee recursive feasibility.

In general, the use of mixed-integer programming in motion planning is a popular approach but this is not to say that there are not significant roadblocks. The use of distributed control strategies, imperfect knowledge of the environment (obstacles awareness, information from neighboring agents

and the like) are difficult to integrate within a mixed-integer formulation. Promising approaches aim to combine the theoretical notions (mixed-integer programming, distributed MPC, etc.) within a learning procedure which aims to reduce computation time and provide near-optimal solutions, (Schoellig, F. L. Mueller, and D'andrea 2012), (Gros and Zanon 2020).

5. A combined use of potential field constructions and NMPC for surface vehicles

This chapter deals with the motion planning problem for surface vehicles in the presence of varying environmental disturbances. The goal is to steer the vehicles from an initial to a final destination while ensuring connectivity and avoiding collision with fixed and moving obstacles. The novelty lies in the coherent combination among various ingredients for path generation and potential field constructions for collision avoidance and connectivity maintenance, all gathered in a distributed NMPC (Nonlinear Model Predictive Control) framework: i) LOS (Line-of-Sight) guidance and RRT* (optimal Rapidly-exploring Random Tree) algorithms are employed for generating a collision-free path considering static surroundings; ii) on-off barrier functions activate the proposed potential field components which are in the view range of the agents, hence ensuring safe navigation in a dynamic coastal environment with a low computation demand; iii) a NDO (Nonlinear Disturbance Observer) is integrated in the NMPC design to compensate the environmental disturbances. The proposed algorithm is validated through simulations and comparisons carried out over a benchmark for USVs (Unmanned Surface Vehicles) safe navigation in the Trondheim fjord, Norway.

The chapter is based on the following papers:

J14 Tran, N., **I. Prodan**, E. Grøtli, and L. Lefevre. "Safe navigation in a coastal environment of multiple surface vehicles under uncertainties: a combined use of potential field constructions and NMPC". in: *Ocean Engineering (**accepted**)*, pp. 1–32, 2020.

C43 Tran, N., **I. Prodan**, E. Grøtli, and L. Lefevre. "Distributed nonliner optimization-based control for safe navigation in a coastal environment", pp. 588–695. In: *Proceedings of the 26th IEEE European Control Conference (ECC'19)*. 2019. Naples, Italy.

C41 Tran, N., **I. Prodan**, E. Grøtli, and L. Lefevre. "Potential-Field Constructions in an MPC Framework: Application for Safe Navigation in a Variable Coastal Environment", pp. 51–58. In: *Proceedings of the 6th IFAC Conference on Nonlinear Model Predictive Control (NMPC'18)*. 2018. Madison, Wisconsin, USA.

C31 Tran, N. Q. H., **I. Prodan**, and L. Lefèvre. "Nonlinear optimization for multi-agent motion planning in a multi-obstacle environment", pp. 488–493. In: *Proceedings of the 21st IEEE International Conference on System Theory, Control and Computing (ICSTCC'17)*. 2017.

5.1. Brief literature review

USVs (Unmanned Surface Vehicles) are used to perform a specific mission with high performance and low costs (Conte et al. 2018; Pang, J. Wang, J. Liu, and Yi 2018). There are, however, some critical issues which need to be addressed for achieving the best performance. Some examples are: generating collision-free paths with static (e.g., islands, shoreline or ship anchoring) and moving obstacles (e.g., other ships) while complying with the International Regulations for Preventing Collisions at Sea (COLREGS) (Commandant 1999) for both manned and unmanned surface vehicles; guaranteeing connectivity among the ships for information exchange in the course of performing tasks and rejecting the external disturbance from the ocean which may highly affect their performance. The collision avoidance issue, formulated as non-convex constraints included in the control optimization problem, is usually addressed through:

i) the use of MIP (Mixed-Integer Programming) where the constraints are explicitly taken into account but with the cost of a high computation time (Cafieri and Omheni 2017), **[M1]**. Some maritime applications using MIP are, for example: determining efficient schedules for USVs/UUVs (Unmanned Underwater Vehicles) in performing a mine countermeasure mission (Bays, Tatum, Cofer, and Perkins 2015), reconfigurable USVs (Gheneti 2019);

ii) the use of indirect methods based on potential field constructions which show good computational performances (K. Shibata, N. Shibata, Nonaka, and Sekiguchi 2018), **C43** but where the local minima issue is still a shortcoming of these approaches. (Blindheim, Gros, and Johansen 2020; S. Wang, Fu, Y. Wang, and L. Zhao 2019; Chao, Feng, Qing, and Shuwu 2017; Lyu and Y. Yin 2017) show the effectiveness in dealing with obstacle and collision avoidance when APF (Artificial Potential Field) is used to establish feasible paths for USVs.

COLREGS-compliance for safe maritime navigation is another important issue and various approaches have been tested in the literature: velocity obstacle (Kuwata, Wolf, Zarzhitsky, and Huntsberger 2011), rapidly-exploring random tree (Chiang and Tapia 2018) or model predictive control, binary variables to choose scenarios with multi-moving obstacles with erratic motion (Johansen, Perez, and Cristofaro 2016), using a rolling horizon optimization approach to find optimal heading angles for collision avoidance with other ships (S. Li, J. Liu, R. R. Negenborn, and F. Ma 2019), slack variables to change the vessel's direction for specific situations (Mohamed, F. Martin, and Axel 2018) or elliptical ship domain (Abdelaal, Fränzle, and Hahn 2016), distributed coordination based on constraint optimization (L. Chen, Hopman, and R. R. Negenborn 2018), (S. Li, J. Liu, and R. R. Negenborn 2019). These approaches consider disc-like approximation for the static/dynamic obstacles, which might be too conservative, w.r.t the conditions in a real environment.

Furthermore, the connectivity maintenance issue ensures that the information exchange and sharing can be reliably realized by forwarding messages among agents while implementing and allocating tasks. Real applications include search and rescue, surveillance, cooperative transportation by autonomous vehicles at sea or on the ground, attitude alignment of clusters of satellites, air traffic management system, etc. There are some recent works which propose solutions for always ensuring that the agents lie within their communication range. We may classify them in two directions:

i) imposing hard constraints based on the Euclidean distance between agents, which must be less than their sensing radius as in (Filotheou, Nikou, and Dimarogonas 2018). This condition is handled in a decentralized NMPC framework. However, this approach is too strict and can lead to infeasibility in complex cases;

ii) using potential fields to ensure that an algebraic connectivity condition holds (i.e., the second-smallest eigenvalue of the Laplacian matrix is strictly greater than zero (H. Fang, Wei, J. Chen, and Xin 2017), (T. Li, R. Zhao, C. P. Chen, L. Fang, and C. Liu 2018)). Here, the problem of flocking of second-order multi-agent systems is addressed using potential hybrid fields to guarantee both, the connectivity and the anti-collision in a distributed control framework.

Finally, it is essential to propose robust controllers for the USVs which handle the uncertainties (Soloperto, Köhler, Müller, and Allgöwer 2019). Tube-based MPC is considered as one of the most popular approaches for coping with perturbations (Ke, Z. Li, and C. Yang 2018; Mayne, M. M. Seron, and Raković 2005). The general idea is to maintain the actual state within a safety region along the optimal state trajectory **C7**. Another approach is to design a disturbance observer to reject the disturbances leading to an improved system robustness (M. Wu, Gao, P. Yu, She, and Cao 2019; P. Yu, M. Wu, She, K.-Z. Liu, and Nakanishi 2018). The authors in (N. Wang, Sun, J. Yin, S.-F. Su, and Sharma 2018; Y. Yang, Du, H. Liu, C. Guo, and Abraham 2014; Do 2010) employ a nonlinear observer within a backstepping technique to design a trajectory tracking robust controller of the underactuated ship. However, the constraints coming from physical limitations are not considered. (Mohamed, F. Martin, and Axel 2018; Z. Liu, Geng, and J. Zhang 2017) deal with the obstacle and collision avoidance in the presence of external disturbances for a single ship which needs to track a given path.

5.2. Contributions

Motivated by all the observations above and the results in **C41**, this chapter introduces enhancements in the distributed motion planning for safe navigation of multiple surface vehicles in the presence of external disturbance in the Trondheim fjord complying with the COLREGS rules. Specifically, the contributions of this paper are:

1. generates LOS (Line-of-Sight) guidance system via a graph-based method, e.g., through the RRT* (optimal Rapidly-Exploring Random Tree) algorithm;

2. consider on-off barrier functions which guarantee the necessary connectivity distance for information exchange among the agents as well as activate the associated repulsive potential for static and dynamic obstacles;

3. consider a NDO (Nonlinear Disturbance Observer) to reject the disturbances from the ocean that may lead to undesirable performance for the ships;

4. integrate the above ingredients in a distributed NDO-NMPC - based algorithm with a threefold purpose: i) track the RRT*- based feasible path through LOS guidance system, ii) activate the constraints in the view range[1] of the agent for on-line collision avoidance complying with the COLREGs rules and iii) exchange information for connectivity maintenance;

5. validation of the proposed algorithm through simulations over a real benchmark for the safe navigation of ships in the Trondheim fjord.

5.3. Geometric tools in a potential field framework

This paper is inspired by the practical application of collision-free motion planning of unmanned surface vessels traveling between harbors in the Trondheim fjord, Norway. Fig. 5.1 illustrates the operating region of the ships which need to navigate to the desired haven while avoiding the shore or small islands and other vessels (moving obstacles) while simultaneously complying with the COLREGS rules[2]:

- Rule 8 - Action to avoid collision: avoidance action must be applied timely, before other vessel approaches. Any alterations of course and/or speed must be large enough to clear the approaching vessels.

[1] As required by COLREGS, all ships shall maintain a proper radar lookout (has a view of up to several kilometers if there are no physical obstructions) to obtain early warning of risk of collision.

[2] http://astat.autonomous-ship.org/

- Rule 13 - Overtaking: The overtaking vessel can pass on either side and must keep out of the way of the vessel being overtaken. The vessel being overtaken must hold the course and speed until other vessel is past and well clear.
- Rule 14 - Head on situation: when two power-driven vessels are meeting on nearly reciprocal courses so as to involve risk of collision, then alter course to starboard so that each pass on the port side of each other.
- Rule 15 - Crossing situation: when two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her own starboard side shall keep out of the way.
- Rule 16 - Actions by give-way vessel: take early and substantial action to keep well clear.

As a benchmark we have used the Automatic Identification Systems (AIS) to provide real numerical data related to the position and velocity of ships, as they navigate between harbors.

In order to efficiently describe the non-convex feasible region for the dynamical (mobile) agent, we briefly recall here the system dynamics and various notions which involve polyhedral sets and repulsive potential constructions.

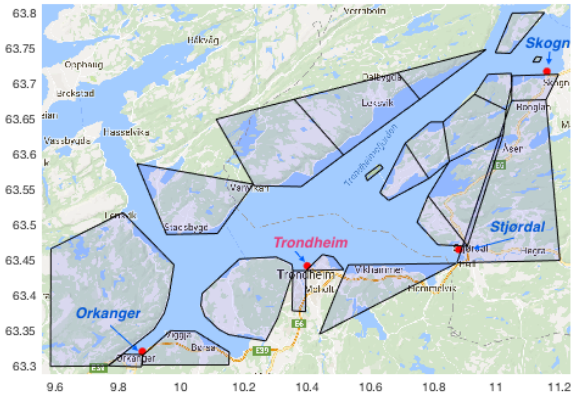


Figure 5.1.: Map of Trodheim obtained from real data: feasible space and considered forbidden cells as in (5.4).

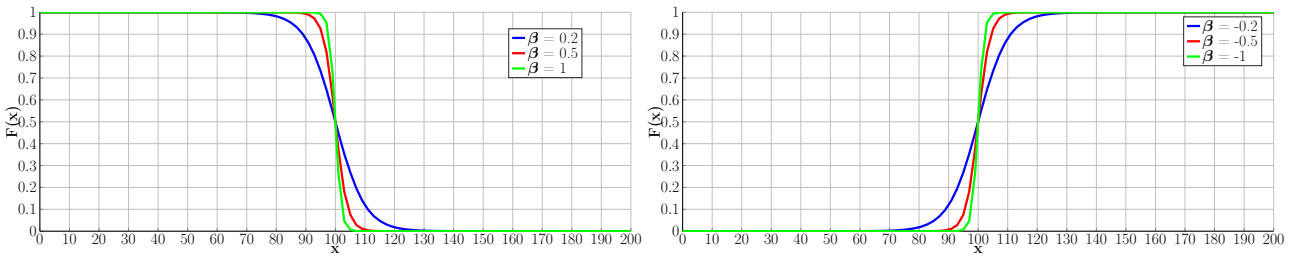
5.3.1. Multi-autonomous surface vehicles dynamics

Let us consider a set $\mathcal{V}$ including N agents, $\mathcal{V} = \{1, ..., N\}$, $N \geq 2$, operating in a two-dimensional work space. The nonlinear dynamics of agent $i \in \mathcal{V}$ under external disturbances (wind, wave and current) are described in (Fossen 2002):

$$\dot{x}_i = f_i(x_i(t), \mathbf{u}_i(t), w_i(t)) = \begin{cases} \dot{\eta}_i & = R_i(\psi_i)\nu_i, \\ M_i\dot{\nu}_i & = -C_i(\nu_i)\nu_i - D_i\nu_i + \mathbf{u}_i + w_i, \end{cases} \tag{5.1}$$

where the state vector, $\mathbf{x}_i = \begin{bmatrix}\eta_i & \nu_i\end{bmatrix}^\top \in \mathbb{R}^6$ includes the vector $\eta_i = \begin{bmatrix}p_i & \psi_i\end{bmatrix}^\top \in \mathbb{R}^3$ with $p_i = \begin{bmatrix}x_i & y_i\end{bmatrix}^\top$, the system position and ψ_i, the yaw angle in the inertial frame. It also includes vector $\nu_i = \begin{bmatrix}u_i & v_i & r_i\end{bmatrix}^\top \in \mathbb{R}^3$ describing the surge, sway and yaw rates. The input vector, $\mathbf{u}_i \in \mathbb{R}^3$, with $\mathbf{u}_i = \begin{bmatrix}\mathcal{T}_{u_i} & 0 & \mathcal{T}_{r_i}\end{bmatrix}^\top$ contains the surge thrust and rudder deflection. The addictive disturbance $w_i = \begin{bmatrix}w_{u_i} & 0 & w_{r_i}\end{bmatrix}^\top \in \mathbb{R}^3$ is not a pure Gaussian noise but rather the output of such a noise (after it passed through a nonlinear filter), and is bounded by $\|w_i\| \leq \varrho_i$ and accounts for the ocean environment [3]. Also, in (5.1), $R_i(\psi_i)$, M_i, $C_i(\nu_i)$ and $D_i \in \mathbb{R}^{3\times3}$ are the rotation, mass, Coriolis and damping

[3] I.e., wind, wave, ocean current affect control input of the ship.



(a) $\beta > 0$. (b) $\beta < 0$.

Figure 5.2.: On-off barrier function with different β as in (5.2).

matrices, respectively[4].

5.3.2. On-off repulsive potential field framework

Let us first introduce the following *Logistic Regression* function, which was first applied for population growth studies (Kyurkchiev and Markov 2016) and is commonly used in machine learning algorithms for two-class classification:

$$F(x) = \frac{L}{1 + e^{(x - x_o)\beta}}, \tag{5.2}$$

where x_o is the value of the sigmoid's midpoint, L is the curve's maximum value and β, the steepness of the curve. Note that a negative or positive value of β will determine the properties of the *on-off barrier function* which we will define hereinafter. Fig. 5.2 shows the on-off barrier function for $L = 1$, $x_0 = 100$ and varying β.

In this work we design on-off repulsive potential fileds as a combination between on-off barrier functions and repulsive potential fields for static and moving obstacles in order to activate and deactivate their influence in the agent's view range.

Assume that agent $i \in \mathcal{V}$ can perceive its neighbors (other agents) and the forbidden cells (fixed obstacles[5]) as well as transmit information to its neighbors within the range $\rho_i(p_i, r_\rho^i)$, where $r_\rho^i \in \mathbb{R} > 0$ is the radius of the ball centered in $p_i \in \mathbb{R}^2$, the current position of agent i.

For simplicity, the range $\rho_i(p_i, r_\rho^i)$ is considered hereinafter simultaneously as the *view range* (i.e., the zone where the ship can detect all the object) and the *communication range* (i.e., the area where the ships can exchange information with each other).

The safe distance between the agent and the fixed obstacles, $D_s^{i,\ell}$, is given by:

$$D_s^{i,\ell} = e_i \Gamma_\ell + \rho_i, \tag{5.3}$$

where e_i is a scaling parameter, Γ_ℓ is the radius of the circle defined from the Chebyshev center of the ℓ^{th} forbidden cell and ρ_i is the range of the ship. Moreover, the range $\rho_i(p_i, r_\rho^i)$ of the i^{th} agent is also considered as a safe distance of itself with respect to the other agents/moving obstacles (see Fig. 5.3).

[4]Note that in model (5.1), the side-slip is neglected as presented in (Eriksen, Breivik, Wilthil, Flåten, and Brekke 2019), therefore yaw and course are the same.

[5]The fixed obstacles such as islands and shorelines are a priori known from maps. The ship's position in relation to the obstacles is known using Global Navigation Satellite Systems.

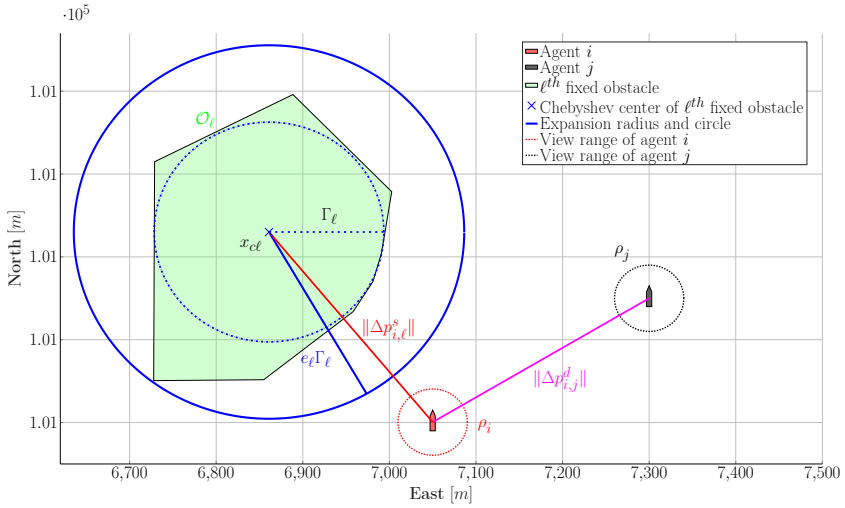


Figure 5.3.: Description of the safe distance of the i^{th} agent w.r.t. the ℓ^{th} cell and j^{th} agent.

Repulsive potential field of static obstacles

Let us define the static obstacles (islands, shore and the like) by a union of forbidden polytopic convex regions as:

$$\mathbb{O} = \bigcup_{\ell=1}^{N_{\text{cell}}} \mathcal{O}_\ell. \tag{5.4}$$

where $\mathcal{O}_\ell = \{p_i \in \mathbb{R}^2 \mid a_k^\ell p_i \leq b_k^\ell, k = 1, ..., n_h\}$, with $a_k^\ell \in \mathbb{R}^{1\times 2}$, $b_k^\ell \in \mathbb{R}$, n_h^ℓ is the number of half-spaces describing $\mathcal{O}_\ell$ and N_{cell} is the number of forbidden polytopic regions.

For each bounded polyhedral $\mathcal{O}_\ell$, we consider the piecewise linear function[6] as in (Camacho and Bordons 2004):

$$\gamma_\ell(p_i) = \sum_{k=1}^{n_h^\ell} (a_k^\ell p_i - b_k^\ell + |a_k^\ell p_i - b_k^\ell|). \tag{5.5}$$

Using (5.5), we define $\mathbb{S}$ as a union of repulsive potentials of the fixed obstacles as:

$$\mathbb{S} = \sum_{i=1}^{N_{\text{cell}}} S_\ell^{\text{fix}}(\gamma_\ell(p_i)), \tag{5.6}$$

with $S_\ell^{\text{fix}}(\gamma_\ell(p_i))$ given as:

$$S_\ell^{\text{fix}}(\gamma_\ell(p_i)) = \frac{c_{1\ell}}{(c_{2\ell} + \gamma_\ell(p_i))^2}, \tag{5.7}$$

where $c_{1\ell}$ and $c_{2\ell}$ are positive parameters representing the strength and effect ranges of repulsive potential.

Hence, the *on-off repulsive potential* of the fixed obstacle, Ω_{cell}, is considered as the repulsive potentials of the static obstacles described as in (5.6) will be activated/deactivated if the on-off barrier function from (5.2) is considered as follows:

$$\Omega_{\text{cell}} = \sum_{\ell=1}^{N_{\text{cell}}} F_{i,\ell}(\Delta p_{i,\ell}^s, D_s^{i,\ell}) S_\ell^{\text{fix}}(p_i), \tag{5.8}$$

[6]The function (5.5) is zero inside the convex set $\mathcal{O}_\ell$ defined as in (5.4) and non-zero with a value which grows piecewise linearly as the distance from the set $\mathcal{O}_\ell$ increases.

with $S_{\ell}^{\text{fix}}(p_i)$ given in (5.7), $F_{i,\ell}(.)$ defined as an on-off barrier function and based on the safety distance $(D_s^{i,\ell})$ between agent i and the forbidden obstacle ℓ:

$$F_{i,\ell}(.) = \frac{L}{1 + e^{\left(\|\Delta p_{i,\ell}^s\| - D_s^{i,\ell}\right)\beta}}, \tag{5.9}$$

Proposition 5.1. *If a set $\mathbb{S}$ of $N_{cell} \in \mathbb{N}$ representing static repulsive potential fields eq. (5.6) are only partially activated by the agent's view range through on-off barrier functions as defined in eq. (5.8), then a null-potential field[7] value which implies a local minima can be excluded.*

Proof: See **J14**.

Repulsive potential field of agents/moving obstacles

Hereinafter, we represent the moving obstacles and the safety regions around the agents as polytopic regions parametrized in function of their current position. Therefore, let us define $\mathcal{P}_{i\backslash\{j\}}$, $(i, j \in \mathcal{V}, i \neq j)$ as the safety region of agent i that agent j must avoid:

$$\mathcal{P}_{i\backslash\{j\}} = \{p_i, p_j \in \mathbb{R}^2 : a_m^i(p_j - p_i) \leq b_m^i, m = 1, ..., n_h^j\}, \tag{5.10}$$

where p_i, p_j are the current positions of agents i, j; $a_m^i \in \mathbb{R}^{1\times 2}$, $b_m^i \in \mathbb{R}$, n_h^i is the number of half-spaces describing $\mathcal{P}_{i\backslash\{j\}}$.

Next, we define a piecewise linear function similar to (5.5) for the safety region of agent i:

$$\theta_{i,j}(p_i, p_j) = \sum_{k=1}^{n_h^i} \left(a_m^i(p_j - p_i) - b_m^i + |a_m^i(p_j - p_i) - b_m^i| \right). \tag{5.11}$$

Similarly, using (5.11), we define the repulsive potentials which take into account the shape of the agent i's safety regions:

$$S_{i,j}^{\text{mov}}\left(\theta_{i,j}(p_i, p_j)\right) = \frac{c_{1j}}{(c_{2j} + \theta_{i,j}(p_i, p_j))^2}. \tag{5.12}$$

Consequently, in order to activate or deactivate the repulsive potentials of agents or moving obstacles established as in (5.12), the on-off barrier function from (5.2) will be integrated into the construction of the repulsive potential to define for the *on-off repulsive potential* of the moving obstacles:

$$\Omega_{\text{mov}} = \sum_{j=1, j\neq i}^{N} F_{i,j}(.) S_{i,j}^{\text{mov}}(p_i, p_j), \tag{5.13}$$

where $S_{i,j}^{\text{mov}}(p_i, p_j)$ is given in (5.12) and $F_{i,j}(\Delta p_{i,j}^d, \rho_i)$ defined as follows:

$$F_{i,j}(.) = \frac{L}{1 + e^{\left(\|\Delta p_{i,j}^d\| - \rho_i\right)\beta}}, \tag{5.14}$$

with ρ_i defined as i^{th} agent's view range.

Note that β in (5.9) and (5.14) is chosen positive to ensure that if the distance between agent and fixed/moving obstacles is greater than the safety distance between them, then the value of the on-off barrier function goes to zero thus inactivating the repulsive potential and, otherwise, and coupled with $L = 1$ it conserves the repulsive potential's value.

[7] Where the various potential components cancel each other.

5.3.3. Connectivity maintenance condition

We consider here that the information known to agents in a group (trajectories, velocities, and the like) can only be exchanged with those which are inside the communication range.

Let us consider the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes representing all agents and $\mathcal{E} = \{(i,j) \in \mathcal{V} \times \mathcal{V} : i \leftrightarrow j \}$ is the set of edges of the graph. The adjacency matrix $\mathcal{A} = [a_{ij}]$ indicates if a pair of nodes are adjacent or not in the graph hence satisfying the property:

$$a_{ij} = \begin{cases} 1, & \text{if } (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases} \tag{5.15}$$

Consequently, the condition of connectivity maintenance for information exchange among agent i and its neighbors ($\mathcal{N}_i$) is defined as follows:

$$\mathcal{N}_i = \{j \in \mathcal{V} \setminus \{i\} : \|\Delta p_{i,j}\| < \min(\rho_i, \rho_j)\}, \tag{5.16}$$

where, ρ_i and ρ_j are the view range of agents i and j in (5.3).

Next, for any edge between two nodes of set $\mathcal{N}_i$, we define an on-off barrier function as in (5.2) whose value is always zero within the interval $[0, \min(\rho_i, \rho_j))$:

$$M_{i,j}(p_i, p_j) = \frac{L_{ij}}{1 + e^{\left(\|\Delta p^d_{i,j}\| - \min(\rho_i, \rho_j)\right)\beta_{ij}}}. \tag{5.17}$$

Note that in this case, $\beta_{ij} < 0$ and L_{ij} has a high value such that the distance among the agents of $\mathcal{N}_i$ does not exceed the $\min(\rho_i, \rho_j)$.

Using the above construction we define a so called "*connectivity maintenance function*" of agent i with its neighbors $\mathcal{N}_i$:

$$\mathcal{M}_i(p_i, p_j) = \sum_{j \in \mathcal{N}_i} M_{i,j}(p_i, p_j), \tag{5.18}$$

5.3.4. Path generation with LOS guidance

Usually for the control part is easier to have a priori defined a path which can be then tracked on-line. In here, RRT* is applied for generating an off-line optimal collision-free path in the workspace for a known initial state (x_{init}) and set of fixed obstacles ($\mathbb{O}$). The general idea of the RRT* algorithm ((**pharpatara2015shortest**; LaValle 1998)) is to initialize a tree $\mathscr{T}$ from an initial vertex (x_{init}) and to bias growth towards unexplored regions of the state space randomly. In the process of exploring the tree, shortest paths are generated while verifying the collision-free condition with respect to the fixed obstacles after each iteration until reaching a new state close enough to the desired target. The feasible path is established based on the connection of shortest paths.

LOS algorithms have been used for autonomous ships navigation by (Fossen 2002). The feasible path obtained from RRT* is a set of waypoints (x_p, y_p) in the horizontal plane.

For simplicity, assuming that the along-track error, x_e, is zero. Thus, cross-track error (y_e) for the surface vehicle position as in Fig. 5.4 is given by:

$$\begin{bmatrix} 0 \\ y_e \end{bmatrix} = \begin{bmatrix} \cos(\delta_p(\alpha)) & -\sin(\delta_p(\alpha)) \\ \sin(\delta_p(\alpha)) & \cos(\delta_p(\alpha)) \end{bmatrix}^\top \begin{bmatrix} x - x_d(\alpha) \\ y - y_d(\alpha) \end{bmatrix}, \tag{5.19}$$

where $x_d(\alpha), y_d(\alpha)$ are parametrized by α as:

$$x_d(\alpha) = x_p + \alpha\cos(\delta_p(\alpha)), \tag{5.20a}$$

$$y_d(\alpha) = y_p + \alpha\sin(\delta_p(\alpha)), \tag{5.20b}$$

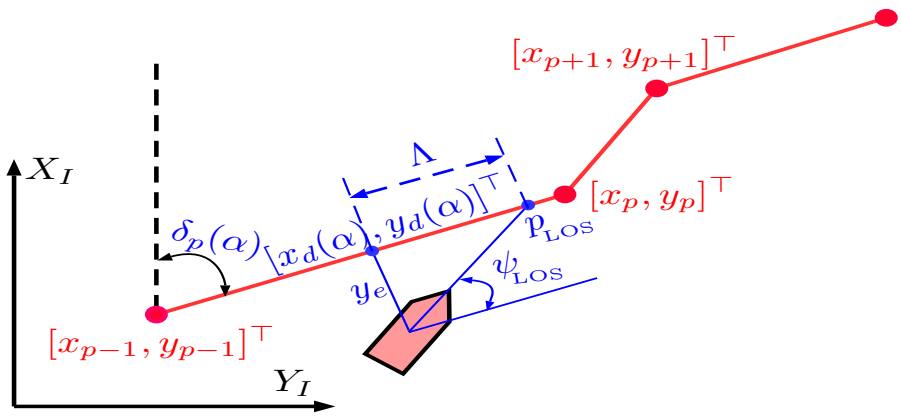


Figure 5.4.: LOS guidance for a feasible path.

where $\delta_p(\alpha)$ and (x_p, y_p) are the orientation and the initial position of the path initialized from waypoint p, with α parametrization variable is a scalar.

The LOS angle, ψ_{LOS} (as can be seen in Fig. 5.4), is calculated based on the cross-track error y_e and the lookahead distance Λ, expressed as:

$$\psi_{LOS} = -\frac{y_e}{\sqrt{y_e^2 + \Lambda^2}} \tag{5.21}$$

The angle, ψ_{LOS}[8] is considered as a reference state of the heading angle ψ of surface vehicles to ensures convergence of the ship's position to a feasible path generated by RRT*.

5.4. Motion planning for connectivity maintenance with COLREGS compliance in the presence of external disturbances

This section presents the motion planning algorithm for multi-surface vehicles connectivity maintenance. The motion planning strategy is illustrated via the block diagram in Fig. 5.5. From a map of the Trondheim fjord, the ground represented by static obstacles and the coordinates of the harbors can be acquired as seen in Fig. 5.1. A collision-free path, and LOS guidance system can then be generated off-line from the starting harbor to the desired destination based on a set of waypoints yielded by the RRT* algorithm. NDO-based NMPC, a feedback control scheme in which an optimization problem is solved on-line at each time step, will ensure collision avoidance for dynamic obstacles and robustness in the presence of the external disturbances. If there are no static or dynamic obstacles, the NDO-based NMPC will try to re-establish path following of the path obtained from RRT*.

5.4.1. NDO design for unknown time-variant disturbance

The unknown varying external disturbances (e.g., ocean currents) impact directly the control inputs of vessels leading to bad performance. Consequently, the nonlinear disturbance observer will provide an estimate of the disturbance, and its estimation is fed back to the NMPC solver. Subsequently, a disturbance observer is presented for a general nonlinear dynamic which will be later used for the estimation of the disturbance vector w_i affecting the i-th agent.

A fundamental idea of disturbance observer in NMPC scheme is to estimate the unknown disturbance w_i by an estimate $\hat{w}_i$ such that $|\hat{w}_i - w_i| \to 0$. In fact, the acceleration, $\dot{\nu}_i$, is not available in many robotic manipulators, and it is also difficult to construct the acceleration signal from the velocity

[8] Note that, since locally Λ corresponds to the inverse proportional gain (**breivik2008guidance**), the convergence to the path depends on the value of Λ. A low value means faster convergence than a larger one, but with a large overshoot.

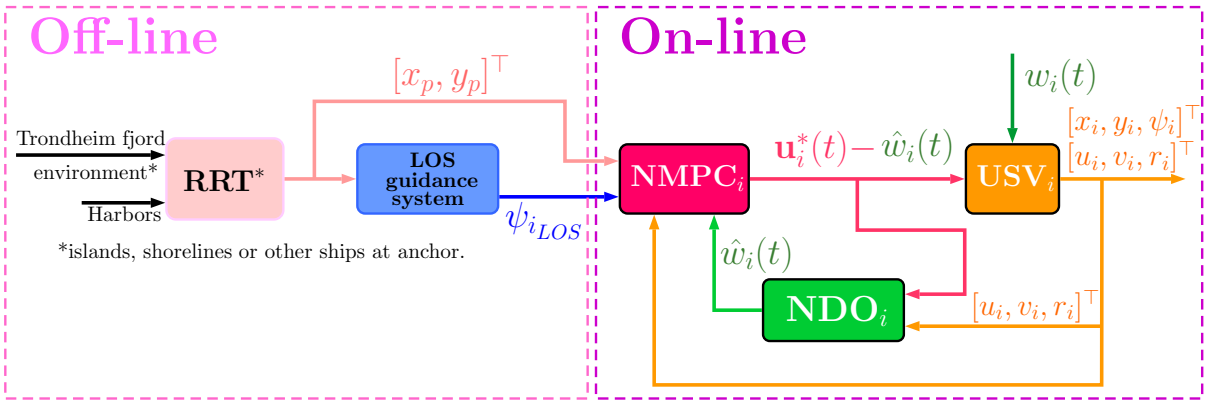


Figure 5.5.: Motion planning strategy.

signal by differentiation due to measurement noise. Therefore, we first need to define the $q_i = M_i\nu_i$ as the auxiliary variable and its estimation $\hat{q}_i$. Consequently, the estimated error, q_{ie} is described as below:

$$q_{ie} = \hat{q}_i - q_i. \tag{5.22}$$

From the (5.1), The time derivative of $\hat{q}_i$ can be rewritten as:

$$\dot{q}_i = -C_i(\nu_i)\nu_i - D_i(\nu_i)\nu_i + \mathbf{u}_i. \tag{5.23}$$

As the results of (Do 2010; W.-H. Chen, Ballance, P. J. Gawthrop, and O'Reilly 2000), the time derivative of auxiliary variable's estimation, $\hat{q}_i$ is defined as follows:

$$\dot{\hat{q}}_i = L(-C_i(\nu_i)\nu_i - D_i(\nu_i)\nu_i + \mathbf{u}_i - h(q_{ie})), \tag{5.24}$$

where $L = diag\{c_1, c_2, c_3\} > 0$, are the gains of the NDO, and $h(q_{ie}) \in \mathbb{R}^3$ is a design vector to be determined.

Consequently, the time derivative of estimated error between auxiliary variable, q_i and its estimation, $\hat{q}_i$ described as:

$$\begin{aligned} \dot{q}_{ie} &= \dot{\hat{q}}_i - \dot{q}_i \\ &= (L - \mathbf{I_3})[-C_i(\nu_i)\nu_i - D_i(\nu_i)\nu_i + \mathbf{u}_i] - Lh(q_{ie}), \end{aligned} \tag{5.25}$$

where $\mathbf{I}_3$ is the identity matrix with dimension related to the state space vector of ν_i.

To ensure that (5.25) converges to zero, (5.25) has to satisfy two conditions:

1. $L = \mathbf{I}_3$,
2. The designed function vector converges towards zero, i.e., $h(q_{ie}) \to 0$.

We propose the use of on-off barrier function with continuous nature in order to design $h(q_{ie})$ based on the estimated error, q_{ie} in (5.22). Hence, the design function, $h(q_{ie})$, is described as follows:

$$h(q_{ie}) = K_{1i}(q_{ie}) + K_{2i}|q_{ie}|\mathcal{J}(q_{ie}) + d_i\mathcal{J}_i(q_{ie}), \tag{5.26}$$

where $\mathcal{J}_i(q_{ie}) = \frac{e^{\beta_i^{NDO} M_i\nu_{ie}}}{\mathbf{1}+e^{\beta_i^{NDO} q_{ie}}} - \frac{e^{-\beta_i^{NDO} q_{ie}}}{\mathbf{1}+e^{-\beta_i^{NDO} q_{ie}}} \in \mathbb{R}^3$ is a vector whose dimension depends on vector $q_{ie} \in \mathbb{R}^3$; $K_{1i} = diag\{K_{1i}^1, K_{1i}^2, K_{1i}^3\}$, $K_{2i} = diag\{K_{2i}^1, K_{2i}^2, K_{2i}^3\}$ and $d_i = diag\{d_i^1, d_i^2, d_i^3\}$ positive semi-definite matrices.

Remark 5.1. In (Zhu, J. Ma, Z. Liu, and K. Liu 2017), the authors employed signum function $\left(\mathcal{J}_i(q_{ie}) = \frac{|q_{ie}|}{q_{ie}}\right)$ to design function $h(q_{ie})$. However, a discontinuity at zero is the main drawback

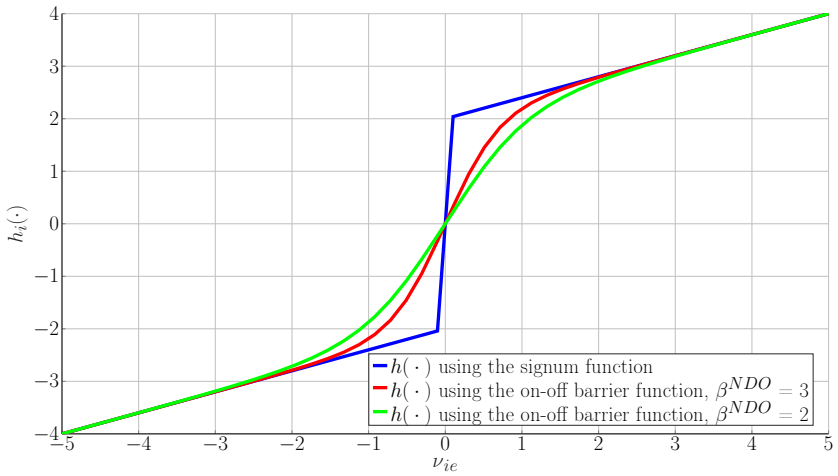


Figure 5.6.: The comparison of designed function $h_i(\cdot)$ (5.26) between on-off barrier function (with the various value of $\beta^{NDO} > 0$) and signum function.

which comes from the property of the signum function. Since design function $h(q_{ie})$ considers the estimation error of the real disturbance, a discontinuity at zero means that the convergence of this error toward zero will be indefinite.

Fig. 5.6 compares the proposed function $h_i(\cdot)$ with two different values for β^{NDO} with the original signum function as in (5.26).

With $h_i(\cdot)$ as in (5.26), (5.24) can be written as:

$$\begin{aligned}\dot{\hat{q}}_i = &- C_i(\nu_i)\nu_i - D_i(\nu_i)\nu_i + \mathbf{u}_i \\ &- K_{1i}(q_{ie}) - K_{2i}|q_{ie}|\mathcal{J}_i(q_{ie}) - d_i\mathcal{J}_i(q_{ie}).\end{aligned} \tag{5.27}$$

The estimated disturbance, $\hat{w}_i$ will be given by:

$$\begin{aligned}\hat{w}_i &= \dot{\hat{q}}_i + C_i(\nu_i)\nu_i + D_i(\nu_i)\nu_i - \mathbf{u}_i \\ &= -K_{1i}(q_{ie}) - K_{2i}|q_{ie}|\mathcal{J}(q_{ie}) - d_i\mathcal{J}(q_{ie}) = h(q_{ie}).\end{aligned} \tag{5.28}$$

The following proposition is introduced to highlight the convergence of estimated disturbances toward the real disturbances.

Proposition 5.2. *Consider the additive disturbances in* (5.1)*. Given the NDO as in Eq.* (5.27)-(5.28)*, its estimate* $\hat{w}$ *can asymptotically track the input disturbances* w*. In other words,* $\|\hat{w} - w\|_\infty \to 0$*.*

Proof. For the detailed proof see **J14**. □

5.4.2. NMPC - based distributed motion planning with disturbance compensation

NMPC - based distributed approach of multi-agent systems has an operational mechanism based on information exchange between an agent and its neighbors to converge towards a standard solution. Figure. (5.7) illustrates the distributed motion planning in the context of this paper, if the connectivity among agents is guaranteed, each controller of the agent receives information from the others and updates its actions to reach a consensus. The disturbances acting on the agents are compensated by using the estimates obtained with the nonlinear observer.

Let us summarize in the following the necessary ingredients for formulating the motion planning with disturbance compensation algorithm:

1. to obtain more flexible behavior, the lookahead distance, Λ from (5.21) is time-varying and optimized to obtain accelerated convergence and little overshoot.

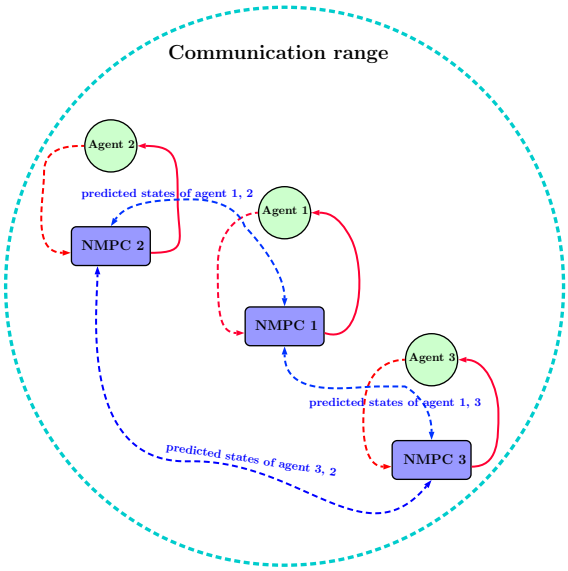


Figure 5.7.: Distributed motion planning architecture.

2. to satisfy collision avoidance with the ground and comply to rules 8 and 13 of the COLREGS with moving obstacles we add the repulsive potential constructions from (5.8), (5.13) in an NMPC optimization problem which will be activated only in the view range of the agents;

3. to comply with rules 14 and 15 of the COLREGS we add a slack variable in the cost to give priority to a negative rate of change of yaw moment, i.e., turning to starboard side as in (Mohamed, F. Martin, and Axel 2018).

4. to ensure the agents' connectivity we add in the optimization problem the connectivity function given in (5.18);

5. to reject the unknown disturbance from the current affecting the control system of the agents, the output of the NDO as in (5.27)-(5.28) is introduced as a compensation signal;

6. the implementation is done in a distributed fashion, i.e., the agents exchange information only with those inside their communication range.

The nominal system of i^{th} agent is defined from Eq. (5.1) by neglecting the disturbances, i.e., $w_i(t) \equiv 0$, and given by

$$\dot{\bar{x}}_i = f_i(\bar{x}_i(t), \bar{\mathbf{u}}_i(t), 0) \tag{5.29}$$

For each nominal system (5.29) we solve a finite horizon open-loop OCP (optimal control problem) at time t, using the measured state $x_i(t)$ over the prediction horizon T_p:

$$\min_{\bar{\mathbf{u}}_i(\cdot),\Lambda_i(\cdot)} \int_t^{t+T_p} \big[\mathcal{L}_i(\bar{x}_i(\tau), \bar{x}_j(\tau), \bar{\mathbf{u}}_i(\tau))\big] d\tau + E(\bar{x}_i(t+T_p)), \tag{5.30}$$

subject to:

$$\dot{\bar{x}}_i = f_i(\bar{x}_i, \bar{\mathbf{u}}_i, 0),\ \bar{x}_i(t) = x_i(t), \tag{5.31a}$$
$$\bar{\mathbf{u}}_i(\tau) \in \mathcal{U}_i, \tag{5.31b}$$
$$\underline{\Lambda} \leq \Lambda(\tau) \leq \bar{\Lambda}, \tag{5.31c}$$
$$\epsilon_i(\tau) > 0, \tag{5.31d}$$
$$\dot{\mathcal{T}}_{r_i}(\tau) - \epsilon_i(\tau) < 0, \tag{5.31e}$$
$$\bar{x}_i(\tau) \in \mathcal{X}_i, \tag{5.31f}$$
$$\bar{x}_j(\tau) \in \mathcal{X}_j,\ \forall \tau \in [t, t+T_p]. \tag{5.31g}$$

The stage cost $\mathcal{L}_i(.)$ has the following expression:

$$\begin{aligned}\mathcal{L}_i(.) = \|\bar{x}_i(\tau) - x_{i,ref}(\tau)\|^2_{Q_i} + \|\bar{\mathbf{u}}_i(\tau)\|^2_{R_i} + \|\dot{\bar{\mathbf{u}}}_i(\tau)\|^2_{\Delta R_i} + \Omega_{\text{cell}}(\bar{x}_i(\tau)) \\ +\Omega_{\text{mov}}(\bar{x}_i(\tau), \bar{x}_j(\tau)) + \mathcal{M}_i(\bar{x}_i(\tau), \bar{x}_j(\tau)) + \|\epsilon_i(t)\|^2_{S_i}.\end{aligned} \tag{5.32}$$

and the terminal cost is defined as:

$$E(\bar{x}_i(t+T_p)) = \|\bar{x}_i(t+T_p) - x_{i,ref}\|^2_{P_i}, \tag{5.33}$$

In (5.31a), $f_i(\cdot,\cdot)$ is presented in (5.1), $\bar{x}_i(\tau), \bar{\mathbf{u}}_i(\tau)$ are the predicted states and inputs while $\bar{\mathbf{u}}_i(.)$ represents the predicted input trajectory along the prediction horizon T_p. In the cost per stage (5.32), $\dot{\bar{\mathbf{u}}}_i(\tau)$ denotes the predicted input variations, $x_{i,ref}$ is the reference trajectory, Ω_{cell} and Ω_{mov} are the on-off barrier repulsive potential for collision avoidance given in (5.8) and (5.13) while $\mathcal{M}_i$ penalises the connectivity maintenance between agent i and its neighbors as defined in (5.18) and the last term ϵ_i is the slack variable prioritizing a certain yaw orientation. Q_i, R_i, P_i and S_i are (semi)-positive definite weighting matrices of appropriate dimensions. The constraint of lookahead distance, $\Lambda(\tau)$ is presented in Eq. 5.31c. Constraint (5.31d) is tightened by the positive slack variable, ϵ_i (5.31e) in order to prioritize the negative rate of change of the yaw moment. This forces the vessel to turn to starboard in any situation, hence complying with the rules 13, 14 and 15 of COLREGS (see Appendix). Rule 8 is also obeyed through the use of the repulsive potential in (5.13).

In (5.33), $x_{i,ref}$ is the collision-free path generated using RRT* and detailed in the forthcoming section.

At each sampling instance, each agent solves its OCP (5.30) and yields the minimizing control sequence for the nominal system (5.29) over the interval $[t, t+T_p]$. Then, only the first sample of the control input will be combined with the estimated disturbance in a closed-loop to yield a compound control as follows:

$$\mathbf{u}_i(t) = \mathbf{u}^*_i(t) - \hat{w}_i(t), \tag{5.34}$$

where $\mathbf{u}^*_i(t)$ is the output of the minimization problem (5.30), (5.31).

From (5.34), agents compensate external disturbances in real-time and obtain a predicted collision-free trajectory $\bar{x}_i(\tau)$ through the use of repulsive potential fields constructed for fixed and moving obstacles. The information (predicted trajectory) of agent i will be transmitted to its neighbor $j \in \mathcal{N}_i$ whenever the connectivity between them holds. Then the agent j with the information updated from agent i will solve its own OCP (5.30). This procedure will continue until agent $i \in \mathcal{V}$ solves its own problem (5.30) and is repeated at next sampling instance.

Note that the repulsive potentials of static obstacles (5.7) and the moving obstacle/other ships (5.12) are taken into account over the prediction horizon for collision avoidance, and is called repeatedly, for each time instant τ. However, they are disabled due to combining with on-off barrier function as in (5.9) and (5.14) (it has a "0" value if nothing is in the ship's view range). The repulsive potential will be activated via on-off barrier function (it has a "1" value if the moving obstacle/other ships or the fixed obstacles are in ship's view range). This approach has shown effectiveness when the potential

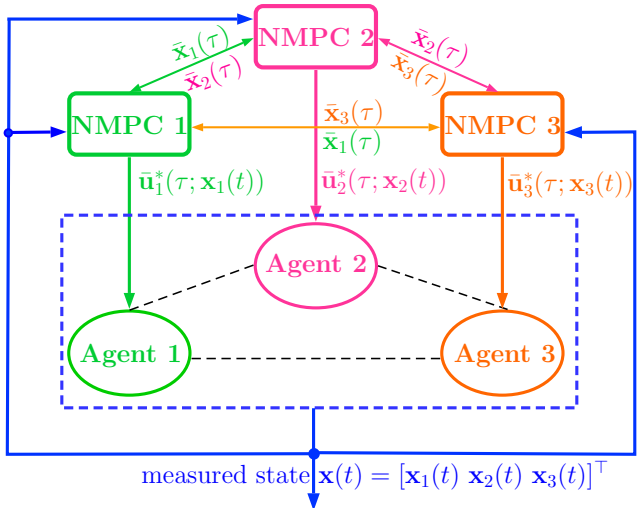


Figure 5.8.: Parallel distributed NMPC architecture.

field changes constantly via on-off barrier function thus allowing the vessel to avoid getting trapped inside a local minima **C41**.

Note that the distributed NMPC scheme employed is a parallel architecture with the aim of improving the closed-loop performance for USVs (compared with the sequential architecture) (J. Liu, X. Chen, Muñoz de la Peña, and P. D. Christofides 2010). More specifically, the separate NMPC controllers are evaluated based on a new measured state (as can be seen $x(t)$ in Fig. 5.8) and broadcast their information (i.e., predicted states, for example, $\bar{x}_1(\tau)$, $\bar{x}_2(\tau)$ and $\bar{x}_3(\tau)$ in Fig. 5.8) to their neigbors if they stay interior of the communication range in *parallel* at the same sampling time. Fig. 5.8 illustrates the communication topology of the distributed control framework for three agents. The details are presented in Algorithm 4.

Several remarks are as follows:

Remark 5.2. Only the nominal system (5.29) is utilized in OCP.

Remark 5.3. The initial state of NDO, $\hat{q}_i(\tau)$ (5.27), has to be provided.

Remark 5.4. Regarding the rate of change of yaw moment, $\dot{\mathcal{T}}_{r_i}$, there are two possible cases: i) $\dot{\mathcal{T}}_{r_i} > 0$ (i.e., ship will turn to port side), ii) $\dot{\mathcal{T}}_{r_i} < 0$, (i.e., ship will turn to starboard).

The slack variable weight should be carefully chosen so that $\dot{\mathcal{T}}_{r_i}$ can only be a very small positive number but still satisfies constraint (5.31e) while the slack variable, ϵ_i, is always a small positive number as in (5.31d). Therefore, this case ($\dot{\mathcal{T}}_{r_i} > 0$) is improbable. In other words, the case ($\dot{\mathcal{T}}_{r_i} < 0$) is strongly encouraged.

Remark 5.5. The prediction horizon should be chosen large enough to ensure cover of the agent's view range or at least assure that the agent has time enough to avoid the imminent collision. Conversely, the prediction horizon cannot be chosen arbitrarily large due to numerical and computational issues.

The procedure of information exchange among the agents in Algorithm 1 is based on (Müller, Reble, and Allgöwer 2012). Collision-free motion planning with the rejection of external disturbances through

NDO is proposed by the authors and described in the following:

Algorithm 4: Information exchange procedure and collision avoidance of the NMPC-based distributed motion planning with the rejection of external disturbances through NDO.

Require: Consider the set of agents $\mathcal{V}$ with the nominal model ((5.1) with $w_i = 0$), the set of repulsive potentials of the forbidden cells (5.7) and of the agents (5.12), the on-off barrier function as in (5.9) and (5.14), the safe distance $D_s^{i,\ell}$, the ranges of agents $i, j \in \mathcal{V}$, and set of waypoints obtained off-line through RRT*.

1: $\tau \in [t, t + T_p]$;
2: Initialize the NDO as in (5.27)-(5.28);
3: **for** $i = 1 : N$ **do**
4: Update the information of its neighbors ($\mathcal{N}_i$);
5: **if** $\|\Delta p_{i,\ell}^s\| > D_s^{i,\ell}$ and $\|\Delta p_{i,j}^d\| > \min(\rho_i, \rho_j)$ **then**
6: inactivate repulsive potential of fixed obstacles and agent j in (5.32);
7: **end if**
8: **if** $\|\Delta p_{i,\ell}^s\| > D_s^{i,\ell}$ and $\|\Delta p_{i,j}^d\| \leq \min(\rho_i, \rho_j)$ **then**
9: inactivate repulsive potential of fixed obstacles and activate repulsive potential of agent j in (5.32);
10: **end if**
11: **if** $\|\Delta p_{i,\ell}^s\| \leq D_s^{i,\ell}$ and $\|\Delta p_{i,j}^d\| > \min(\rho_i, \rho_j)$ **then**
12: activate repulsive potential of fixed obstacles and inactivate repulsive potential of agent j in (5.32);
13: **end if**
14: **if** $\|\Delta p_{i,\ell}^s\| \leq D_s^{i,\ell}$ and $\|\Delta p_{i,j}^d\| \leq \min(\rho_i, \rho_j)$ **then**
15: activate repulsive potential of fixed obstacles and agent j in (5.32);
16: **end if**
17: **end for**
18: Agent i solves OCP (5.30) and obtains the predicted collision-free trajectory $\bar{x}_i(\tau)$;
19: Transmit $\bar{x}_i(\tau)$ to its neighbors $j \in \mathcal{N}_i$;
20: Apply only the first sample of the compound control sequence as in (5.34) to both the nominal system and NDO over the interval $[t, t + T_p]$;
21: Continue to the next sampling instance;
22: RETURN step 1;

5.5. Simulation results

For the simulations we use the Cybership II model of a ship. This characterizes a real ship at a scale of 1:70 (thus, various parameters which appear later like view range, ship length, surge bound, are scaled proportionally). Since we aim to test our algorithm on a realistic benchmark (using real movement data, as given by the Automatic Identification Systems (AIS)), we scale all the information (distances, velocities) proportionally to the scale mentioned earlier (e.g., the distance between harbors used in simulation is 70 times less than the real one).

Consider a set of $N = 3$ underactuated ships with the dynamical model (5.1), where the length overall (LOA) is 1.255 [m], the matrices[9] R, M, D and B taken from (Fredriksen and Pettersen 2006):

$$R_i = \begin{bmatrix} \cos\psi_i & -\sin\psi_i & 0 \\ \sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, M_i = \begin{bmatrix} 25.8 & 0 & 0 \\ 0 & 33.8 & 1.0115 \\ 0 & 1.0115 & 2.76 \end{bmatrix}, D_i = \begin{bmatrix} 0.9257 & 0 & 0 \\ 0 & 2.8909 & -0.2601 \\ 0 & -0.2601 & 0.5 \end{bmatrix}.$$

These vessels are simulated to navigate in the Trondheim fjord (Norway). Fig. 5.1 illustrates the

[9]For simplicity, the Coriolis matrix is neglected.

operating region of the ships which need to navigate to different harbors while maintaining connectivity and avoiding the shore or small islands and other vessels. The AIS is used to provide real numerical data related to the position and velocity of ships, as well as time of navigation between harbors.

The three agents in the group need to maintain the connectivity among them while traveling from Orkanger to Trondheim harbor (for situation 1) and Stjordal harbor (for situation 2) and avoid fixed obstacles and a mobile obstacle which has the dynamical model described in (5.1).

The number of cells considered from partitioning the map as in Fig. 5.1 is $N_{\text{cell}} = 22$.

Since the view-range is used both to receive (i.e., detect) obstacles' information and transmit signals to other ships, the view-range of the ships (5.3) with $\rho_i = 70$ [m] is assumed. The scaling coefficient is $e_i \in [1.2, 1.5]$.

The steepness of the repulsive potential defined in (5.9) and (5.14) is given by $\beta_i = 1$. The steepness of connectivity maintenance function (5.18), β_{ij} and curve's maximum value, L_{ij} used as in (5.17) are $\beta_{12} = \beta_{13} = \beta_{23} = -0.002$ and $L_{12} = L_{13} = L_{23} = 1000$. Other parameters of the NMPC optimization problem in (5.30) are chosen as follows: the weighting matrices $Q_i = 0.1I_6$, $R_i = 0.1I_2$, $P_i = [0.5I_2 \quad 0_2 \quad 0_2; 0_2 \quad I_2 \quad 0_2; 0_2 \quad 0_2 \quad I_2]$, $S_i = 10^{-6}$, we consider a conservatively chosen prediction horizon $T_p = 6s$ in order to asses less than ideal conditions (decisions have to be made when the other ships/obstacles are already very close) and the sampling time is $2s$.

The design parameters of the NDO in (5.27) and (5.28) are chosen as follows $K_{1i} = diag\{0.2, \quad 0, \quad 0.2\}$, $K_{2i} = diag\{0.025, \quad 0, \quad 0.025\} = d_i$, $\lambda_i = 1$ and $\beta_i^{NDO} = 1$.

We consider the surge velocity of agent 1 to be the highest ($u_1 \in [-0.2, 0.6]$ [m/s]); the surge velocities of agent 2 and 3 equal ($u_2 = u_3 \in [-0.2, 0.4]$ [m/s]) and less than the surge velocity of agent 1. The yaw angles' constraint is $[-\pi, \pi]$. We consider constraints on the actuation force $\mathcal{T}_{u_1} = \mathcal{T}_{u_2} = \mathcal{T}_{u_3} \in [-2, 2]$ [N] and on the yaw moment $\mathcal{T}_{r_1} = \mathcal{T}_{r_2} = \mathcal{T}_{r_3} \in [-0.2, 0.2]$ [Nm]. The lookahead distance is taken in the interval, $\Lambda(\tau) \in [2\text{LOA}, 10\text{LOA}]$ where LOA is the maximum length of a ship's hull, 1.255 [m] as stated in (Fredriksen and Pettersen 2006).

The simulations are done using IPOPT solver and CASADI (Andersson, Gillis, Horn, Rawlings, and Diehl In Press, 2018) toolkit in Matlab R2016a on a computer with the following configuration: Intel Core i7-4790CPU, 3.60GHz, 8GB RAM.

In the following we present only two scenarios, more detailed simulations are given in **J14**.

5.5.1. Scenario 1 - connectivity maintenance

Scenario 1 evaluates the proposed algorithm for a group of three vessels which need to maintain connectivity while complying with the COLREGS rules. In this scenario no disturbances are consider.

Situation 1

Fig. 5.9 shows the off-line collision-free path generated using the RRT* algorithm after 1500 iterations. Fig. 5.10 depicts the actual motion of the three vessels traveling from Orkanger to Trondheim harbor.

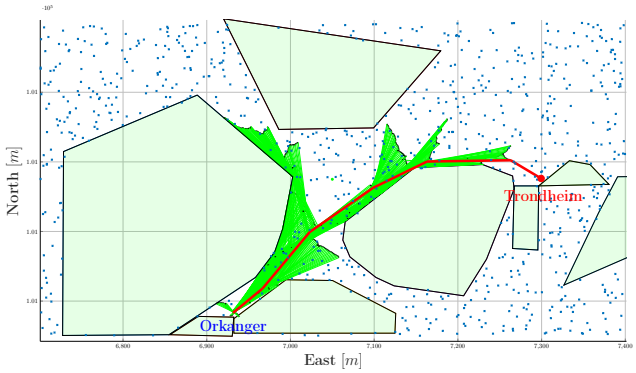


Figure 5.9.: Feasible collision-free path from Orkanger to Trondheim harbor.

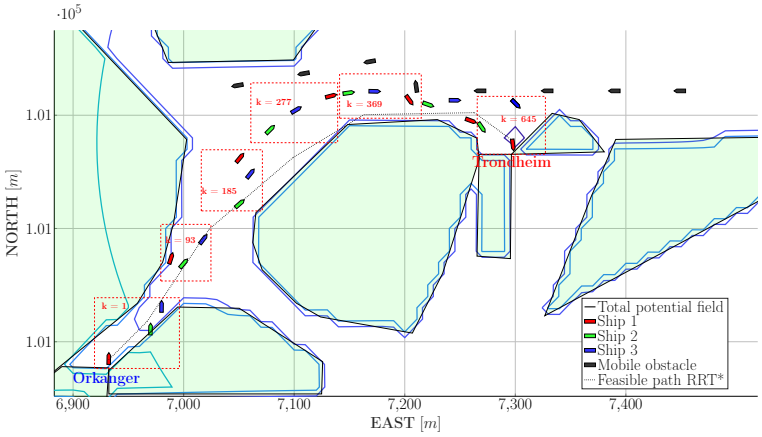


Figure 5.10.: Connectivity maintenance of the 3 USVs while traveling from Orkanger to Trondheim harbor with COLREGS compliance - rules 13 and 14.

In the same figure, we show the group of agents at 7 different time instances as they are changing their formation configuration, for instance due to the moving obstacle motion illustrated in black. Fig. 5.14 shows that the connectivity condition is maintained. At time instance $k = 93$, the ship 1 (in red)

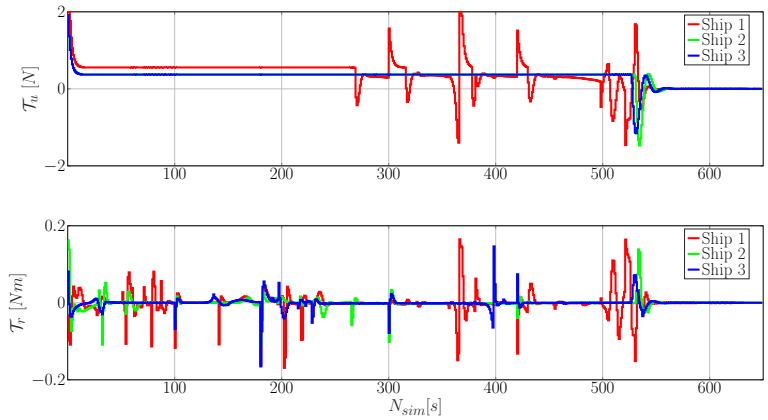


Figure 5.11.: Control inputs of the three vessels in situation 1.

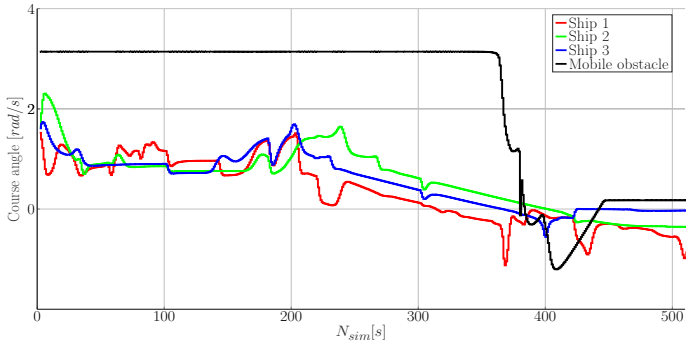


Figure 5.12.: Course angles of the three vessels and obstacle ship in situation 1.

is overtaking on the port side of the second and third ship, hence satisfying the COLREGS rule 13 - the overtaking situation. This can also be observed in Fig. 5.13 and 5.12 where ship 2 (in green) and 3 (in blue) are being overtaken, not changing their course and speed until step $k = 135$. In time interval $k \in [365, 400]$ the COLREGS rule 14 concerning head-on collisions between ships and/or

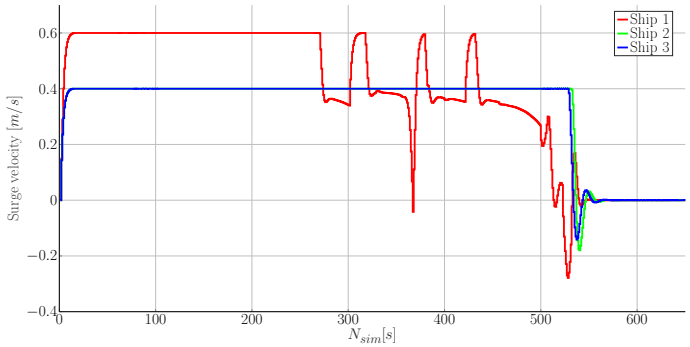


Figure 5.13.: Surge velocities of the three vessels in situation 1.

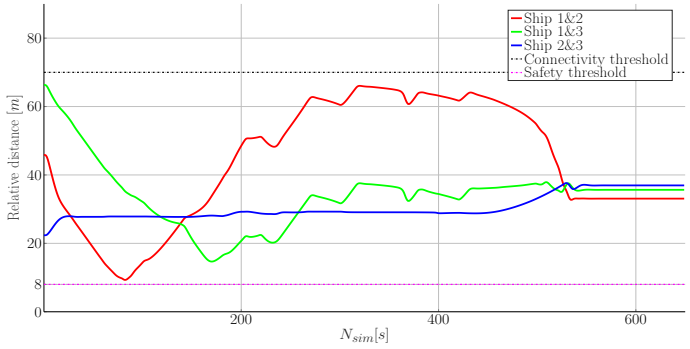


Figure 5.14.: Relative distances of the three agents in situation 1.

mobile obstacles, becomes active. The first ship in the group (in red) and the mobile obstacle start changing their course angles to turn to their starboard (right side) at time step $k = 365$ (Fig. 5.12), then the course angle of the third ship of the group (in blue) has a significant increase at time step $k = 400$ (see also Fig. 5.12). Note that if the course angle of the ship is increasing, then the vessel is steered to its starboard and vice-versa.

Fig. 5.11 depicts the control inputs corresponding to surge thrust and rudder deflection of the three vessels.

5.5.2. Scenario 2 - connectivity maintenance and COLREGS compliance

Assume that the disturbances from the ocean environment are time-varying and are described as follows:

$$\begin{cases} w_{u_i} = 0.96sin(0.02t) + 0.84sin(0.03t), \\ w_{v_i} = 0, \\ w_{r_i} = -0.16sin(0.09t + \frac{\pi}{3}) - 0.02sin(0.01t). \end{cases} \tag{5.35}$$

Then the external disturbances are bounded by $\begin{bmatrix} \|w_{u_i}\| & \|w_{r_i}\| \end{bmatrix}^\top \leq \begin{bmatrix} 1.72 & 0.18 \end{bmatrix}^\top$.

Situation 1

In this scenario, the NMPC-based optimized control is combined with NDO (eq. (5.27) and (5.28)) in the presence of the time varying disturbance as in eq. (5.35). The effectiveness of the combination between NMPC and NDO is shown in Fig. 5.15 where the trajectories of three vessels are similar with those shown in situation 1 of scenario 1. The connectivity maintenance is also guaranteed as observed in Fig. 5.16.

Although the course angle of the three ships is oscillating more than in the case of no disturbance

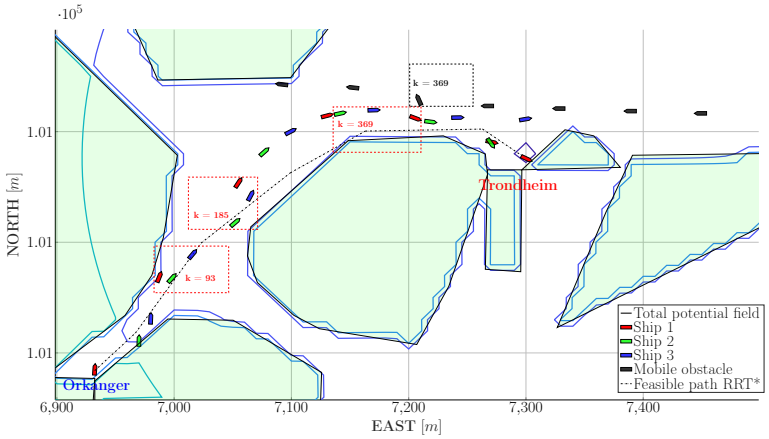


Figure 5.15.: Connectivity maintenance of the three ships group for situation 1 in the presence of disturbances and NDO.

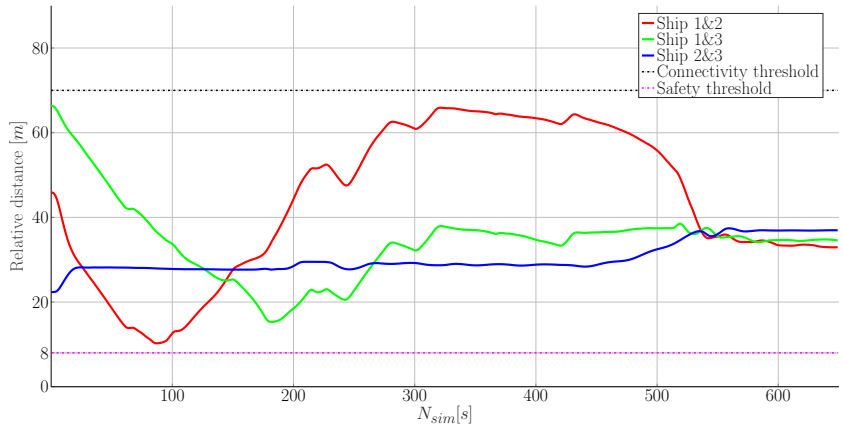


Figure 5.16.: Relative distances of the three agents in situation 1 in the presence of disturbances and NDO.

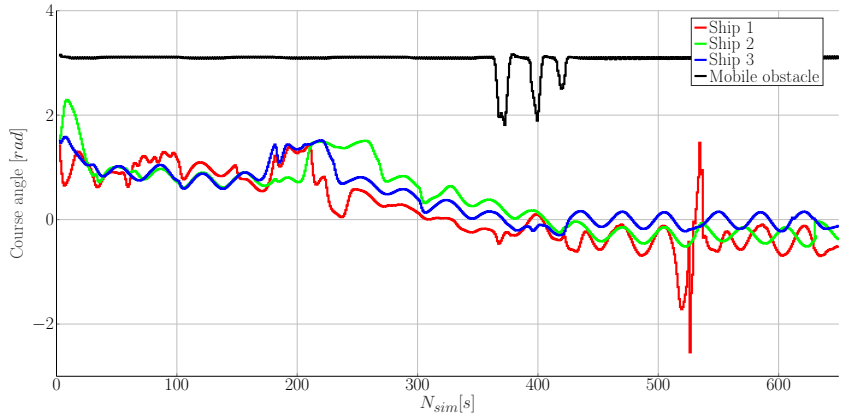


Figure 5.17.: Course angles of the three ships in situation 1 in the presence of disturbances and NDO.

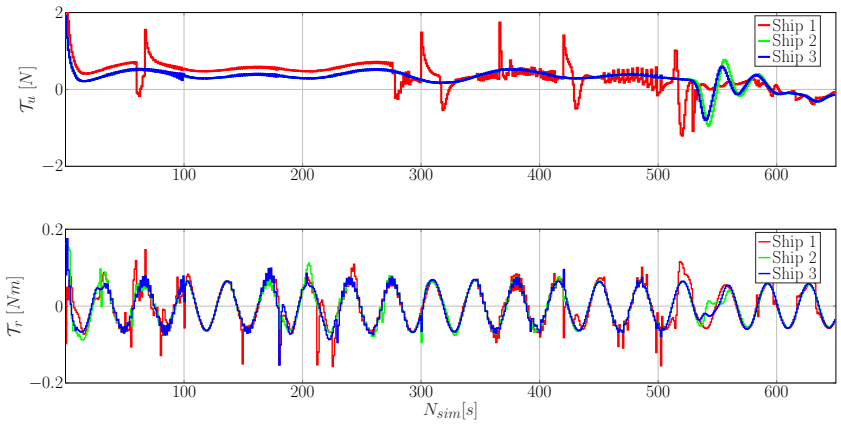


Figure 5.18.: Control inputs of the three ships in situation 1 in the presence of disturbances and NDO.

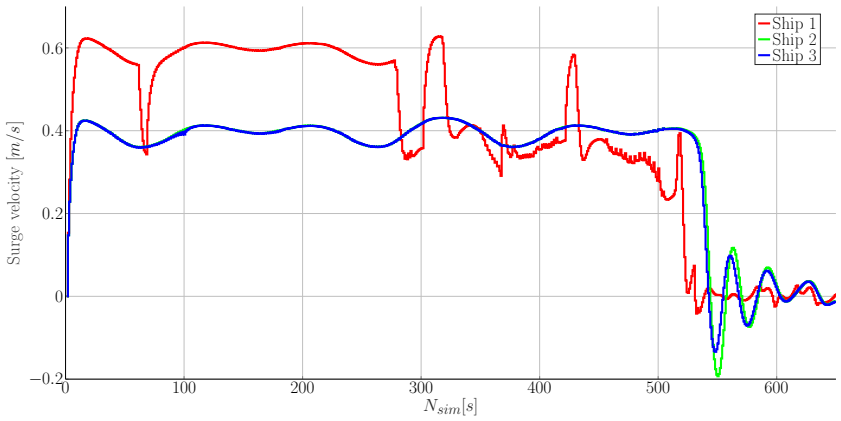


Figure 5.19.: Surge velocities of the three ships in situation 1 in the presence of external disturbances and NDO.

(in Fig. 5.12), there is also an increased trend for the ships and the mobile obstacle to steer to their starboard side. For example, at the time instance $k = 368$, the course angle of ship 1 (red line) and mobile obstacle (black line) increase, which also can be observed in Fig. 5.17. As a consequence, the COLREG rules 13 and 14 are maintained as in situation 1 of Scenario 1.

Fig. 5.18 and 5.19 illustrate the control inputs and surge velocities of the three vessels.

Situation 2

In this situation, the connectivity maintenance and collision avoidance among agents in the group are also maintained in the simulation as illustrated in Fig. 5.22. Rule 13 of COLREG - overtaking is obeyed as illustrated in Fig. 5.21 (at time instance $k = 101$, ship 1 in red, overtakes on the left side of the two ships). Moreover, rule 15 - crossing, is also complied when the obstacle ship keeps its direction and ship 1 turns starboard for collision avoidance as can be seen ($k = 486$) in Fig. 5.21. This can also be seen in Fig. 5.23, the course angle of ship 1 (red) which increases at $k = 486$ while course angles of ship 1 (green) and 3 (blue) are not changing since the mobile obstacle has passed by the time they enter this area. Similar to situation 1 in this scenario, good results are obtained since NDO provides an estimate that rapidly converges towards the real disturbances.

Note that in Fig. 5.10, 5.15 and 5.21, we also illustrate the repulsive and attractive potentials projected in 2D (the repulsive potentials appear around the forbidden region preventing the ship's collison with the fixed and moving obstacles and the attractive potential is represented by circles

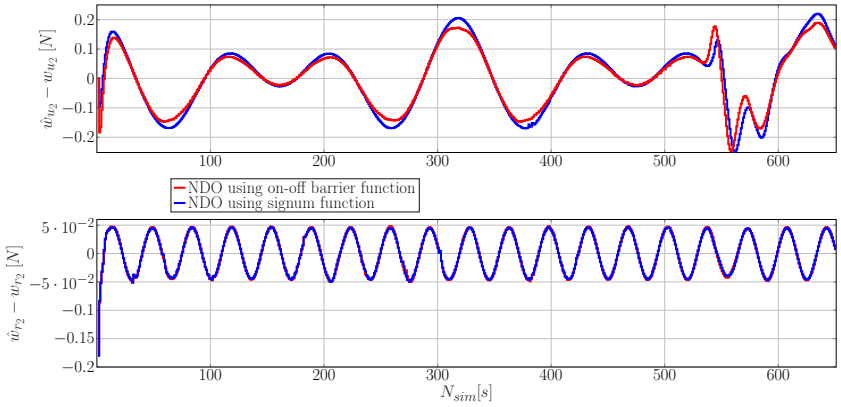


Figure 5.20.: Comparison of NDO using signum function and on-off barrier function.

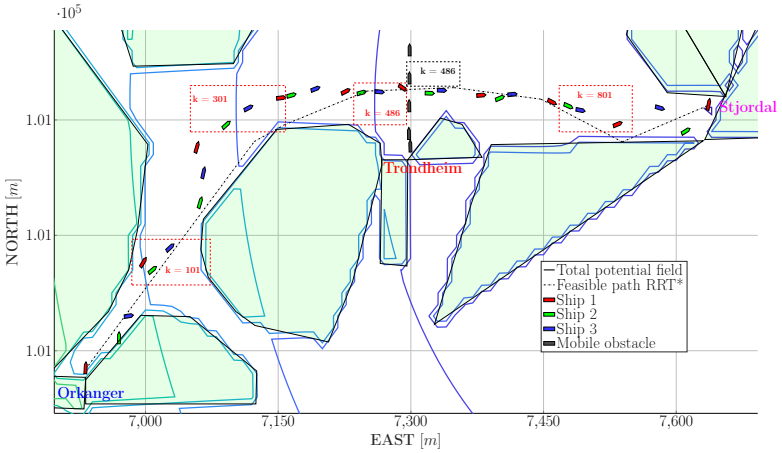


Figure 5.21.: Connectivity maintenance of the three ships group for situation 2 in the presence of external disturbances and NDO.

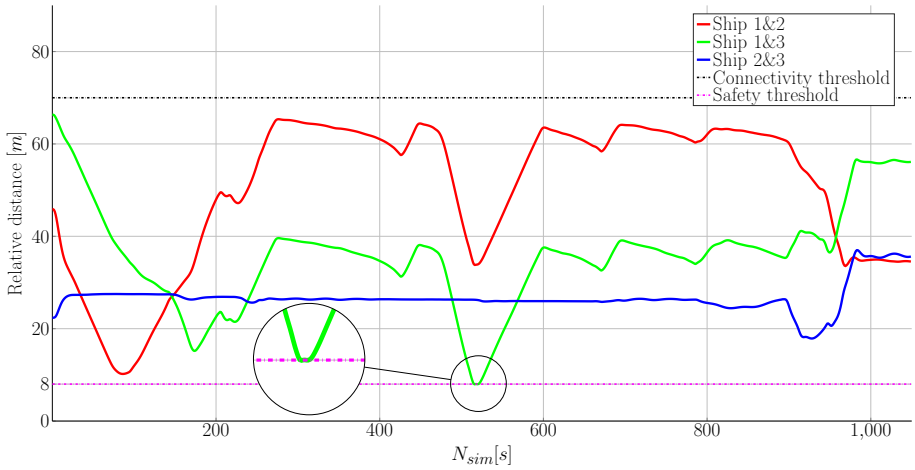


Figure 5.22.: Relative distances of the three agents in situation 2 in the presence of disturbances and NDO.

around the destination harbor).

Computation time of the optimization problem, trajectory length and number of simulations to

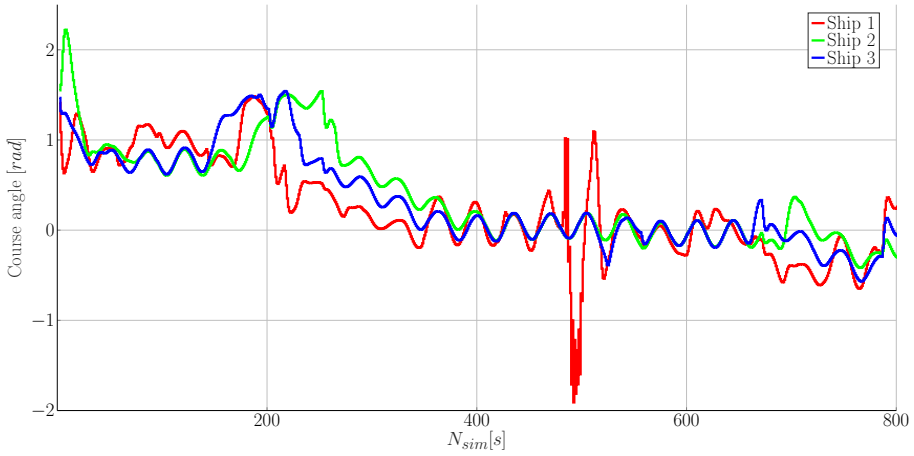


Figure 5.23.: Course angles of the three ships in situation 2 in the presence of disturbances and NDO.

arrive at the destination are delineated in Table 5.1 and 5.2.

Table 5.1.: Performance criteria for the motion planning algorithm of situation 1 for the two scenarios.

Situation 1	Scenario 1	Scenario 2
Prediction horizon [s]	6	6
Number of simulations	650	650
CPU time [s]/step	0.0039	0.0039
Length of trajectory [m]	505.2	503.4

Table 5.2.: Performance criteria for the motion planning algorithm of situation 2 for the two scenarios.

Situation 2	Scenario 1	Scenario 2
Prediction horizon [s]	6	6
Number of simulations	1050	1050
CPU time [s]/step	0.0103	0.0103
Length of trajectory [m]	847.6	858.5

5.6. Conclusions and discussions

An NDO-based distributed NMPC scheme has been developed for motion planning problem with connectivity maintenance and collision avoidance for multi-surface vehicles. The classical non-convex constraints from this problem were penalized in the cost function through appropriate potential field constructions. NDO can estimate the external disturbance and compensate for their impact via feedback control.

Although simulation results prove excellent performance of the algorithm as well as give insights for real implementation, there are still some limitations that will be considered in the future work.

On one hand, instead of assuming a fixed slope parameter β, we may obtain it through an estimation/fitting procedure which maximizes the appearance of the obstacles in the agent's view range, i.e., a collision risk may be alerted earlier. The *Newton-Raphson* (Al-Daffaie and Khan 2017; Bakari,

Adegoke, and Yahya 2016) and *Gradient Descent* (Kim, Song, S. Wang, Xia, and Jiang 2018; Manogaran and Lopez 2018) method as well as its variants such as *Batch Gradient Descent* and *Stochastic Gradient Descent* can be employed to estimate this parameter.

On the other hand, one of the most challenging perspectives is to guarantee asymptotic stability of the potential field-based MPC scheme by using the predicted input trajectory along the prediction horizon. Without doubt, this is not a trivial problem to deal with since MAS have to leave their predicted paths due to repulsive forces coming from the forbidden areas. Some well-known approaches from the literature are promising. Stability can be ensured for finite-horizon problems by suitably choosing a weighting matrix for terminal cost and an attractive terminal region (H. Chen and Allgöwer 1998) **J13**. Alternatively, closed-loop stability can also be achieved for relatively long horizons without the need to use a terminal cost or a terminal constraint (Grüne and Pannek 2017; Grüne 2012). Stability is guaranteed by tuning the weighting matrices of the cost function.

The control of surface vehicles is challenging due to the factors involved (model uncertainties and disturbances) and classical methods often fail to provide acceptable solutions. Hence, recent approaches use heuristic methods for trajectory generation and subsequent tracking. The advantages of these methods lie in their capacity to handle complex and changing environments, and even to fulfill risk criteria, (Blindheim, Gros, and Johansen 2020).

6. From feedback linearization to NMPC design with stability guarantees

This chapter addresses first the benefits of a computed-torque control law within an NMPC framework for a particular class of systems, the "computed-torque like" systems. We propose the necessary ingredients to guarantee the stability of an NMPC design using an admissible invariant set as a terminal constraint set. We apply the method to design an NMPC attitude controller for stabilizing the rotation dynamics of a multicopter system.
We further concentrate in this chapter on presenting two different NMPC (Nonlinear Model Predictive Control) schemes (with and without terminal cost and constraints) for stabilizing the translational dynamics of a multicopter system. Both approaches make use of an elaborated nonlinear feedback linearization controller and its associated ellipsoidal invariant set under restrictive input constraints, hence, guaranteeing the closed-loop asymptotic stability. The terminal constraint set of the corresponding NMPC design is easy to tune due to its clear formulation expressed directly in terms of the tuning variables while for the NMPC scheme without terminal constraint, the design allows to stabilize the system with a significantly shorter prediction horizon in comparison with existing method in the literature. Simulation and experimental tests over a nano-drone platform validate the proposed approaches.

The chapter is based on the following papers:

J13 Nguyen, N. T., **I. Prodan**, and L. Lefèvre. "Stability guarantees for translational thrust-propelled vehicles dynamics through NMPC designs". In: *IEEE Transactions on Control Systems Technology*, pp. 1–13, 2020. IEEE.
DOI: 10.1109/TCST.2020.2974146.

C46 Nguyen, N., **I. Prodan**, and L. Lefevre. "A stablizing NMPC design for thrust-propelled vehicles dynamics via feedback linearization", pp. 735–742. In: *Proceedings of the IEEE American Control Conference (ACC'19)*. 2017. Philadelphia, USA.

C45 Nguyen, N., **I. Prodan**, and L. Lefevre. "On the use of a computed-torque control law for the terminal region of an NMPC scheme", pp. 432–439. In: *Proceedings of the IEEE American Control Conference (ACC'19)*. 2017. Philadelphia, USA.

C53 Nguyen, N. T., **I. Prodan**, and L. Lefevre. "Multicopter attitude control through NMPC design with guaranteed stability", pp. 1–6. In: *Proceedings of the 21th IFAC World Congress*. 2020. Berlin, Germany.

6.1. Brief literature review

In the control research community a popular method for easily handling the constraints is MPC (Model Predictive Control) (Badgwell and Qin 2015; Alamir 2006; Mayne, Rawlings, Rao, and Scokaert 2000) which is increasingly employed by many robotics applications (Badgwell and Qin 2015; Ferrara, Incremona, and Magni 2013; Poignet and Gautier 2000), e.g., automotive manufacturing, inspection and even robotic surgery (J. Craig 2018; Lewis, Dawson, and Abdallah 2003). The well-known MPC design employing: i) a terminal cost, ii) a terminal constraint set, also referred as terminal region and iii) a stage cost, achieves superior performance when compared to other MPC-based design methods, e.g., *zero state* and *terminal constraint set* MPC (Mayne, Rawlings, Rao, and Scokaert 2000; H. Chen and Allgöwer 1998). Moreover, it possesses the designing rules which, if satisfied, guarantee the asymptotic (exponential) stability of the closed-loop system Mayne, Rawlings, Rao, and Scokaert 2000. These conditions revolve around an ingredient hidden from the MPC scheme, i.e., a local controller under which the constraint set is positively invariant. In the literature, almost all related NMPC applications employ a linear local controller (Mayne, Rawlings, Rao, and Scokaert 2000; H. Chen and Allgöwer 1998), due to the complexity of the nonlinear controlled systems. However, employing a linear controller for a general nonlinear system obviously restricts the corresponding invariant set (also serving as the terminal region), reducing the efficiency of the associated MPC controller.

In view of these shortcomings and also motivated by seeking a more appropriate local controller for an NMPC design of robotics systems, the CTC (Computed-Torque Control) law appears as a promising candidate which hopefully provides a larger invariant set and a better insight into the system's behavior (w.r.t. the set under the linear controller). Indeed, one way to classify robotic control schemes is to divide them into "computed-torque-like" or "non-computed-torque-like" (Lewis, Dawson, and Abdallah 2003). There is a broad range of systems employing a "computed-torque-like" controllers, such as, aerospace crafts, industrial robot arms and mobile robots (Ferrara, Incremona, and Magni 2013; Poignet and Gautier 2000; Uebel, Minis, and Cleary 1992) or the attitude controller of a multicopter system **C53**.

Usually, the drone platforms already have their built-in controllers which control the rotors to track the four inputs which in turn govern the thrust level and the three Euler angles (Budaciu, Botezatu, Kloetzer, and Burlacu 2019). Therefore, more recently, controlling the thrust propelled vehicles requires only the maneuvering of their translation dynamics **J9**, **C39**. This is already a challenging task as the dynamics are not only strongly nonlinear but also subject to many operating constraints (Zanelli, Horn, Frison, and Diehl 2018), **J9**.

It is well known that the computational burden of the MPC-based optimization problem may pose serious issues in many time-critical applications. However, with recent technological advances, it becomes possible to employ MPC for many applications requiring fast response (Gros, Zanon, Quirynen, Bemporad, and Diehl 2020) as is the case in the control and coordination of aerial vehicles (Zanelli, Horn, Frison, and Diehl 2018; M. W. Mueller and D'Andrea 2013). E.g., (Zanelli, Horn, Frison, and Diehl 2018) approximates the NMPC (Nonliner MPC) controller for stabilizing a human-sized quadcopter vehicle with a quadratic problem and succeeds in solving it by using a Cortex A9 800-MHz microcontroller. Also, it has been proved that a linear MPC-based problem can be even executed over a 8-bit microcontroller by using a convex lifting approach (Gulan et al. 2017).

Another important issue of MPC is that it does not provide an explicit controller equation (Mayne, Rawlings, Rao, and Scokaert 2000), thus, causing difficulty in analyzing the stability and the feasibility of the closed-loop scheme (except explicit MPC whose numerical limitations make it difficult for many real-time implementations (Munir, Hovd, and Olaru 2018; Bemporad, Morari, Dua, and Pistikopoulos 2002)). The use of MPC laws without stability guarantees may prove shortsighted in certain applications. In particular, the authority loss for attitude and position control in UAVs will not only risk the mission objectives but lead to critical behavior.

In the literature, the stability of an MPC controller can be achieved either by adding a terminal constraint (Munir, Hovd, and Olaru 2018; Cannon, Deshmukh, and Kouvaritakis 2003; H. Chen and Allgöwer 1998) or by enlarging the prediction horizon (Grüne 2012). In this chapter, we employ

both approaches to design two different NMPC schemes for stabilizing the translation dynamics of multicopter systems. These designs make use of an elaborated feedback linearization controller which was introduced in the previous work of the authors **J9** and was successfully applied to a nano-drone.

6.2. Contributions

First, by exploiting the existing CTC law, we propose several contributions related to improvements for the NMPC design, which are, to the best of our knowledge, new to the state of the art:

1. an application of the CTC law to the existing NMPC design principles with guaranteed stability. The CTC law allows to construct an ellipsoidal invariant set (serving as the terminal region) associated with the closed-loop linear dynamics. It also guarantees the input constraint validation within this set;

2. an upper bound is provided for the weighted norm of the CTC controller in terms of the corresponding state within the terminal region. This is obtained using the Taylor's approximation for the nonlinear local controller;

3. an explicit formulation of the terminal region is provided in terms of the design parameters, thus, the NMPC design is capable of easily modifying (e.g., re-orientating or enlarging) the proposed terminal region.

Furthermore, we employ an elaborated feedback linearization controller as the local controller (i.e., locally constraint admissible) to design the two NMPC schemes with guaranteed stability. Note that the role of a local controller is to ensure that once inside the terminal region, the system's trajectories remain inside. Thus, the local controller only acts as a guarantee of feasibility and asymptotic stability and is not actually used throughout the simulation. Next we enumerate this paper's contributions which, to the best of our knowledge, are new to the state of the art. We:

i) propose an NMPC design with terminal ellipsoidal region which is invariant and constraint admissible under a nonlinear feedback linearization controller.

ii) propose an NMPC design without terminal constraint and the corresponding tuning procedure for choosing the optimal prediction horizon which guarantees the stability.

iii) validate the NMPC controllers through simulation and experimental testing over the Crazyflie 2.0 nano-quadcopter platform (Budaciu, Botezatu, Kloetzer, and Burlacu 2019).

Notations: Denote by $\boldsymbol{I}_n$ the identity matrix of size n and by $0_{m\times n}$ the zero matrix of size $m \times n$. $\mathbf{0}$ represents a zero matrix of appropriate dimension inferred from the context. Let $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the minimum and maximum eigenvalues of the square matrix A having only real eigenvalues. For a vector $\boldsymbol{x} \in \mathbb{R}^n$ and a positive semi-definite matrix $P \in \mathbb{R}^{n\times n}$, $\|\boldsymbol{x}\|_P^2$ denotes the weighted norm $\boldsymbol{x}^\top P\boldsymbol{x}$ while $\|\boldsymbol{x}\|$ represents the Euclidean norm of the vector $\boldsymbol{x}$. Furthermore, denote by $\boldsymbol{x}(i|k)$ and $\boldsymbol{u}(i|k)$ the values of the state $\boldsymbol{x}$ and the input $\boldsymbol{u}$ at time instant i predicted upon the information available at time $k \in \mathbb{N}$.

6.3. On the use of a computed-torque control law in an NMPC scheme

This section introduces briefly the result presented in **C45** where we start by presenting the classical dynamics of a system admitting a CTC (Computed-Torque Control) law J. J. Craig 2005, the associated invariant set that we design and pass directly to some simulations results and comparisons validated on the classical inverted pendulum robot on a cart. All the details can be found in **C45**.

The dynamics of a system admitting a CTC (Computed-Torque Control) law is expressed in the following:

$$\boldsymbol{M}(q)\ddot{q} + \boldsymbol{N}(\dot{q}, q) = \boldsymbol{u}, \tag{6.1}$$

with the state $q \in \mathbb{R}^n$ and the actuator input $\boldsymbol{u} \in \mathbb{R}^n$. $\boldsymbol{M}(q) \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite inertia matrix, $\boldsymbol{N}(\dot{q}, q) \in \mathbb{R}^n$ is the vector gathering the nonlinear terms (e.g., Coriolios forces, centrifugal forces J. J. Craig 2005; Lewis, Dawson, and Abdallah 2003). By defining the new state vector $\boldsymbol{x} = [q^\top \ \dot{q}^\top]^\top$, the system (6.1) is transformed into state-space formulation as follows:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}), \tag{6.2}$$

with

$$f(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0}_n \\ \boldsymbol{0}_n & \boldsymbol{M}(q)^{-1} \end{bmatrix} \begin{bmatrix} \dot{q} \\ -\boldsymbol{N}(\dot{q}, q) + \boldsymbol{u} \end{bmatrix}. \tag{6.3}$$

The equilibrium point of the system (6.1)-(6.19) is fixed at:

$$\boldsymbol{x}_e = \boldsymbol{0} \text{ and } \boldsymbol{u}_e = \boldsymbol{0}. \tag{6.4}$$

The state $\boldsymbol{x}$ and the input $\boldsymbol{u}$ are constrained as follows:

$$\boldsymbol{x} \in \mathcal{X}, \ \boldsymbol{u} \in \mathcal{U} = \{\boldsymbol{u} \in \mathbb{R}^n | -\boldsymbol{u}_{\max} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max}\}, \tag{6.5}$$

where $\mathcal{X}$ is a convex set in $\mathbb{R}^{2n}$ containing $\boldsymbol{x}_e$ in its relative interior and $\boldsymbol{u}_{\max}$ defines all the maximal values of the actuator inputs. For the system (6.1)-(6.19), there exists a special feedback linearization law called CTC law J. J. Craig 2005; Lewis, Dawson, and Abdallah 2003 given as:

$$\boldsymbol{u}_b(\boldsymbol{x}, v) = \boldsymbol{M}(q)v + \boldsymbol{N}(\dot{q}, q), \tag{6.6}$$

where $v \in \mathbb{R}^n$ gathers the so-called virtual control inputs. The CTC law also has to satisfy $\boldsymbol{u}_b(\boldsymbol{x}_e, v_e) = \boldsymbol{u}_e = \boldsymbol{0}$ with[1] $v_e = \boldsymbol{0}$. Furthermore, if $\boldsymbol{u}_b(\boldsymbol{x}, v)$ satisfies the input constraint (6.22), it transforms the system (6.1)-(6.19) into the linear system:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + Bv, \tag{6.7}$$

with $A \in \mathbb{R}^{2n\times 2n}$ and $B \in \mathbb{R}^{2n\times n}$ given by:

$$A = \begin{bmatrix} \boldsymbol{0}_n & \boldsymbol{I}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix}, \ B = \begin{bmatrix} \boldsymbol{0}_n \\ \boldsymbol{I}_n \end{bmatrix}. \tag{6.8}$$

According to the MPC design classification presented in (Mayne, Rawlings, Rao, and Scokaert 2000), the NMPC problem ae deal with belongs to the category in which both terminal cost and constraint set are employed. For this category, (Mayne, Rawlings, Rao, and Scokaert 2000; H. Chen and Allgöwer 1998) also provide four conditions, that, if satisfied, ensure the recursive feasibility[2] and the closed-loop asymptotic (exponential) stability. The conditions are re-formulated for our scheme as follows:

C1: States constraints satisfaction in $\mathcal{X}_f$, i.e.:

$$\mathcal{X}_f \subseteq \mathcal{X}, \ \boldsymbol{x}_e \in \mathcal{X}_f. \tag{6.9}$$

C2: There exists a local controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x})$ such that:

$$\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}) \in \mathcal{U}, \ \forall \boldsymbol{x} \in \mathcal{X}_f. \tag{6.10}$$

C3: $\mathcal{X}_f$ is positively invariant for the system (6.1)-(6.19) under $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x})$.

[1] Due to the considered equilibrium point $(\boldsymbol{x}_e, \boldsymbol{u}_e)$ in (6.4), $f(\boldsymbol{x}_e, \boldsymbol{u}_e) = \boldsymbol{0}$ with f in (6.3). This leads to $\boldsymbol{M}(\boldsymbol{0}) = \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{0}) = \boldsymbol{0}$.

[2] The initial iteration successfully executed implies the feasibility of all the further steps (H. Chen and Allgöwer 1998).

C4: $\forall \boldsymbol{x} \in \mathcal{X}_f$, the trajectory of the system (6.1)-(6.19) under $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x})$ satisfies:

$$\frac{d}{dt}\left(\|\boldsymbol{x}\|_P^2\right) + \|\boldsymbol{x}\|_Q^2 + \|\boldsymbol{u}_{\text{loc}}^2(\boldsymbol{x})\|_R^2 \leq 0. \tag{6.11}$$

Hereinafter, a positive invariant set will be constructed inside the state constraint set $\mathcal{X}$ (6.22) while validating input constraints. This will allow in fact the construction of a constraint admissible positive invariant set.

Lemma 6.1. *Let us construct the matrix K as:*

$$K = \left[\text{diag}(K_{p_1}, \ldots, K_{p_n}) \;\; \text{diag}(K_{d_1}, \ldots, K_{d_n})\right], \tag{6.12}$$

where the $2n$ control gains $K_{p_1}, \ldots, K_{p_n}$ and $K_{d_1}, \ldots, K_{d_n}$ are chosen such that:

$$\begin{cases} K_{p_i} < 0, \;\; K_{d_i} < 0, \\ 4K_{d_i}^2 > -K_{p_i}(K_{p_i}+1)^2 - K_{p_i} - \frac{(K_{p_i}+1)^2}{K_{p_i}}, \end{cases} \tag{6.13}$$

with $i \in \{1, \ldots, n\}$ and define the set $\mathcal{X}_f$ as:

$$\mathcal{X}_f = \{x \in \mathbb{R}^n |\; \boldsymbol{x}^\top (\boldsymbol{I}_{2n} + K^\top K)\boldsymbol{x} \leq \boldsymbol{\varepsilon}^2\}, \tag{6.14}$$

where $\boldsymbol{\varepsilon}$ is chosen such that:

$$\begin{cases} \boldsymbol{\varepsilon} \leq \varepsilon_{max}, \\ \mathcal{X}_f \subseteq \mathcal{X}, \end{cases} \tag{6.15}$$

with ε_{max} satisfying

$$C\varepsilon_{max} + \frac{M_{\boldsymbol{\varepsilon}_{max}}}{2}\varepsilon_{max}^2 \leq \boldsymbol{u}_{max}, \tag{6.16}$$

with $C \triangleq [C_1, \; \ldots, C_n]^\top$ and $\boldsymbol{u}_{max}$ in (6.22) and

$$|\boldsymbol{u}_b(\boldsymbol{x}, v)| \leq \boldsymbol{u}_{max}, \;\; \forall (\boldsymbol{x}, v) \in \mathcal{C}(\varepsilon_{max}), \tag{6.17}$$

Then, we have that:

(i) $\boldsymbol{x} \in \mathcal{X}_f$ imposes that the control action $\boldsymbol{u}_b(\boldsymbol{x}, K\boldsymbol{x}) \in \mathcal{U}$ (i.e., the CTC law $\boldsymbol{u}_b(\boldsymbol{x}, v)$ (6.40) employing the virtual input design $v = K\boldsymbol{x}$ satisfies the input constraints);

(ii) the set $\mathcal{X}_f$ is positively invariant for the controlled system $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}_b(\boldsymbol{x}, K\boldsymbol{x}))$ (6.19). □

Proof. The proof is provide in **C45**. □

For validating the benefits of a CTC law in an NMPC setup we consider the problem of stabilizing a well-known mechatronics system, an inverted pendulum robot on a cart (Srinivasan, Huguenin, and Bonvin 2009). The angular dynamics of the system (6.1) with $q \in \mathbb{R}$, the angle between the vertical line and the pendulum. $\boldsymbol{M}(q)$ and $\boldsymbol{N}(q, \dot{q})$ as in (6.1) are given by:

$$\boldsymbol{M}(q) = \mu \cos q - \frac{mJ}{\mu \cos q}, \;\; \boldsymbol{N}(q, \dot{q}) = mg - \mu \dot{q}^2 \sin q, \tag{6.18}$$

with $m = 0.3235$, $\mu = 1.3625 \times 10^{-3}$ and $J = 1.5265 \times 10^{-4}$ the physical parameters of the system, $g = 9.81$ is the gravity and the force applied to the cart. The input $\boldsymbol{u} \in \mathbb{R}$ is subject to the constraint $|\boldsymbol{u}| \leq \boldsymbol{u}_{max}$ with $\boldsymbol{u}_{max} = 0.6$ (corresponding to the maximum voltage of $2V$ given in (Srinivasan, Huguenin, and Bonvin 2009)). The equilibrium point of the system (6.18) is $(\boldsymbol{0}, 0)$ and the state is constrained by $|q| \leq 0.16$, $|\dot{q}| \leq 0.3$. The initial state is fixed at $\boldsymbol{x}_0 = [0.15 \; 0]^\top$.

For simulation we consider three scenarios given as follows:

Scenario 1: Stabilizing the inverted pendulum (6.18) using the CTC law $\boldsymbol{u}_b(\boldsymbol{x}, K_1\boldsymbol{x})$ (6.40) with the matrix K_1 chosen as in (6.13) such that the set $\mathcal{X}_{f_1}$ (6.14) contains $\boldsymbol{x}_0$, thus, being positively invariant.

Scenario 2: Stabilizing the dynamics (6.18) using $\boldsymbol{u}_b(\boldsymbol{x}, K_2\boldsymbol{x})$ (6.40) with the matrix K_2 chosen as in (6.13) such that the terminal region $\mathcal{X}_{f_2}$ (6.14) does not contain $\boldsymbol{x}_0$.

Scenario 3: Stabilizing the dynamics (6.19) using the qMPC controller recalled in **C45** from (H. Chen and Allgöwer 1998), with the terminal region denoted by Ω_α.

Figure 6.1 illustrates the three terminal regions, $\mathcal{X}_{f_1}$ (blue ellipsoid), $\mathcal{X}_{f_2}$ (red ellipsoid) and Ω_α (green ellipsoid). Notice that the proposed terminal regions, both $\mathcal{X}_{f_1}$ and $\mathcal{X}_{f_2}$, are significantly larger than the terminal region Ω_α. Also, our proposed NMPC controller can be executed with any prediction horizon larger than 0.2 seconds (we employ $T_2 = 0.4$ seconds for better performances), while the qMPC in Scenario 3 does not accept any prediction horizon smaller than $T_3 = 0.6$ seconds.
The two sets $\mathcal{X}_{f_1}$, $\mathcal{X}_{f_2}$ corresponding to the first two scenarios and the state trajectory under Scenario 2 (red line) all stay within the state constraint set $\mathcal{X}$ (black rectangle) which illustrates the ease of modifying the terminal region $\mathcal{X}_f$ (6.14) and the state constraint satisfaction of the proposed NMPC scheme.

Furthermore, under Scenario 1, the initial state $\boldsymbol{x}_0$, illustrated in Figure 6.1 by the black circle, is contained in the set $\mathcal{X}_{f_1}$ and the trajectory resulted from the CTC law $\boldsymbol{u}_b(\boldsymbol{x}, K_1\boldsymbol{x})$ (6.40) (plotted in dash-dotted blue line) always stays inside $\mathcal{X}_{f_1}$ which illustrates the positive invariance property of the set $\mathcal{X}_{f_1}$ proved in Lemma 6.1. As observed from Figure 6.2, both our proposed NMPC scheme under Scenario 2 (red lines) and the qMPC controller under Scenario 3 (green lines) obtain the similar convergence times of 2 seconds. This is mostly due to the fact that both qMPC and NMPC designs employ the same matrices Q, R of the cost function. The performance of the proposed NMPC controller can be enhanced by choosing more appropriate matrices Q and R. Note that, the results under Scenario 1 are not plotted in Figure 6.2 since the performance of the CTC law is not within the scope of this paper. Also, all the inputs of the corresponding controllers under the three Scenarios given in Figure 6.3 satisfy the input constraint $|\boldsymbol{u}| \leq \boldsymbol{u}_{\max} = 0.6$. Lastly, the execution time is also an essential aspect when considering any MPC control scheme. Thus, Figure 6.4 presents the computation times per step for the two Scenarios 2 and 3. It can be observed that for the proposed NMPC controllers (red line with circle marks), after several first iterations, the computation times are all around 0.15 seconds, while the computation times of the qMPC under Scenario 3 (green line with triangle marks) are significantly larger. These differences are due to the longer prediction horizon employed in the qMPC controller under Scenario 3 (i.e., $T_3 = 0.6$ seconds over $T_2 = 0.4$ seconds of our NMPC controller). Moreover, it shows the advantage of the larger terminal region $\mathcal{X}_{f_2}$ (red ellipsoid in Figure 6.1) of the proposed NMPC design when compared to the qMPC.

Discussions

Recall that this work employs the design principles of an NMPC scheme from (Mayne, Rawlings, Rao, and Scokaert 2000; H. Chen and Allgöwer 1998) (namely, conditions **C1**-**C4** as in (6.9)-(6.11)) and presents the advantages of validating them using a nonlinear local controller, the CTC law in (6.40). In our opinion the present results are evolutionary and differ from the qMPC approach (H. Chen and Allgöwer 1998) as in the following:

1. we use the CTC law (6.40) as the local controller while in (H. Chen and Allgöwer 1998), a linear controller is employed. Thus, in our case, the considered system (6.1)-(6.19) is exactly linearized into a simple linear system while (H. Chen and Allgöwer 1998) has to bound the behavior of the nonlinear system under the linear controller.

2. the CTC law (6.40) is bounded by using the Taylor's approximation which firstly provides the condition for respecting the input constraints and then paves the way to the satisfaction of condition **C4** (6.11).

3. our NMPC design using the CTC law provides flexibility in modifying the terminal region

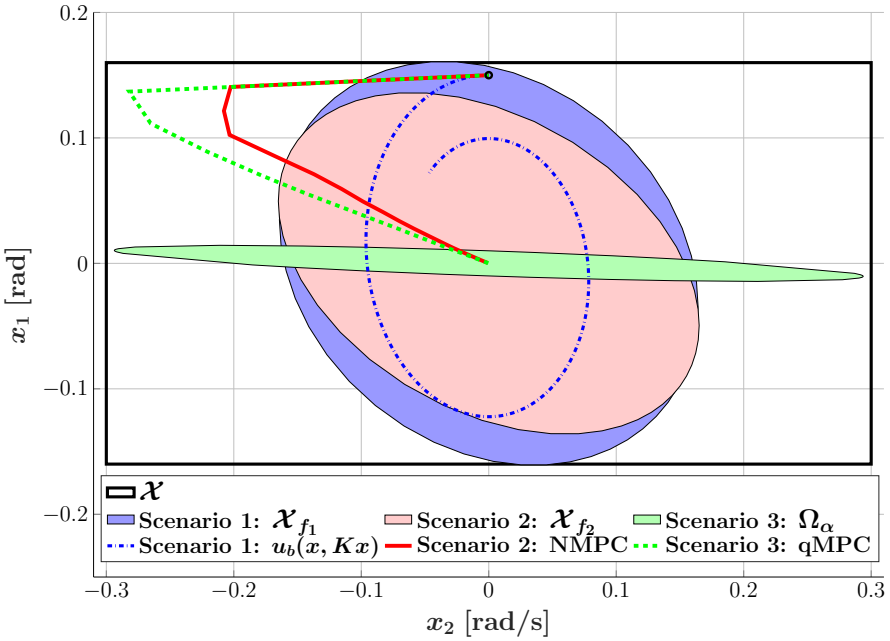


Figure 6.1.: Terminal regions $\mathcal{X}_{f_1}, \mathcal{X}_{f_2}$ and Ω_α and state trajectories under different scenarios.

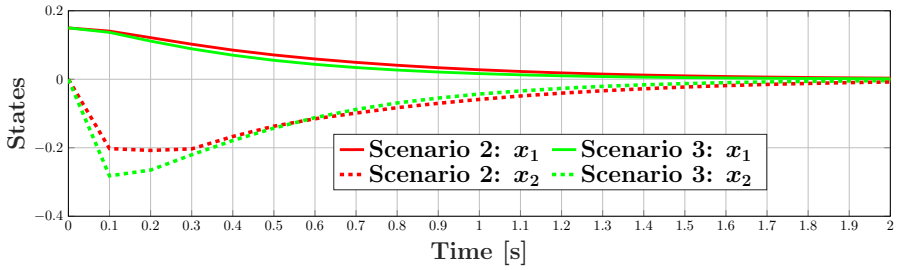


Figure 6.2.: Convergence of states under different scenarios.

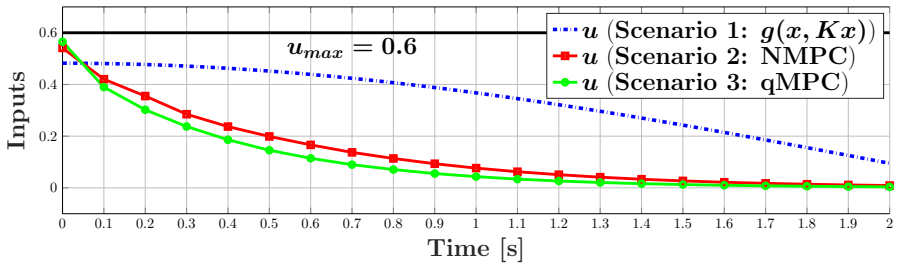


Figure 6.3.: Values of the input u under different scenarios.

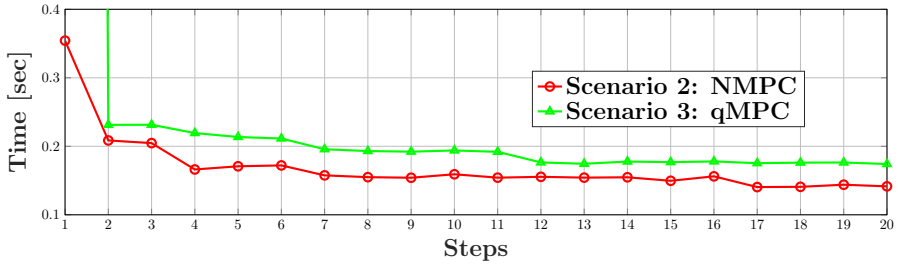


Figure 6.4.: Computing time for the two Scenarios 2 and 3.

according to the user's interest since the design procedure decouples the terminal region and terminal weighting matrix design. While in qMPC, the terminal weighting matrix P_q is defined from the beginning and also employed as the weighting matrix of its ellipsoid terminal region $\Omega_\alpha = \{\boldsymbol{x} \in \mathbb{R}^n | \ \boldsymbol{x}^\top P_q \boldsymbol{x} \leq \alpha\}$.

Moreover, our proposed construction particularized for the CTC controller including: i) the upper bound of the CTC law through Taylor's approximation, ii) the invariant set construction by employing the linear system and iii) the bound of the input term, can be generalized for any feedback linearization controller. Hence, using these results various applications of nonlinear controllers on stabilizing NMPC schemes can be conducted. Note that, taking into account uncertainties which may affect the system is out of the scope of this paper since we concentrated on the performance of the proposed approach w.r.t. existing methods and it will be consider for future work.

The next sections will present in more detail the results from **J13** where an elaborated feedback linearization controller is employed in NMPC design for the control of the translation dynamics of a multicopter system.

6.4. Multicopter dynamics and NMPC setups

6.4.1. System modeling

This section recapitulates the discrete translation dynamics of a standard thrust-propelled aircraft system. The dynamics are obtained by applying the Runge–Kutta fourth-order discretization method (Butcher and Wanner 1996) to the continuous thrust-propelled system found in **J9** and are expressed in their state-space representation as follows:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) = A\boldsymbol{x}_k + h_\psi(\boldsymbol{u}_k), \tag{6.19}$$

where the state $\boldsymbol{x}_k \in \mathbb{R}^6$ gathers the position and the velocity at the time step k, $\boldsymbol{x}_k \triangleq [x_k \ y_k \ z_k \ v_{x_k} \ v_{y_k} \ v_{z_k}]^\top$, the input $\boldsymbol{u}_k \in \mathbb{R}^3$, $\boldsymbol{u}_k \triangleq [T_k \ \phi_k \ \theta_k]^\top$ consists of the normalized thrust T (the normal thrust divided by the system's mass), the roll, ϕ, and pitch, θ, angles. The yaw angle $\psi \in [-\pi, \pi]$ is an assumed known constant influencing the system (usually considered to be zero as in (H. Lu, C. Liu, L. Guo, and W.-H. Chen 2017) but here we aim to exploit a general fix value of ψ). Explicitly, the matrix A and the input terms $h_\psi(\boldsymbol{u}_k)$ are given by:

$$A = \begin{bmatrix} \boldsymbol{I}_3 & \Delta_t \boldsymbol{I}_3 \\ 0_{3\times 3} & \boldsymbol{I}_3 \end{bmatrix}, \tag{6.20}$$

$$h_\psi(\boldsymbol{u}_k) = \begin{bmatrix} \frac{\Delta_t^2}{2}\boldsymbol{I}_3 \\ \Delta_t \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} T_k(\cos\phi_k \sin\theta_k \cos\psi + \sin\phi_k \sin\psi) \\ T_k(\cos\phi_k \sin\theta_k \sin\psi - \sin\phi_k \cos\psi) \\ -g + T_k \cos\phi_k \cos\theta_k \end{bmatrix}, \tag{6.21}$$

where Δ_t is the discretization time step and g stands for the gravitational acceleration[3]. Note that, the discrete system (6.19) is linear in the state since the original continuous system taken from **J9** already possesses this property. The vehicle has to respect thrust limits, with $T_{limit} > g$, and roll and pitch constraints as follows:

$$\boldsymbol{u}(k) \in \mathcal{U} = \{0 \leq T_k \leq T_{limit}, |\phi_k| \leq \epsilon_c, |\theta_k| \leq \epsilon_c\}, \tag{6.22}$$

with $\epsilon_c \in (0, \pi/2)$, the desired maximum value of the angles.
Then, without loss of generality, we aim to stabilize the system (6.19) around the equilibrium point:

$$\boldsymbol{x}_e = \boldsymbol{0}, \ \boldsymbol{u}_e = [g \ 0 \ 0]^\top. \tag{6.23}$$

[3]We have used in (6.21) "c" and "s" to denote the $\cos(\cdot)$ and $\sin(\cdot)$ functions, respectively.

Note that, the thrust-propelled system (6.19) can be stabilized at any position $\boldsymbol{x}'_e \in \mathbb{R}^3$ with zero velocity by simply choosing the origin of the coordinate system at $\boldsymbol{x}'_e$.

6.4.2. NMPC with and without terminal constraint

Model Predictive Control (MPC) is a control strategy in which at each time step, an open-loop optimal control problem of prediction horizon N, subject to the system dynamics (6.19) and constraints on states (including the current state as the initial condition) and inputs (6.22), is solved to obtain an optimal control sequence. From the sequence, only the first element is applied to the system. At the next sampling instant, the state is measured again and introduced to the optimization problem. The process is iteratively executed to establish the closed-loop controlled system (Mayne, Rawlings, Rao, and Scokaert 2000). Hereinafter, we introduce two different NMPC setups, i.e., with and without terminal constraint for the considered dynamics (6.19) which are adopted from their general designs given in (Grüne 2012; H. Chen and Allgöwer 1998). We also recapitulate the corresponding design principles for guaranteeing the stability of the two NMPC schemes (Grüne 2012; Mayne, Rawlings, Rao, and Scokaert 2000).

NMPC with terminal constraint (NMPC-t)

The NMPC design detailed hereby employs both the terminal cost and the terminal constraint[4]. For more details, the open-loop optimization problem $V_{T,N_T}(\boldsymbol{x}_k)$ at time step k with the measured state $\boldsymbol{x}_k$ is solved over the prediction horizon N_T:

$$V_{T,N_T}(\boldsymbol{x}_k) := \min_{\boldsymbol{u}(\cdot|k)} J_{T,N_T}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k)), \tag{6.24}$$

subject to

$$\boldsymbol{x}(i+1|k) = f(\boldsymbol{x}(i|k), \boldsymbol{u}(i|k)), \tag{6.25a}$$
$$\boldsymbol{u}(i|k) \in \mathcal{U},\ i \in \{0, \dots, N_T - 1\}, \tag{6.25b}$$
$$\boldsymbol{x}(0|k) = \boldsymbol{x}_k,\ \boldsymbol{x}(N_T|k) \in \mathcal{X}_f, \tag{6.25c}$$

with the cost function $J_{T,N_T}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k))$ defined as:

$$J_{T,N_T}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k)) = \sum_{i=0}^{N_T-1} \ell(\boldsymbol{x}(i|k), \boldsymbol{u}(i|k)) + F(\boldsymbol{x}(N_T|k)).$$

The predicted state and input sequences employed for the optimization problem at time step k are denoted by $\boldsymbol{x}(\cdot|k) \triangleq [\boldsymbol{x}(0|k)\ \dots\ \boldsymbol{x}(N_T|k)]$ and $\boldsymbol{u}(\cdot|k) \triangleq [\boldsymbol{u}(0|k)\ \dots\ \boldsymbol{u}(N_T-1|k)]$, respectively. The terminal region $\mathcal{X}_f \subset \mathbb{R}^{6\times 6}$ is designed according to the stability conditions introduced hereinafter. Furthermore, the stage cost, $\ell(\boldsymbol{x}, \boldsymbol{u})$, and terminal cost, $F(\boldsymbol{x})$, are defined as:

$$\ell(\boldsymbol{x}, \boldsymbol{u}) = \|\boldsymbol{x} - \boldsymbol{x}_e\|_Q^2 + \|\boldsymbol{u} - \boldsymbol{u}_e\|_R^2, \tag{6.26}$$
$$F(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_e\|_P^2, \tag{6.27}$$

in which, the symmetric matrices $Q \in \mathbb{R}^{6\times 6}$ (positive definite) and $R \in \mathbb{R}^{3\times 3}$ (positive semi-definite) are defined by the users, while the positive definite matrix $P \in \mathbb{R}^{6\times 6}$ is a solution of the Lyapunov equation introduced later (see (6.50)). Note that, the formulation of $\ell(\cdot)$ in (6.26) allows $R = \mathbf{0}$ as similar to (Köhler, Müller, and Allgöwer 2018; Limón, Alamo, Salas, and Camacho 2006). Assuming that the optimization problem (6.24)-(6.25) is successfully solved, it provides the optimal state and input trajectories $(\boldsymbol{x}^*(\cdot|k), \boldsymbol{u}^*(\cdot|k))$. Then, the MPC control action applied to the system (6.19) at

[4] Usually referred as *quasi-infinite horizon NMPC* (H. Chen and Allgöwer 1998).

time step k is defined as:

$$\mu_T(\boldsymbol{x}_k) = \boldsymbol{u}_T^*(0|k). \tag{6.28}$$

Finally, the nominal closed-loop system is given by:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \mu_T(\boldsymbol{x}_k)). \tag{6.29}$$

Hereinafter, we summarize the NMPC principles given in (Grüne and Pannek 2011; H. Chen and Allgöwer 1998) to establish the sufficient conditions for the stability of the nominal closed-loop system (6.29).

Assumption 6.1. Consider the closed terminal set $\mathcal{X}_f$ associated to (6.24) and the terminal cost F from (6.27). We assume that for each $\boldsymbol{x}_k \in \mathcal{X}_f$ at any time step k, there exists an admissible control value[5] $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \in \mathcal{U}$ such that the two following conditions hold:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)) \in \mathcal{X}_f, \tag{6.30}$$
$$F(\boldsymbol{x}_{k+1}) + \ell(\boldsymbol{x}_k, \boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)) \leq F(\boldsymbol{x}_k). \tag{6.31}$$

Theorem 6.1. *Let Assumption 6.1 hold. Then, the nominal closed-loop system (6.29) under the feedback law μ_T (6.28) from the NMPC-t scheme (6.24)-(6.25) achieves recursive feasibility[6] and asymptotic stability.* □

Proof. See Chapter 5 in (Grüne and Pannek 2011) and Theorem 1 in (H. Chen and Allgöwer 1998). □

NMPC without terminal constraint (u-NMPC)

The NMPC scheme without terminal cost and terminal stabilizing constraint is also referred as unconstrained NMPC as in (Reble and Allgöwer 2012; Grüne and Pannek 2011), hence, being denoted by u-NMPC within the paper. The following optimization control problem $V_{U,N_U}(\boldsymbol{x}_k)$ at time k is solved over the prediction horizon N_U:

$$V_{U,N_U}(\boldsymbol{x}_k) := \min_{\boldsymbol{u}(\cdot|k)} J_{U,N_U}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k)), \tag{6.32}$$

subject to

$$\boldsymbol{x}(i+1|k) = f(\boldsymbol{x}(i|k), \boldsymbol{u}(i|k)), \tag{6.33a}$$
$$\boldsymbol{u}(i|k) \in \mathcal{U},\ i \in \{0, \dots, N_U - 1\}, \tag{6.33b}$$
$$\boldsymbol{x}(0|k) = \boldsymbol{x}_k, \tag{6.33c}$$

with the cost function $J_{U,N_U}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k))$ defined in terms of the stage cost ℓ from (6.26) as:

$$J_{U,N_U}(\boldsymbol{x}_k, \boldsymbol{u}(\cdot|k)) = \sum_{i=0}^{N_U-1} \ell(\boldsymbol{x}(i|k), \boldsymbol{u}(i|k). \tag{6.34}$$

Then, the u-NMPC control action and the nominal closed-loop system at time k are given by:

$$\mu_U(\boldsymbol{x}_k) = \boldsymbol{u}_U^*(0|k), \tag{6.35}$$
$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \mu_U(\boldsymbol{x}_k)), \tag{6.36}$$

with $\boldsymbol{x}_U^*(\cdot|k)$ the optimal input trajectories. The stability of system (6.29) is well studied and presented in (Köhler, Müller, and Allgöwer 2018; Grüne and Pannek 2011) and summarized hereinafter.

[5] The notation $\boldsymbol{u}_{\text{loc}}$ stands for local controller, i.e., being locally admissible.
[6] The initial iteration successfully executed implies the feasibility of all the further steps.

Assumption 6.2. There exist constants $\gamma, c \in \mathbb{R}_{>0}$ such that for any $N_U \geq 2$ and for all the initial state $\boldsymbol{x}_0$ satisfying $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\|_Q^2 \leq c$, we have:

$$V_{U,N_U}(\boldsymbol{x}_k) \leq \gamma \|\boldsymbol{x}_0 - \boldsymbol{x}_e\|_Q^2. \tag{6.37}$$

Theorem 6.2. *Let Assumption 6.2 hold. Then, there exists an $N_0 \in \mathbb{N}$, such that for the prediction horizon length $N_U \geq N_0$, the equilibrium point $\boldsymbol{x}_e$ is uniformly exponentially stable under the nominal closed-loop dynamics (6.29) for any initial state $\boldsymbol{x}_0$ satisfying $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$.* □

Proof. See Theorem 1 in (Köhler, Müller, and Allgöwer 2018) and Theorem 3.6 in Grüne 2012. At first, in (Köhler, Müller, and Allgöwer 2018), the authors show that $V_{N,U}(\boldsymbol{x}_k) \leq c\gamma$ implies that $V_{U,N_U}(\boldsymbol{x}_k) \leq \gamma \|\boldsymbol{x}_k - \boldsymbol{x}_e\|_Q^2$ with a case dictinction based on whether $\|\boldsymbol{x}_k - \boldsymbol{x}_e\|_Q^2 \leq c$ or not (Köhler, Müller, and Allgöwer 2018).
Secondly, whenever $V_{U,N_U}(\boldsymbol{x}_k) \leq \gamma \|\boldsymbol{x}_k - \boldsymbol{x}_e\|_Q^2$ holds, in (Grüne 2012), the authors show that $V_{U,N_U}(\boldsymbol{x}_k)$ decreases for all $N_U \geq N_0$ with N_0 given by (see Variant 3 in (Grüne 2012) for more details):

$$N_0 = 2 + \frac{\ln(\gamma - 1)}{\ln \gamma - \ln(\gamma - 1)}. \tag{6.38}$$

Lastly, the recursive feasibility and exponential stability are obtained with the initial condition $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$. □

Remark 6.1. One can impose state constraints into the two NMPC controllers by adding the following state condition into (6.25b) and (6.33b):

$$\boldsymbol{x}(i|k) \in \mathcal{X}, \tag{6.39}$$

with $\mathcal{X} \subseteq \mathbb{R}^6$, the non-empty state constraint set required to be convex and to contain the equilibrium point $\boldsymbol{x}_e$. Then, Theorems 6.1 and 6.2 accordingly require $\mathcal{X}_f \subseteq \mathcal{X}$ and $\boldsymbol{x}_k \in \mathcal{X}, \forall \|\boldsymbol{x}_k - \boldsymbol{x}_e\|_Q^2 \leq c$. Within this paper, we do not consider state constraints for the thrust-propelled dynamics (6.19) even though the proposed approaches provide the means to scale the terminal region $\mathcal{X}_f$ (by tuning its radius as will be discussed in Section 6.5) and the set of $\|\boldsymbol{x}_0 - \boldsymbol{x}_e\|_Q^2 \leq c$ (by increasing the value of Q). This is due to the fact that we want to analyze the maximum performance (convergence speed - shown by the velocity) of the NMPC controllers. Furthermore, the initial condition has to be chosen close to the equilibrium point for guaranteeing the stability (c.f. Theorem 6.2) due to hardware limitations (with largest possible prediction horizon of only 10 steps), hence, realistic constraints on position are never activated under our experimental tests. □

6.4.3. Motivation

Assumptions 6.1 and 6.2, which are required to design the two NMPC constructions provided in Sections 6.6.1 and 6.6.2 are hard to validate for a generic nonlinear system given as in (6.25a) and (6.33a). In the literature, the standard approach is to linearize the dynamics around the equilibrium point, from which it follows the construction of a linear controller and its associated invariant set (of ellipsoidal form as in (Köhler, Müller, and Allgöwer 2018; H. Chen and Allgöwer 1998) or polyhedral form as in (Cannon, Deshmukh, and Kouvaritakis 2003)). However, employing a linear controller for a general nonlinear system (6.25a),(6.33a) obviously restricts the corresponding invariant set (also serving as the terminal region $\mathcal{X}_f$ in (6.25)), hence, arguably reducing the efficiency of the NMPC-t controller (6.24)-(6.28) and leading to an impractically large prediction horizon for the u-NMPC controller (6.32)-(6.35).
Thus, it is worthwhile to ask whether, for particular dynamics (as those shown in (6.19)), we may dispense with the linearized dynamics/linear controller construction and, instead, check Assumptions 6.1 and 6.2 by applying a feedback linearization controller.
In our opinion this helps not only to account for the nonlinearities of the system (6.19) but also to apply

various tools available for linear dynamics when considering the resulting closed-loop linear system (i.e., invariant set construction becomes much easier, as the difficulties are "pushed" in the nonlinear input). As is to be expected, the simplified closed-loop dynamics come at the price of a complex input formulation which, in turn, makes the input constraint and stage cost $\ell(\boldsymbol{x}, \boldsymbol{u})$ in (6.22),(6.26) formulations difficult to handle. These difficulties will be tackled by employing previously obtained results of the authors **J9**, **C46** and which will be recapitulated in the forthcoming sections.

6.5. Feedback linearization law and input constraint satisfaction

This section recapitulates the results of a feedback linearization controller for quadcopter trajectory tracking **J9**, **C39** which will be employed as the local controller for guaranteeing the stability of the two NMPC controllers detailed in Section 6.4.2. In general, feedback linearization control approach reduces the controlled system to an equivalent linear system (assuming no mismatches on state feedback and system dynamics, as similar to nominal NMPC applications in (6.28) and (6.35)). Thus, the method does not require any approximation and does not restrict the system to operate only in the neighborhood of the linearization point (J. J. Craig 2005). Let us start by introducing the standard feedback linearization law $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \triangleq \left[T_{\text{loc}}(\boldsymbol{x}_k)\ \phi_{\text{loc}}(\boldsymbol{x}_k)\ \theta_{\text{loc}}(\boldsymbol{x}_k)\right]^\top$ defined as follows (more details can be found in **J9**, **C39**):

$$T_{\text{loc}}(\boldsymbol{x}_k) = \sqrt{u_x^2(\boldsymbol{x}_k) + u_y^2(\boldsymbol{x}_k) + (u_z(\boldsymbol{x}_k) + g)^2}, \tag{6.40a}$$

$$\phi_{\text{loc}}(\boldsymbol{x}_k) = \arcsin\left(\frac{u_x(\boldsymbol{x}_k)\sin\psi - u_y(\boldsymbol{x}_k)\cos\psi}{T_{\text{loc}}(\boldsymbol{x}_k)}\right), \tag{6.40b}$$

$$\theta_{\text{loc}}(\boldsymbol{x}_k) = \arctan\left(\frac{u_x(\boldsymbol{x}_k)\cos\psi + u_y(\boldsymbol{x}_k)\sin\psi}{u_z(\boldsymbol{x}_k) + g}\right), \tag{6.40c}$$

where the virtual control inputs $u_q(\boldsymbol{x}_k)$ with $q \in \{x, y, z\}$ are calculated in terms of the positions and velocities (i.e., $\boldsymbol{x}_k \triangleq [x_k\ y_k\ z_k\ v_{x_k}\ v_{y_k}\ v_{z_k}]^\top$ from (6.19)):

$$u_q(\boldsymbol{x}_k) = K_{1_q} q_k + K_{2_q} v_{q_k}, \tag{6.41}$$

with $K_{1_q}, K_{2_q} \in \mathbb{R}$ the control gains. If the feedback linearization control action $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ as in (6.40) is not affected by the saturation limit, i.e., $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \in \mathcal{U}$ with $\mathcal{U}$ in (6.22), then $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ transforms the dynamics (6.19) into the following linear system:

$$\boldsymbol{x}_{k+1} = A_{cl}\boldsymbol{x}_k, \tag{6.42}$$

with the matrix $A_{cl} \in \mathbb{R}^{6\times 6}$ given by:

$$A_{cl} = \begin{bmatrix} \boldsymbol{I}_3 + \frac{\Delta_t^2}{2}K_1 & \Delta_t \boldsymbol{I}_3 + \frac{\Delta_t^2}{2}K_2 \\ \Delta_t K_1 & \boldsymbol{I}_3 + \Delta_t K_2 \end{bmatrix}, \tag{6.43}$$

and with $K_1 = \text{diag}(K_{1_x}, K_{1_y}, K_{1_z})$ and $K_2 = \text{diag}(K_{2_x}, K_{2_y}, K_{2_z})$. Furthermore, the control gains K_{1_q}, K_{2_q} are required to satisfy the following conditions:

$$-\frac{2}{\Delta_t} < K_{2_q} < \frac{\Delta_t}{2}K_{1_q} < 0, \tag{6.44}$$

which serve to establish the asymptotic stability of the resulted linear system (6.42) according to the Routh-Hurwitz criterion. Then, A_{cl} is a Schur matrix having all its eigenvalues strictly inside the unit circle.

However, the control action $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ has to respect the input constraints (i.e., $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k) \in \mathcal{U}$ (6.22)) for the linearization to be valid. Thus, hereinafter, we will introduce some conditions on the virtual

control inputs $u_x(\boldsymbol{x}_k)$, $u_y(\boldsymbol{x}_k)$ and $u_z(\boldsymbol{x}_k)$ in (6.41) to ensure the input constraint satisfaction.

Proposition 6.1. *By choosing the values of the three positive saturation limits U_x, U_y and U_z such that:*

$$U_z < g, \tag{6.45}$$
$$U_x^2 + U_y^2 \leq (-U_z + g)^2 \tan^2 \epsilon_c, \tag{6.46}$$
$$\sqrt{U_x^2 + U_y^2 + (U_z + g)^2} \leq T_{limit}, \tag{6.47}$$

we have that, if the virtual inputs (6.41) satisfy $|u_x(\boldsymbol{x}_k)| \leq U_x$, $|u_y(\boldsymbol{x}_k)| \leq U_y$ and $|u_z(\boldsymbol{x}_k)| \leq U_z$, then, the followings hold:
(i) the feedback linearization control action $\boldsymbol{u}_{loc}(\boldsymbol{x}_k) \in \mathcal{U}$ (6.40), for all value of $\psi \in [-\pi, \pi]$;
(ii) the values of $\|\boldsymbol{u}_{loc}(\boldsymbol{x}_k) - \boldsymbol{u}_e\|^2$ are bounded by:

$$\|\boldsymbol{u}_{loc}(\boldsymbol{x}_k) - \boldsymbol{u}_e\|^2 \leq \boldsymbol{x}_k^\top (K^\top K + 2\Gamma)\boldsymbol{x}_k, \tag{6.48}$$

where $K = [K_1 \ K_2]$ (6.42) and $\Gamma \in \mathbb{R}^{6\times 6}$ is given by:

$$\Gamma = \frac{1}{(-U_z + g)^2} K_{xy}^\top K_{xy}, \tag{6.49}$$

with $K_{xy} = \big[\text{diag}(K_{1_x}, K_{1_y}, 0) \ \ \text{diag}(K_{2_x}, K_{2_y}, 0)\big]$. □

Proof. See **C46**. □

By Proposition 6.1, we guarantee the input constraint satisfaction for the feedback linearization controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ (6.40) for the whole region of $|u_q(\boldsymbol{x_k})| \leq U_q, \forall q \in \{x, y, z\}$ (6.41). Hereinafter, we will construct an ellipsoidal invariant set within this region, thus, being able to employ a simple analysis resulted from the linear system (6.42).
Let us consider a symmetric positive definite matrix $M \in \mathbb{R}^{6\times 6}$. We obtain the matrix $P \in \mathbb{R}^{6\times 6}$, also being symmetric positive definite, as the unique solution of the following Lyapunov equation:

$$A_{cl}^\top P A_{cl} = P - M. \tag{6.50}$$

with the Schur matrix A_{cl} from (6.42).

Proposition 6.2. *Let us consider the set $\mathcal{X}_f$ defined as follows:*

$$\mathcal{X}_f = \{\boldsymbol{x} \in \mathbb{R}^6 | \ \|\boldsymbol{x}\|_P^2 \leq \delta\}, \tag{6.51}$$

with δ given by:

$$\delta = \lambda_{min}(P) r^2, \ \text{with } r^2 = \min_{q\in\{x,y,z\}} \left\{ \frac{U_q^2}{K_{1_q}^2 + K_{2_q}^2} \right\}, \tag{6.52}$$

and U_q from (6.45)-(6.47). Then, for all $\boldsymbol{x}_k \in \mathcal{X}_f$, we have that:
(i) the feedback linearization controller $\boldsymbol{u}_{loc}(\boldsymbol{x}_k) \in \mathcal{U}$ (6.40), for all $\psi \in [-\pi, \pi]$;
(ii) $\mathcal{X}_f$ is forward invariant[7] under $\boldsymbol{u}_{loc}(\boldsymbol{x}_k)$. Furthermore, $\boldsymbol{x}_{k+1}$ satisfies:

$$\|\boldsymbol{x}_{k+1}\|_P^2 \leq \rho \|\boldsymbol{x}_k\|_P^2, s.t. \ \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_{loc}(\boldsymbol{x}_k)), \tag{6.53}$$

with $\rho \in (0, 1)$ given by:

$$\rho = 1 - \lambda_{min}(M)/\lambda_{max}(P). \tag{6.54}$$

[7] A sub-level set of a Lyapunov function is a forward invariant set.

□

Proof. See **J13**. □

6.6. Stability of different NMPC schemes

In the following, we gather all the results obtained from the feedback linearization controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ (6.40) to design two NMPC schemes with and without terminal constraint. We check that Assumptions 6.1 and 6.2 given in Section 6.4.1 hold, hence, guaranteeing the asymptotic stability property for stabilizing the thrust-propelled system (6.19).

6.6.1. NMPC with terminal constraint (NMPC-t)

Proposition 6.3. *Consider the setup of the NMPC-t scheme (6.26)-(6.28) using the terminal region $\mathcal{X}_f$ in (6.51) with the matrix P from the Lyapunov equation (6.50). If the symmetric matrix M in (6.50) is chosen such that $M \succeq Q^*$ with the matrix Q^* verifying:*

$$Q^* = Q + \lambda_{max}(R)(K^\top K + 2\Gamma), \tag{6.55}$$

with Q and R from (6.26), K and Γ from (6.49), then, Assumption 6.1 is satisfied with the feedback linearization controller $\boldsymbol{u}_{loc}(\boldsymbol{x}_k)$ from (6.40). Thus, the NMPC-t scheme (6.26)-(6.28) guarantees the recursive feasibility and the (nominal) asymptotic stability for the closed-loop dynamics (6.29). □

Proof. See **J13**. □

Hereinafter, let us summarize the design procedure of the NMPC scheme with terminal constraint given in (6.26)–(6.28) for stabilizing the dynamics (6.19).

Procedure 6.1. **NMPC-t design**

1. Choose the positive definite symmetric matrices $Q \in \mathbb{R}^{6\times 6}$ and $R \in \mathbb{R}^{3\times 3}$ to formulate the stage cost in (6.26).
2. Choose the saturation limits U_x, U_y and U_z satisfying (6.45)–(6.47).
3. Choose the control gains K_{1_q}, K_{2_q} with $q \in \{x, y, z\}$ satisfying (6.44).
4. Find the matrix Q^* in (6.55), then, define the symmetric matrix $M \succeq Q^*$ and solve the Lyapunov equation (6.50) for P.
5. Find δ in (6.52) to obtain the terminal region $\mathcal{X}_f$ in (6.51).
6. Define the prediction horizon N_T such that the first NMPC iteration w.r.t. the initial state $\boldsymbol{x}_0$ is feasible. ■

Remark 6.2. For a predefined initial state $\boldsymbol{x}_0$, Step 6 in Procedure 6.1 requires the users to run the NMPC-t algorithm once, while progressively increasing prediction horizon until the optimization problem becomes feasible. However, note that, the computation time is greatly affected by any increase of the prediction length. Thus, in order to continue increasing the region of attraction when the prediction horizon N_T is already large, one can increase the size of the terminal constraint set $\mathcal{X}_f$ given in (6.51) instead. The parameters which affect most the size of the terminal region $\mathcal{X}_f$ (6.51) are the control gains K_{1_q}, K_{2_q} with $q \in \{x, y, z\}$ in (6.44). In general, decreasing the magnitudes of the control gains increase the size of the set $\mathcal{X}_f$ and vice versa. Therefore, when state constraints are considered as mentioned in Remark 6.1, the control gains K_{1_q}, K_{2_q} with $q \in \{x, y, z\}$ are required to be appropriately chosen such that $\mathcal{X}_f \subseteq \mathcal{X}$ with the state constraint set $\mathcal{X}$ from (6.39). □

6.6.2. NMPC without terminal constraint (u-NMPC)

Proposition 6.4. *Consider the translational thrust-propelled dynamics (6.19) and the corresponding u-NMPC setup (6.32)-(6.36). By choosing the matrix $M \in \mathbb{R}^{6\times 6}$ used in (6.50) being symmetric positive definite and setting the constants $c, \gamma \in \mathbb{R}_{>0}$ from (6.37) as:*

$$c = \delta \frac{\lambda_{min}(Q)}{\lambda_{max}(P)}, \;\; \gamma = \frac{\lambda_{max}(Q^*)\lambda_{max}(P)}{\lambda_{min}(Q)\lambda_{min}(P)(1-\rho)}, \tag{6.56}$$

with δ in (6.52), Q in (6.26), Q^* in (6.55), ρ in (6.54) and P satisfying (6.50), then, Assumption 6.2 is satisfied. Furthermore, the equilibrium point $\boldsymbol{x}_e$ in (6.23) is uniformly exponentially stable under the nominal closed-loop dynamics (6.36) for all the initial states $\boldsymbol{x}_0$ satisfying $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$. □

Proof. See **J13**. □

In the following, the design procedure of the u-NMPC controller in (6.32)–(6.35) is summarized.

Procedure 6.2. **u-NMPC design**

1) 2) 3) Similar to the first three steps of Procedure 6.1.

4) Choose the symmetric positive definite matrix M in (6.50) by following the analysis detailed in Section 6.7 and solve the Lyapunov equation (6.50) for P.

5) Find δ as in (6.52), ρ as in (6.53), c and γ as in (6.56) in order to find N_0 given in (6.38).

6) Define the prediction horizon $N_U \geq N_0$ as in (6.38). ■

Remark 6.3. The region of attraction of the u-NMPC design (6.32)-(6.35) which consists of all the initial states $\boldsymbol{x}_0$ satisfying $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$ is difficult to estimate since it does not show an explicit requirement on the initial state $\boldsymbol{x}_0$ but requires solving the first u-NMPC iteration. Thus, the usual approach in the literature (Köhler, Müller, and Allgöwer 2018; Grüne 2012) is to employ the condition $\|\boldsymbol{x}_0\|_Q^2 \leq c$ instead since $V_{U,N_U}(\boldsymbol{x}_0) \leq \gamma\|\boldsymbol{x}_0\|_Q^2$ holds. However, it is worth noting that the region of $\|\boldsymbol{x}_0\|_Q^2 \leq c$ is very conservative w.r.t. the original region of attraction satisfying $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$. □

6.7. Tuning the u-NMPC controller

It is well-known in the literature that the u-NMPC scheme requires a sophisticated tuning procedure in order to obtain the reasonable values of the required minimum prediction horizon N_0 and also the region of attraction which guarantee the stability (Köhler, Müller, and Allgöwer 2018; Grüne 2012). However, to the best of our knowledge, the tuning problems of the NMPC without terminal stabilizing constraints have been underestimated in various relating works (Köhler, Müller, and Allgöwer 2018; Grüne 2012; Reble and Allgöwer 2012; Limón, Alamo, Salas, and Camacho 2006). That is to say, people concentrate mostly on the stability proofs of their NMPC designs (e.g. as our contribution in Proposition 6.4), then, provide one illustrative example with specific parameters (Köhler, Müller, and Allgöwer 2018). Theses examples actually aim to show how the results are obtained (e.g. in order for the readers to validate again the calculation process by themselves) but do not give the insight into the actual tuning process. For our particular u-NMPC design (6.32)-(6.35), the most influential parameters are the control gains K_{1_q}, K_{2_q} with $q \in \{x, y, z\}$ satisfying (6.44), the matrix M from (6.50) and the weighting matrices Q, R from (6.26) which are not easy to tune due to a large amount of decision variables. In Table 6.1, we show that an arbitrary choice of these parameters can easily lead to an impractically large prediction horizon length while appropriate changes in the tuning parameters allow to reduce the required prediction horizon N_0 and to increase the domain of attraction (represented by $\|\mathsf{p}\|_Q^2 \leq c$ as in (6.37)).

In order to analyze how to tune the prediction horizon N_0 given in (6.38), we will mostly consider the

Description	Q	R	K_1	K_2	M	N_0	c
First choice	$\begin{bmatrix} \boldsymbol{I}_3 & 0_{3\times3} \\ 0_{3\times3} & 0.1\boldsymbol{I}_3 \end{bmatrix}$	$0.1\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$\boldsymbol{I}_6$	3121	0.024
Tuning R	$\begin{bmatrix} \boldsymbol{I}_3 & 0_{3\times3} \\ 0_{3\times3} & 0.1\boldsymbol{I}_3 \end{bmatrix}$	$0.01\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$\boldsymbol{I}_6$	2786	0.024
Tuning Q	$\boldsymbol{I}_6$	$0.01\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$-\boldsymbol{I}_3$	$\boldsymbol{I}_6$	176.1	0.2397
Tuning K_1, K_2	$\boldsymbol{I}_6$	$0.01\boldsymbol{I}_3$	$-1.2\boldsymbol{I}_3$	$-0.9\boldsymbol{I}_3$	$\boldsymbol{I}_6$	154.1	0.2438
Tuning M	$\boldsymbol{I}_6$	$0.01\boldsymbol{I}_3$	$-1.2\boldsymbol{I}_3$	$-0.9\boldsymbol{I}_3$	$\begin{bmatrix} \boldsymbol{I}_3 & 0.1\boldsymbol{I}_3 \\ 0.1\boldsymbol{I}_3 & 1.3\boldsymbol{I}_3 \end{bmatrix}$	153.1	0.2708

Table 6.1.: Prediction horizon length w.r.t different tuning parameters.

parameter γ as in (6.56). By introducing ρ from (6.54) to (6.56), we can express γ as the multiplication of γ_1 and γ_2 as follows:

$$\gamma = \underbrace{\frac{\lambda_{\max}(Q^*)}{\lambda_{\min}(Q)}}_{\gamma_1} \underbrace{\frac{\lambda^2_{\max}(P)}{\lambda_{\min}(P)\lambda_{\min}(M)}}_{\gamma_2}, \tag{6.57}$$

with P obtained from solving (6.50) with a symmetric positive definite matrix $M \in \mathbb{R}^{6\times6}$, Q in (6.26) and Q^* in (6.55). At first, from the formulation of N_0 given in (6.38), decreasing the value of γ as in (6.57) also reduces the prediction horizon N_0. Therefore, it is in our interest to minimize the values of γ_1 and γ_2 defined in (6.57). In order to reduce the decision variables, we follow some reasonable assumptions:

- We give equal importance to the motions along the three axes, i.e.:

$$Q = \text{diag}\{q_1, q_1, q_1, q_2, q_2, q_2\}, \tag{6.58}$$
$$K_{1_x} = K_{1_y} = K_{1_z} = k_1, \tag{6.59}$$
$$K_{2_x} = K_{2_y} = K_{2_z} = k_2. \tag{6.60}$$

  with $q_1, q_2 > 0$ and $k_1, k_2 < 0$ satisfying (6.44). Consequently, the matrix A_{cl} in (6.42) becomes:

$$A_{cl} = \begin{bmatrix} \left(1 + \frac{\Delta_t^2}{2}k_1\right)\boldsymbol{I}_3 & \Delta_t\left(1 + \frac{\Delta_t}{2}k_2\right)\boldsymbol{I}_3 \\ \Delta_t k_1 \boldsymbol{I}_3 & (1 + \Delta_t k_2)\boldsymbol{I}_3 \end{bmatrix}. \tag{6.61}$$

- The matrix $M \in \mathbb{R}^{6\times6}$ in (6.50) is parametrized as follows:

$$M = \begin{bmatrix} m_1\boldsymbol{I}_3 & m_3\boldsymbol{I}_3 \\ m_3\boldsymbol{I}_3 & m_2\boldsymbol{I}_3 \end{bmatrix}, \tag{6.62}$$

  with $m_1 = 1$ and $m_2, m_3, m_4 \in \mathbb{R}$ satisfying the following conditions:

$$m_1 m_2 > m_3^2, \tag{6.63}$$

  in which (6.63) is to guarantee the positive definiteness of M. With $m_1 = 1$, (6.63) is simplified to $m_2 > m_3^2$.

Remark 6.4. The reason for fixing $m_1 = 1$ in (6.62) is that any scaling on M results on the same scaling on P due to the Lyapunov equation (6.50). Then, all the eigenvalues of M and P are scaled

similarly, hence, γ_1 in (6.57) does not change and therefore, fixing $m_1 = 1$ does not cause any loss of generality within the analysis (but reduces the numerical issues). □

Next, the Lyapunov equation (6.50) is linear in P which provides a full-rank system of linear equations with A_{cl} and M from (6.61)-(6.62). Thus, by solving this, the matrix P is explicitly given as follows:

$$P = \begin{bmatrix} p_1 \boldsymbol{I}_3 & p_3 \boldsymbol{I}_3 \\ p_3 \boldsymbol{I}_3 & p_2 \boldsymbol{I}_3 \end{bmatrix}, \tag{6.64}$$

with p_1, p_2, p_3 calculated as:

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 1 - a_1^2 & -2a_1 a_3 & -a_3^2 \\ -a_2^2 & -2a_2 a_4 & 1 - a_4^2 \\ -a_1 a_2 & 1 - a_1 a_4 - a_2 a_3 & -a_3 a_4 \end{bmatrix}^{-1} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}, \tag{6.65}$$

in which, $a_1, \ldots, a_4$ are taken from $A_{cl} \triangleq \begin{bmatrix} a_1 \boldsymbol{I}_3 & a_2 \boldsymbol{I}_3 \\ a_3 \boldsymbol{I}_3 & a_4 \boldsymbol{I}_3 \end{bmatrix}$ with A_{cl} in (6.61). The eigenvalues of P from (6.64) are given by:

$$\lambda_{\substack{\max \\ \min}}(P) = \frac{1}{2}\left(p_1 + p_2 \pm \sqrt{(p_1 - p_2)^2 + 4p_3^2}\right). \tag{6.66}$$

Tuning γ_1

By introducing the matrix Q from (6.58) and the control gains k_1, k_2 as in (6.59)–(6.60) to the formulation of the matrix Q^*(6.55), we obtain the maximum eigenvalue of Q^* as follows:

$$\begin{aligned} \lambda_{\max}(Q^*) = \frac{1}{2}\Bigg(& q_1 + q_2 + \overline{r}(k_1^2 + k_2^2) \\ & + \underbrace{\sqrt{\left(q_1 - q_2 + \overline{r}(k_1^2 - k_2^2)\right)^2 + 4\overline{r}^2 k_1^2 k_2^2}}_{\leq \sqrt{2(q_1 - q_2)^2 + 2\overline{r}^2(k_1^4 + k_2^4)}} \Bigg), \end{aligned} \tag{6.67}$$

where $\overline{r}$ is directly proportional with the maximum eigenvalue of R:

$$\overline{r} = \lambda_{\max}(R)\left(1 + \frac{2}{(-U_z + g)^2}\right), \tag{6.68}$$

with g the gravitational acceleration and $U_z \in \mathbb{R}_+, U_z < g$ from Proposition 6.1. Then, γ_1 is obtained by introducing (6.58)-(6.67) into (6.57) and is bounded by:

$$\begin{aligned} & \frac{q_1 + q_2 + \overline{r}(k_1 + k_2)^2}{2\min(q_1, q_2)} \leq \gamma_1 \leq \\ & \frac{q_1 + q_2 + \overline{r}(k_1^2 + k_2^2) + \sqrt{2(q_1 - q_2)^2 + 2\overline{r}^2(k_1^4 + k_2^4)}}{2\min(q_1, q_2)}, \end{aligned} \tag{6.69}$$

in which, the first inequality is by introducing $\left(q_1 - q_2 + \overline{r}(k_1^2 - k_2^2)\right)^2 \geq 0$ to (6.67) and the latter one is by using the inequality given below (6.67). From (6.69), it can be observed that reducing the values of $|q_1 - q_2|$, $\lambda_{\max}(R)$, k_1^2 and k_2^2 probably provide a smaller value of γ_1 (we use "probably" since the reductions of the aforementioned parameters actually make both the upper and lower bounds as in (6.69) smaller). Especially, in case of employing $q_1 = q_2 = q$ with $q \in \mathbb{R}^+$, γ_1 from (6.57) is simplified

to γ_1^* as follows:

$$\gamma_1^* = 1 + \frac{\overline{r}(k_1^2 + k_2^2)}{q}, \tag{6.70}$$

which actually allows us to obtain a specific value of γ_1 (i.e., ≥ 1) by tuning only two weighting matrices Q and R from (6.26) regardless the predefined values of (k_1, k_2) as in (6.59)–(6.60). Therefore, for tuning γ_1 as in (6.57), we propose several general directions:

- decrease the ratio $\lambda_{\max}(Q)/\lambda_{\min}(Q)$ as much as possible.
- decrease the value of $\lambda_{\max}(R)$ based on the employed values of (k_1, k_2) but bear in mind that a small value of R causes large input consumption.
- decrease the values of k_1^2 and k_2^2. However, this is not encouraged since it can cause an unexpected increase in the value of γ_2 defined as in (6.57) and probably result in a larger prediction horizon N_0.

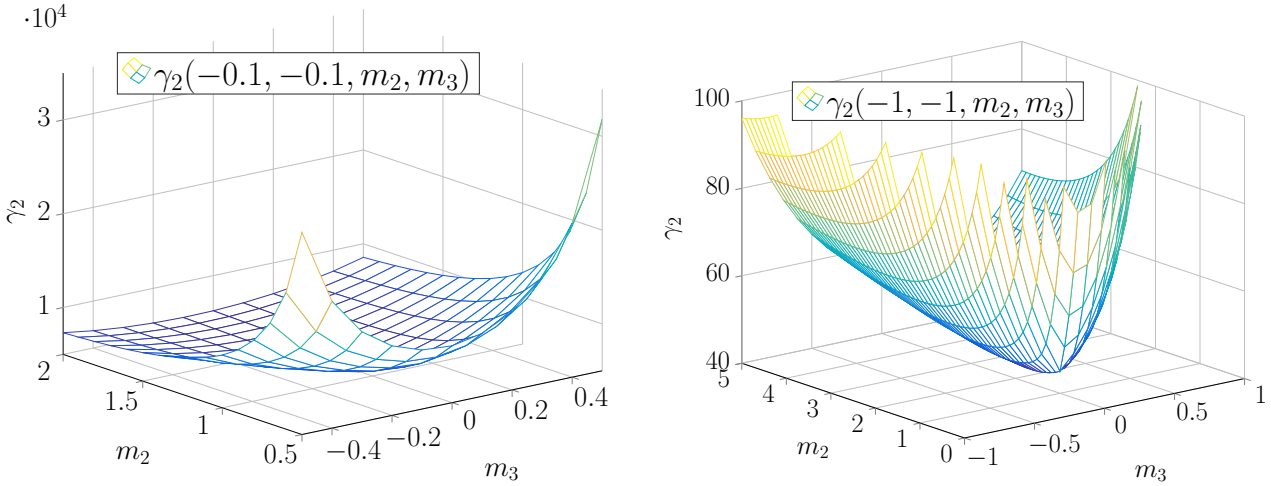
Remark 6.5. Formulation (6.70) explains the effects of the first two tuning steps given in Table 6.1: i) reducing $\lambda_{\max}(R)$ and ii) eliminating the term $|q_1 - q_2|$. However, it does not explain for the rest of Table 6.1 which requires us to analyze the parameter γ_2 as detailed in the following. □

Tuning γ_2

Our approach is to construct an explicit formulation of γ_2 from (6.57) by using the parametrizations of M as in (6.62) and P as in (6.66). The function is expressed in terms of the tuning parameters, i.e., the control gains k_1, k_2 from (6.59)–(6.60) and the scalars m_2, m_3 from (6.62) as follows:

$$\gamma_2 \triangleq \gamma_2(k_1, k_2, m_2, m_3). \tag{6.71}$$

It will be in our interest to analyze and to find the (local) minimums of the function $\gamma_2(\cdot)$ from (6.71).



(a) $(k_1, k_2) = (-0.1, -0.1)$ (b) $(k_1, k_2) = (-1, -1)$

Figure 6.5.: Illustration of $\gamma_2(k_1, k_2, m_2, m_3)$ as in (6.71) with different values of (k_1, k_2).

Since the function is strongly nonlinear, non-convex and contains up to four variables, we have to divide the task into two steps:

1. Find the optimal values of (m_2, m_3) which provide the local minimum value of γ_2 corresponding

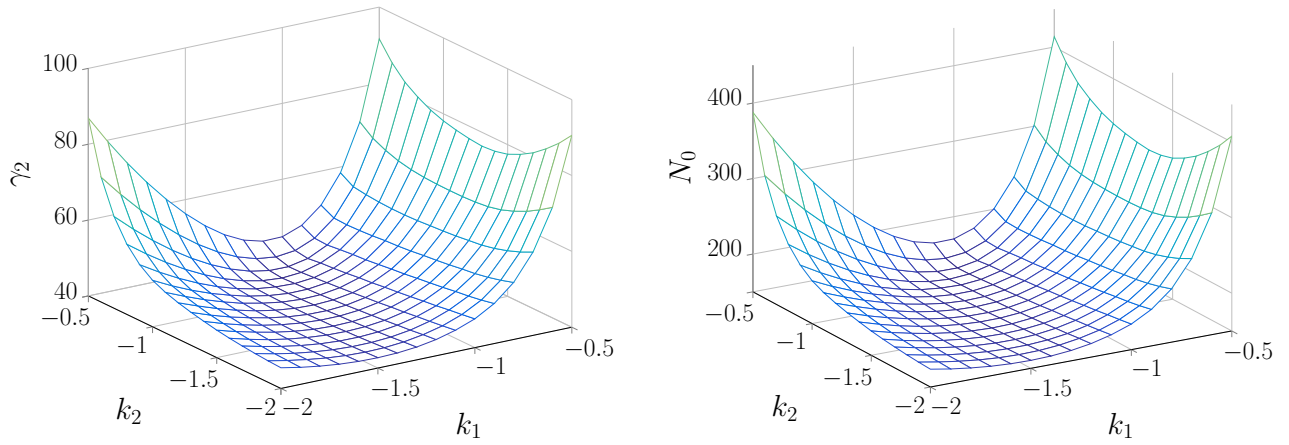
to a specific values of (k_1, k_2):

$$\gamma_2^*(k_1, k_2) = \min_{m_2, m_3} \gamma_2(k_1, k_2, m_2, m_3). \tag{6.72}$$
$$\text{subject to } m_2 > m_3^2 \text{ as required in (6.63)},$$
$$(m_2, m_3) \in [m_{2_{\min}}, m_{2_{\max}}] \times [-m_{3_{\max}}, m_{3_{\max}}],$$

with $(m_{2_{\min}}, m_{2_{\max}}, m_{3_{\max}})$ positive scalars specifying the ranges of m_2 and m_3 as in (6.62) which are defined accordingly to the condition $m_2 > m_3^2$ as required in (6.63) and are chosen by user. The results of this step are illustrated in Figure 6.5 for $(k_1, k_2) = (-0.1, -0.1)$ and $(-1, -1)$.

2. Apply Step 1 for different choices of (k_1, k_2), then, compare the obtained minimum values γ_2^* as in (6.72) in order to provide the complete analysis as illustrated in Figure 6.6.

Remark 6.6. The solution of the optimization problem in (6.72) is obtained by checking a mesh grid of the variables (m_2, m_3) within the specific range $[m_{2_{\min}}, m_{2_{\max}}] \times [-m_{3_{\max}}, m_{3_{\max}}]$, hence, the solution's accuracy depends on the resolution of the mesh grid. However, the accuracy problem is not critical due to the fact that we always have to choose the prediction horizon $N_U \geq N_0$ (c.f. Theorem 6.2) and $N_U \in \mathbb{N}$ (e.g., it does not matter if we obtain $N_0 = 152.4$ instead of the precise value assumed to be 152.36 since the minimum prediction horizon needs to be an integer, which leads in both case to $N_U = 153$). Furthermore, using the mesh grid method provides very fast computing times. E.g., it takes only 0.08 seconds to construct the whole data points for Figure 6.6 which contains 51×41 points of (m_2, m_3) and 16×16 points of (k_1, k_2). □



(a) Illustration of the optimal function $\gamma_2^*(k_1, k_2)$ defined in (6.72)

(b) N_0 obtained at the optimal choices of (m_2, m_3), $Q = \boldsymbol{I}_6$, $R = 0.01\boldsymbol{I}_3$

Figure 6.6.: Illustration of the analysis on N_0 from (6.38).

We enumerate in Table 6.2 all the scenarios in which, we obtain the smallest value of the required prediction horizon N_0 by using the weighting matrices $Q = \boldsymbol{I}_6$ and $R = 0.01\boldsymbol{I}_3$. Table 6.2 also shows the corresponding values of c from (6.56) which helps identifying all the feasible initial states p_0, i.e., $V_{U,N_U}(\boldsymbol{x}_0) \leq c\gamma$ as stated in Theorem 6.2. All the choices of (k_1, k_2, m_2, m_3) gathered in Table 6.2 require the minimum prediction horizon around $153 \sim 154$ but they provide a large range of c from 0.205 to 0.287. Furthermore, using the control gains $(k_1, k_2) = (-1.2, -0.8)$ and $M = \boldsymbol{I}_6$ provides the shortest prediction horizon $N_0 = 153$ and the largest value of $c = 0.287$, hence, being the best choice in our analysis.

For comparison, we use the method proposed in (Köhler, Müller, and Allgöwer 2018) which employs a linear controller as well as the linearized model of the dynamics (6.19). The results are given in the last three lines in which both the value of c and the prediction horizon N_0 are much more conservative than the results of our proposed method. This is due to the restriction of using the linear controller for

k_1 (6.59)	k_2 (6.60)	m_3 (6.62)	m_2 (6.62)	N_0 (6.38)	c (6.56)
-1.3	-1.1	0.2	1.6	153.8	0.206
-1.3	-1	0.1	1.3	153.5	0.215
-1.2	-0.9	0.1	1.3	153	0.272
-1.2	-0.8	0	1	152.2	0.287
-0.1	-0.1	Method in (Köhler, Müller, and Allgöwer 2018),		10^6	10^{-4}
-1	-1	using linear controller		5124	10^{-7}
-2	-2	and the linearized dynamics		1112	10^{-7}

Table 6.2.: Optimal values of (k_1, k_2, m_2, m_3) which provide the smallest N_0 in comparison with the method in (Köhler, Müller, and Allgöwer 2018) (using $Q = \boldsymbol{I}_6$ and $R = 0.01\boldsymbol{I}_3$ as in (6.34)).

the nonlinear system (6.19). This strongly confirms the effectiveness of our NMPC design approach using the local FL controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ as defined in (6.40).

We also notice that even the shortest prediction horizon in our analysis $N_0 = 153$ steps is still extremely large for a real implementation. However, bear in mind that the obtained results (c.f. Table 6.2) are still more promising than employing the linear controller as considered in (Köhler, Müller, and Allgöwer 2018). This also indicates a big gap which still exists between the theory on NMPC design and their practical formulations since through various simulations and experimental tests, the u-NMPC controller (6.34)–(6.35) requires a prediction horizon of only 10 steps to stabilize the system (6.19). This is due to the fact that the u-NMPC controller can fully exploit the inputs of the system while a standard controller (e.g. FL controller $\boldsymbol{u}_{\text{loc}}(\boldsymbol{x}_k)$ as in (6.40) or the linear controller as employed in (Köhler, Müller, and Allgöwer 2018)) enforces the inputs to follow their explicit formulations, hence, the convergence speed of the standard controller (e.g. ρ as in (6.53)) can not be fast enough to obtain this short prediction horizon of 10 steps. Also note that, better solutions can be found with more thorough parametrizations of the control gains (6.59)–(6.60) and of the matrix M (6.62). However, the trade-off is between increasing the computation time and the complexity of the optimization problem (6.72) (which may even cause infeasibility).

6.8. Simulation results

To validate the presented results, we consider the simulation model (6.19) of a Crazyflie 2.0 nano-quadcopter platform Budaciu, Botezatu, Kloetzer, and Burlacu 2019. It is assumed that the rotation dynamics are stabilized by a fast torque control mechanism **J9**. As specified in (6.22), the vehicle has to respect thrust limits, with $T_{limit} = 2g$, and roll and pitch constraints, i.e., $\epsilon_c = 10^\circ$ in (6.22)) which lead to the choice of $U_x = U_y = U_z = 1.0875$ which satisfy the conditions (6.45)-(6.46) as given in Tables 6.3 and 6.4. We fix the direction angle $\psi = 0$ and the initial state at $\boldsymbol{x}_0 = \begin{bmatrix} 0 & 0.15 & -0.1 & 0 & 0 & 0 \end{bmatrix}^\top$. Hereinafter, we introduce first the numerical values of the designs of the two NMPC schemes, i.e., with the terminal constraint (denoted by NMPC-t) and without the terminal constraint (u-NMPC) detailed in Sections 6.6.1 and 6.6.2, respectively. Their performances are analyzed and compared through simulation and experimental results. The NMPC optimization problems are implemented by using Pyomo **Pyomo** and solver IPOPT **ipopt** in Python 3.0. The sampling time is fixed at $\Delta_t = 0.1$ seconds with Δ_t as in (6.20)-(6.21) which means that the NMPC computing time should be less than $\Delta_t = 0.1$ seconds for making possible the application to real experiments.

6.8.1. NMPC setup

NMPC-t

The chosen parameters of the NMPC-t control scheme (6.26)–(6.28) are gathered into Table 6.3 (c.f. Procedure 6.1). We choose the control gains $K_{1_q} = -5$, $K_{2_q} = -5$ for all $q \in \{x, y, z\}$ (6.44) so that the corresponding region $\mathcal{X}_f$ is large enough for employing the prediction horizon $N_T = 5$ and the controlled system also has a good convergence speed.

Table 6.3.: Parameters of the NMPC-t controller.

	Values
Q, R in (6.26)	$\boldsymbol{I}_6$, $0.1\boldsymbol{I}_3$
U_x, U_y, U_z in (6.45)–(6.47)	1.0875, 1.0875, 1.0875
K_{1_q}, K_{2_q}, $q \in \{x, y, z\}$ in (6.44)	$-5, -5$
M in (6.50) ($M \succeq Q^*$ (6.55))	$20\boldsymbol{I}_6$
P in (6.50)	$\begin{bmatrix} 22.02\boldsymbol{I}_3 & 10.92\boldsymbol{I}_3 \\ 10.92\boldsymbol{I}_3 & 12.68\boldsymbol{I}_3 \end{bmatrix}$
δ in (6.52)	0.1285
N_T in (6.24)	5

u-NMPC

Table 6.4 presents the parameters of the u-NMPC control scheme (6.32)–(6.35). The values given in Table 6.4 are the best choices according to Section 6.7.

Table 6.4.: Parameters of the u-NMPC controller.

	Values
Q, R in (6.26)	$\boldsymbol{I}_6$, $0.01\boldsymbol{I}_3$
U_x, U_y, U_z in (6.45)–(6.47)	1.0875, 1.0875, 1.0875
k_1, k_2, in (6.59)–(6.60)	$-1.2, -0.8$
M in (6.50)	$\boldsymbol{I}_6$
P in (6.50)	$\begin{bmatrix} 18.22\boldsymbol{I}_3 & 4.02\boldsymbol{I}_3 \\ 4.02\boldsymbol{I}_3 & 12.22\boldsymbol{I}_3 \end{bmatrix}$
δ in (6.52), ρ in (6.54)	5.78, 0.95
c, γ in (6.56)	0.29, 40.96
N_0 in (6.38), N_U in (6.32)	$N_0 = 152.22$, $N_U = 153$

6.8.2. Simulation results

We provide the simulation results for four scenarios:
Scenario 1: Stabilize the dynamics (6.19) with $\psi = 0$ using the NMPC-t controller detailed in (6.26)–(6.28) with the parameters gathered in Table 6.3.
Using the u-NMPC controller given in (6.32)–(6.35) with the parameters in Table 6.4.
Scenario 3: Using the u-NMPC controller given in (6.32)–(6.35) with the prediction horizon of 10

steps (enough for stabilizing the system through simulations).
Scenario 4: Using the local feedback linearization controller $\boldsymbol{u}_{\text{loc}}$ given in (6.40) with the control gains as in Table 6.4.

The results are given in Figures 6.7–6.10 where they are plotted in red, green, blue and black corresponding to the four aforementioned scenarios, respectively. Figure 6.7 presents the convergence of the state trajectories under simulation. Note that, only the results of the y and z axes are plotted since the system does not move along the x axis. The terminal region $\mathcal{X}_f \subset \mathbb{R}^6$ given in (6.51) of the NMPC-t scheme is illustrated by the yellow ellipsoid (which is obtained by slicing $\mathcal{X}_f$ along three $2D$ spaces of the position and the corresponding velocity, then, making a convex union of the three obtained $2D$ sets). All the closed-loop state trajectories under four scenarios converge to the origin but with different convergence speeds as being illustrated with more details in Figure 6.8 and also given in Table 6.5. Also, the input results are shown in Figure 6.9 where all the proposed NMPC controllers respect the input constraints (6.22). We only provide the input results during the first one second for better clarity since they are all monotonously converging to the equilibrium $\boldsymbol{u}_e$ (6.23). Furthermore, Figure 6.10 and also Table 6.5 present the computing time per step of the three NMPC controllers employed under the first three scenarios.

Controller's information	Convergence time (95%) [s]	Computing time [ms]		
		Mean	Min	Max
Scenario 1	1.8	48.5	31.2	84.6
Scenario 2	3.1	243.7	209.3	298.4
Scenario 3	4.4	47.9	31.3	84.6
Scenario 4	2.1	Not applicable		

Table 6.5.: Comparison between different controllers.

From the presented simulation results, the NMPC-t controller employed under Scenario 1 obtains the shortest transient state (1.8 seconds as given in Table 6.5) due to its appropriate terminal constraint set $\mathcal{X}_f$ (yellow ellipsoid in Figure 6.7). This is also better than using the feedback linearization controller $\boldsymbol{u}_{\text{loc}}$ defined in (6.40) which provides the second shortest convergence time of 2.1 seconds. Next, when not using the terminal constraints, we can see that the u-NMPC controller under Scenario 2 (using the 153-step prediction horizon) obtains the transition time of 3.1 seconds while employing the u-NMPC controller with the prediction horizon of only 10 steps under Scenario 3 makes the states converging in longer time, i.e., 4.4 seconds. It is also trivial that the computation burden when using the 153-step prediction horizon under Scenario 2 is much heavier than those under other scenarios as can be seen from Figure 6.10. The average computing time of the NMPC controller with the 153-step prediction horizon is 243.7 milliseconds while it is only 47.9 second for the u-NMPC controller with the 10-step prediction horizon, being the best value in our analysis.
We also notice that the u-NMPC controller using the 153-step prediction horizon (which is found by our analysis given in Section 6.7) is not ready for real implementation since its minimum computing time is already 209.3 milliseconds while the sampling time is fixed at 100 milliseconds. However, the proposed calculation process based on the feedback linearization controller $\boldsymbol{u}_{\text{loc}}$ (6.40) already reduces significantly the required prediction horizon (153 steps) in comparison with the method using the standard linear controller introduced in Köhler, Müller, and Allgöwer 2018 which requires thousands of steps to stabilize the system as mentioned in Table 6.2.
In the next section, we show that both the NMPC-t and u-NMPC controllers considered in Scenarios 1 and 3 (using the 10-step prediction horizon) are successfully validated through experiments over a real quadcopter platform.

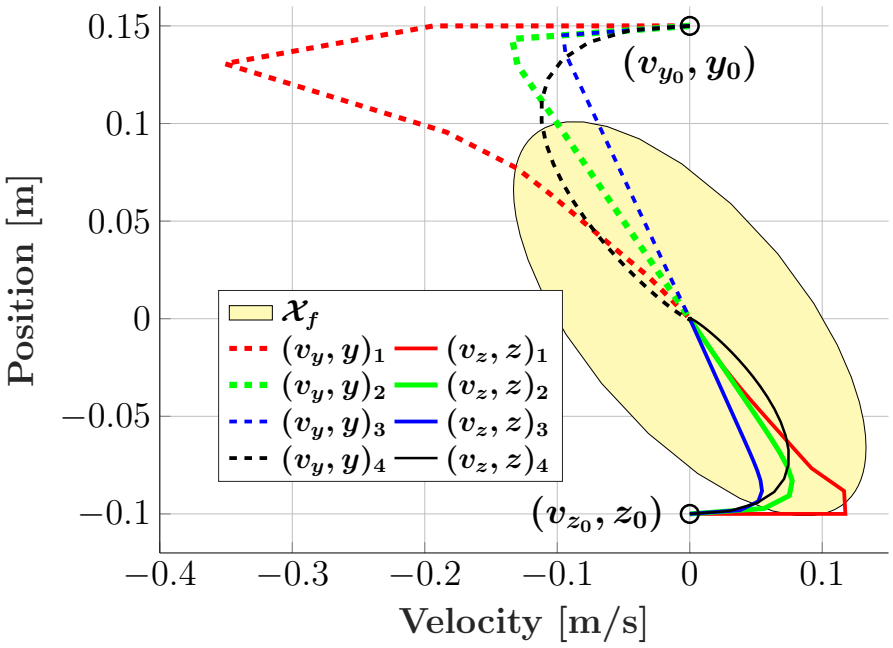


Figure 6.7.: Terminal region $\mathcal{X}_f$ (approximated illustration) and the trajectories $(y, v_y), (z, v_z)$ under different scenarios.

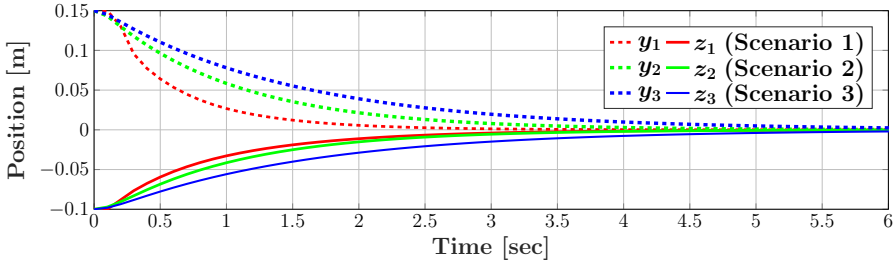


Figure 6.8.: Convergences of x, y, z under different scenarios.

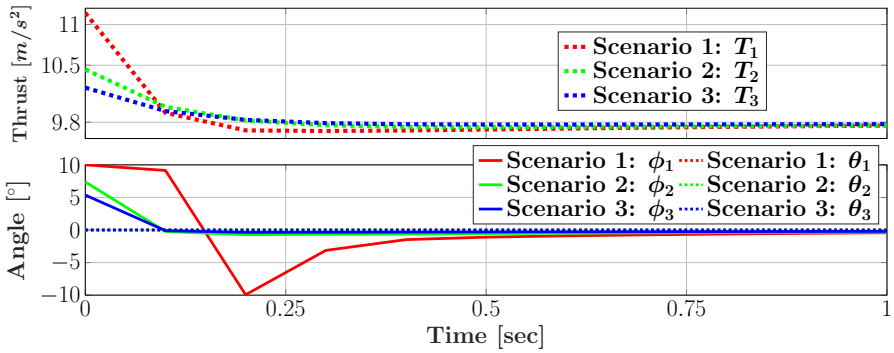


Figure 6.9.: Input values under different scenarios.

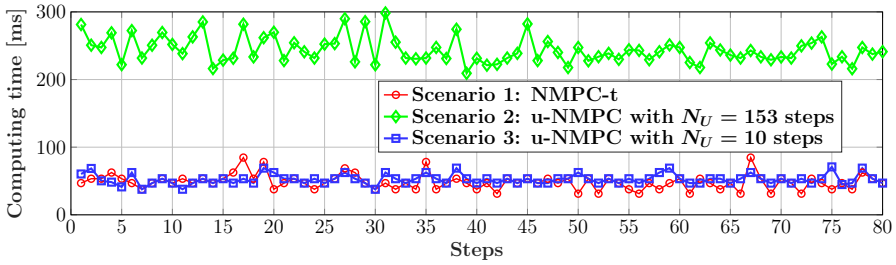


Figure 6.10.: Computation time under simulations.

6.9. Experimental results

This section introduces the experimental validation of the two NMPC controllers, NMPC with and without terminal constraint given in Sections 6.6.1 and 6.6.2 over the laboratory Crazyflie 2.0 (CF) nano-quadcopter. The CF has its inner controller running at 500 Hz to track the set-point of the thrust and the three angles sent from the station computer **NguyenIJC2018**. In the station computer, we implement the NMPC optimization problems (6.24) and (6.32) with the chosen sampling time $\Delta_t = 0.1$ seconds (fast enough to stabilize the system) by using Pyomo (**Pyomo**) and the IPOPT solver **ipopt** in Python 3.0. In the following, we describe the limits of the platform and the mismatches between the theoretically nominal NMPC application and real implementation, then, illustrates the obtained results of the two proposed NMPC controllers.

6.9.1. Experimental validation limits and how to overcome them

Firstly, as introduced in Section 6.4.1, the yaw angle ψ is assumed to be a known constant. However, maintaining a constant direction angle for an aerial vehicle is obviously impossible for long running times. Therefore, we try to stabilize the yaw angle of the CF around zero and update the actual yaw value to the NMPC optimization problem at each sampling time. This alternative approach still guarantees the nominal stability of the closed-loop scheme with a less strict assumption, i.e., requiring the yaw angle value to be constant only during the prediction horizon [8].

Secondly, the execution time is always significant when considering an NMPC controller, especially for controlling the Crazyflie 2.0 quadcopter system with the required sampling time of 0.1 seconds (as illustrated in Figure 6.10 for simulation results). Thus, even with a perfect state feedback at time step k, we cannot obtain the MPC control action immediately at the same time step k as assumed in (6.28) and (6.35). Hence, we have to relax this assumption by introducing the obtained MPC input of time step k to the CF system at time step $k+1$.

Remark 6.7. If the reader is interested in using embedded NMPC with low-power hardware, a discussion on how to re-formulate the NMPC problems (6.24) and (6.32) into their approximated quadratic formulations, and thereafter, solve them by using a modified interior-point solver is detailed in (Zanelli, Horn, Frison, and Diehl 2018). Another discussion on solving a linear MPC problem with a 8-bit microcontroller by using a convex lifting method is presented in Gulan et al. 2017. □

6.9.2. Experimental results

For experiments, we consider the two Scenarios 1 and 3 as given in Section 8.5 with the parameters of the controllers detailed in Tables 6.3 and 6.4 (the u-NMPC controller has its prediction horizon of 10 steps). Figure 6.11 presents the results of x (green lines), y (blue lines) and z (red lines) under two scenarios which shows that both the NMPC-t and u-NMPC controller succeed in stabilizing the CF quadcopter. It can be seen again that the NMPC-t provides faster convergence speed than the u-NMPC controller due to its terminal constraint (1.5 seconds vs. 3 seconds, from blue lines in Figure 6.11). The terminal constraint also results in more input consumption as clearly observed from Figure 6.12. The maximum values of the thrust and the roll angle of the NMPC-t (dashed lines) are significantly higher than the ones from the u-NMPC controller (solid lines). All the inputs respect their constraints as given in (6.22). The computing times under experiment of both NMPC controllers are given in Figure 6.13 which shows that the NMPC-t controller (with $N_T = 5$ steps and the terminal constraint set $\mathcal{X}_f$ as in (6.51)) requires more computation effort than the u-NMPC controller (with $N_U = 10$ steps) but both succeeds in running within the sampling time $\Delta_t = 0.1$ seconds.

[8] The stability is still guaranteed since all the required ingredients detailed in Section 6.5 hold with a general constant yaw angle $\psi \in [-\pi, \pi]$.

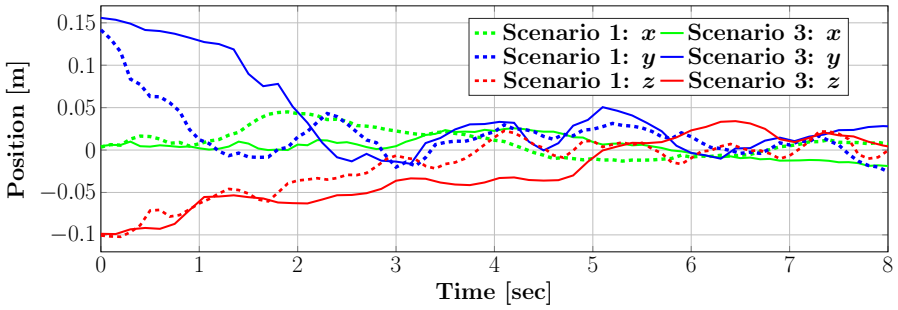


Figure 6.11.: Convergences of x, y, z under different scenarios.

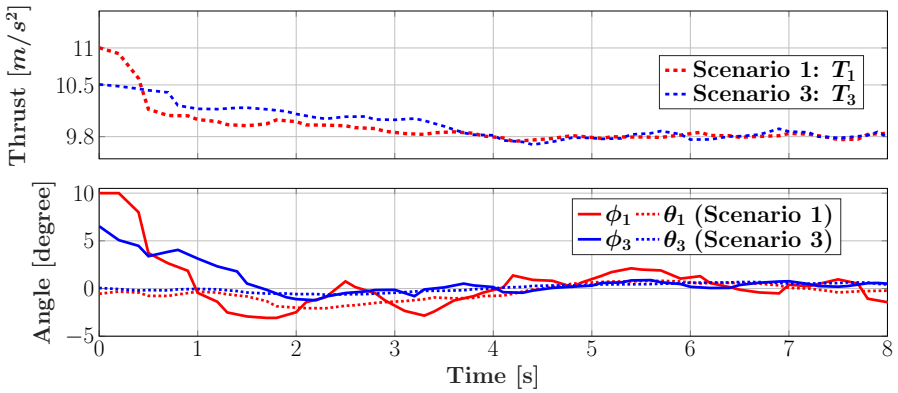


Figure 6.12.: Values of the input under different scenarios.

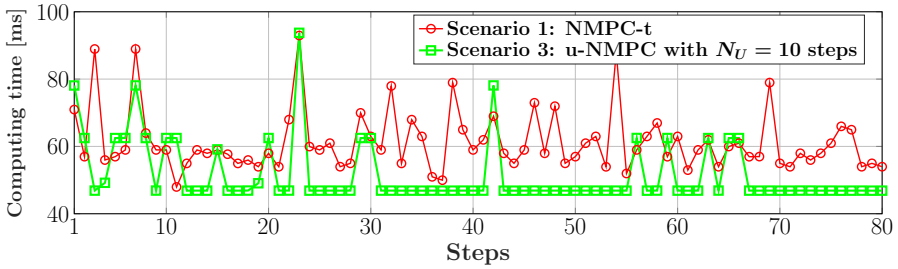


Figure 6.13.: Computation time under experiments using Pyomo in Python 3.0.

6.10. Conclusions and discussions

This chapter gave first an insight on the use of a computed torque control law as a local controller within an NMPC design scheme with stability guarantees **C45**. The ideas were also extended for the attitude control of a multicopter system **C53**. Next, we presented the application of a nonlinear feedback linearization controller for designing two different NMPC schemes, i.e., NMPC with and without terminal constraint, to stabilize the thrust-propelled vehicles dynamics with asymptotic stability guarantee **J13**. By using several standard mathematical inequalities, an ellipsoidal invariant set was constructed under the feedback linerization controller ensuring input constraint satisfaction. Within this invariant set, the feedback law was also bounded by the weighted norm of the corresponding state. These ingredients were gathered to ensure the satisfaction of the stability conditions of the two NMPC designs. The proposed control methods were validated through extensive simulations and experiments over a real quadcopter platform. For this particular work, future development will concentrate on analyzing the mismatches on angle tracking and the delays on sending NMPC input to the controlled system as employed in the experiments.

More general results which can ensure NMPC stability and robustness are worthwhile to follow. For example, there are various recent works related to NMPC/economic MPC without terminal constraints (Alamir and Pannochia 2020; Schwenkel, Köhler, Müller, and Allgöwer 2019; Alamir 2018), robust MPC (Lorenzen, Cannon, and Allgöwer 2019) or data-driven MPC (Berberich, Köhler, Muller, and Allgower 2020; Zanon and Gros 2020) with their references therein.

7. Hierarchical control for DC microgrid systems

This chapter employs hierarchical control strategies for power balancing in DC microgrid systems. We first consider Port-Hamiltonian (PH) modeling as an appropriate method to describe the power-preserving interconnections among the elements of the network. The proposed PH model is inverted using differential flatness to further provide the flat representation of each component of the microgrid (PV, ES, loads, connected to the DC/DC converters). With the inverse dynamics, the states and the control inputs are derived in function of the flat outputs and are used, subsequently, in the formulation of the optimization problem.
The multi-scale dynamics is controlled through a hierarchical architecture divided in three levels. Firstly, at the high level, optimal profiles are generated under a flatness-based optimization control approach for power balancing, cost minimization and decrease of the power dissipation. B-spline parametrization handles continuous-time constraint validation. At the middle level, a model predictive controller is developed to track the optimal profiles obtained at the high level in the presence of bounded disturbances. Finally, at the low level, the tracking profiles are considered as references for the control of the DC/DC converters through switch regulation.
The proposed approach is validated via extensive simulations over a meshed DC microgrid and a multi-source elevator system using real profile data.

The chapter is based on the following papers:

J11 Zafeiratou, I., **I. Prodan**, L. Lefèvre, and L. Piétrac. "Meshed DC microgrid hierarchical control: A differential flatness approach". In: *Electric Power Systems Research*, pp. 106–133, 2020. Elsevier.

J12 Pham, T. H., **I. Prodan**, D. Genon-Catalot, and L. Lefèvre. "Economic constrained optimization for power balancing in a DC microgrid: A multi-source elevator system application". In: *International Journal of Electrical Power & Energy Systems*, pp. 105–153, 2020. Elsevier.

C54 Zafeiratou, I., **I. Prodan**, F. Boem, and L. Lefevre. "Handling power losses in a DC microgrid through constrained optimization", pp. 1–6. In: *Proceedings of the 21th IFAC World Congress*. 2020. Berlin, Germany.

C42 Zafeiratou, I., D. Nguyen, **I. Prodan**, L. Lefevre, and L. Pietrac. "Flatness-Based Hierarchical Control of a Meshed DC Microgrid", pp. 33–40. In: *Proceedings of the 6th IFAC Conference on Nonlinear Model Predictive Control (NMPC'18)*. 2018. Madison, Wisconsin, USA.

C37 Zafeiratou, I., **I. Prodan**, L. Lefèvre, and L. Piétrac. "Dynamical modelling of a DC microgrid using a port-Hamiltonian formalism". In: *IFAC-PapersOnLine, Proceedings of the 9th Vienna International Conference on Mathematical Modelling*, pp. 469–474, 2018.

7.1. Brief literature review

Despite the domination of AC transmission networks, the interest on DC microgrids has increased as a result of the constant development and production of the DC equipment both for energy sources (e.g. solar panels, batteries) and loads (e.g. electrical vehicles, elevators and various smaller DC loads such as computers, LED lights, etc.). To decide the appropriate structure of a microgrid many aspects must be considered, such as its topology (Cortes, Contreras, and Shahidehpour 2017), the distance among the sources and the loads as well as the number of the sources and the type of the converters. A sufficient amount of storing elements is indispensable and depends on many factors such as the sizing of the batteries or their lifetime (Hernández, Sanchez-Sutil, and Muñoz-Rodríguez 2019). Considering all of the above, several modeling and control methods have been developed through the years.

Modeling methodologies: In the literature, different modeling methodologies have been studied to describe such complex dynamical systems. For instance, the Takagi−Sugeno fuzzy modeling approach (Takagi and Sugeno 1985) or the multi−agent paradigm, where different types of agents can be employed in a microgrid system: control agents for controlling the physical units of the system; management agents for managing the microgrid and taking decisions; ancillary agents for performing tasks such as communication and data storage (Mahmoud, Hussain, and Abido 2014). Another classical way to model the microgrid components is by using differential equations (Parisio, Rikos, and Glielmo 2016), which gives an explicit representation of the system dynamics through constitutive equations (for resistors, batteries, capacitors, etc.) and balance equations (Kirchoff's laws).

However, in these approaches the power conservation property and the components' interconnections are not explicitly described. Such an approach is the PH formulation approach which applies for general multiphysical systems (Duindam, Macchelli, Stramigioli, and Bruyninckx 2009), (Schiffer, Fridman, Ortega, and Raisch 2016). It aims at providing an exhaustive, explicit and modular description of the power routing through the network topology, together with the constitutive equations for the resistor-like, capacitor/inductor-like, transformers and sources components. In this structural description, balance equations and energy properties of the systems (passivity, conservation) are satisfied independently by the specific constitutive equations and the numerical values of the model parameters. The overall interconnection structure topology and balance equations are summarized in a geometrical linear structure (namely, a Dirac structure) in the Bond space of effort and flow variables (i.e., the voltage and the current respectively). This Dirac structure allows, for instance, structure preserving discretization (Kotyczka and Lefèvre 2018) or model order reduction (Polyuga and Van der Schaft 2010) which may be needed in the analysis and supervision of grid systems **J11**, **J12**. It may be also used in physically−based control design such as passivity-based control (Ortega, Van Der Schaft, Maschke, and Escobar 2002) or flatness-based approaches such as the one developed in this work.

Control approaches: Various issues, which need to be taken into account when choosing the control approach for a microgrid system, include, e.g., different timescales or islanding and grid−connecting modes. The microgrid energy management problem is generally considered as a constrained optimization problem not straightforward to solve. A multi-level control approach is required (Siad, Malkawi, Damm, Lopes, and Dol 2019) due to the existence of different timescales, the fast and the slow dynamics of the components. Generally, the hierarchical control is divided into three layers, *primary, secondary and tertiary control*, to better deal with real microgrid characteristics.

The primary control (low level) includes a localized supervision of the power distribution and the voltage/current adjustment among the distributed energy resources (DERs) and the converters. The converters contain an internal switching activity that obeys to an external power loop based on a management strategy (B. Wang et al. 2019). One such example is the maximum power point tracking (L. Liu, Meng, and C. Liu 2016), which includes controllers for extracting the maximum available power from solar panels or other renewable energy resources. Another approach is the master/slave control (Mazumder, Tahir, and Acharya 2008), where a device is selected as the master controller over one or more devices that act as slaves. Furthermore, droop control (Bouzid et al. 2019; Vu et al. 2017) allows load balancing during the operation of parallel generators.

The secondary control (middle level) targets the internal processes of the system under voltage and

frequency disturbances. The main purpose is to reduce voltage, frequency or power deviations. These deviations can be controlled locally, but in case of a possible failure the whole system is affected. Therefore, both centralized (Shafiee, Guerrero, and Vasquez 2014) and distributed (Lou, Gu, Y. Xu, Cheng, and W. Liu 2017) secondary controllers have been investigated and various control methods have been proposed. In (Shafiee, Guerrero, and Vasquez 2014), a phase controller is used which regulates the phase angle of the distributed generators instead of the frequency to decrease frequency and amplitude discrepancies. In (P. Wang, X. Lu, X. Yang, W. Wang, and D. Xu 2015), the authors aim to reduce the voltage deviations caused by droop control method in a distributed control framework for accuracy in current sharing. Fuzzy controllers are also investigated, for example in (Xie, Yue, H. Zhang, and Peng 2017) a control structure for the aforementioned Takagi−Sugeno fuzzy systems is developed, mitigating the computational burden. (Baghaee, Mirsalim, and Gharehpetian 2016) improves the performance of a microgrid by controlling the reactive power under disturbances caused by power outages, short circuits and the like. In (Simpson-Porco et al. 2015), frequency and voltage controllers are designed, enabling the possibility to the system to achieve either frequency or voltage regulation while maintaining reactive power sharing. Furthermore, (Ahumada, Cárdenas, Saez, and Guerrero 2015) implements MPC in combination with a PI controller in order to decrease the discrepancies between the nominal and the actual frequency, to increase the stability of the system and to diminish the communication delays.

The tertiary control (high level) deals with the power flow and optimization by taking into account different economical aspects. The power flow is optimally regulated succeeding the load balancing within the transmission network and providing the most economical solution. Several methods have been developed which consider the maximum generation capacity of the energy storage (Mortaz and Valenzuela 2017), the constraints or the uncertainties. The combined use of MPC with MIP for battery scheduling is employed in **J6**. Machine learning algorithms are employed in (Drgoňa, Picard, Kvasnica, and Helsen 2018) for developing approximate MPC laws for household temperature control.

7.2. Contributions

We focus here on DC microgrid systems, e.g.,

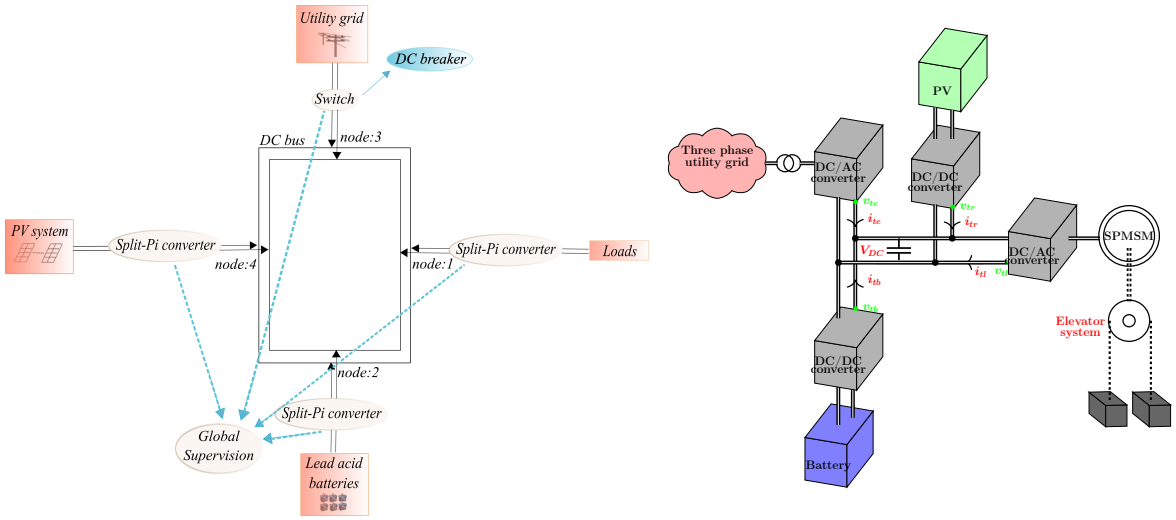
- the meshed DC microgrid system described in Figure 7.1a and
- the multi-source elevator system described in Figure 7.1b,

due to their ability to integrate (through DC/DC or AC/DC converters) different distributed renewable energy resources which generate DC power (e.g., the solar panel) or AC power with varying frequencies (e.g., wind turbine). Nonetheless, DC to AC converters are still necessary due to the fact that some sources and loads cannot be directly connected to the DC bus.

Within the global energy system the fast dynamics correspond to the actuators (e.g., converter, motor), transmission lines and high power energy storage (e.g., supercapacitor) which need to be stabilized around a set-point. The slow dynamics correspond to the energy storage unit (e.g., battery, elevator system, thermal system) and are governed by cost criteria. Thus, at the control design step we need to take into account the different timescales appearing in the system dynamics.

The contributions reside in a coherent combination among Port-Hamiltonian formalism for the DC microgrid modeling, flatness for constrained profile generation and Model Predictive Control for profile tracking under bounded disturbances. More specifically, we employ:

- a PH formalism to ensure power−preserving interconnections among all the DC microgrid's elements. Furthermore, aspects such as energy conservation, the isolation of physical quantities for further use (battery charge, voltages, etc.), dissipation conditions, etc.;
- differential flatness, an advantageous method which allows the off-line study of physical systems in order to predict their behavior. It is considered also as a suitable tool to inverse the system



(a) Meshed DC microgrid architecture.

(b) DC microgrid elevator system.

Figure 7.1.: DC microgrid architectures.

dynamics: the states and inputs are given as combinations of the system's flat outputs. Furthermore, the B-spline properties ensure continuous time constraints validation (the flat output, given as a weighted sum of B-spline basis functions, is fully described by these weights);

- the multi-level control design which produces at the high and middle level optimal profiles to be followed by the lower level. For instance, power balancing requirements at the high level lead to an optimal profile for battery usage, to be tracked at the middle level; the middle level provides voltage and current references for the battery. These profiles are tracked at the low level by an explicit switching law of the DC/DC converters. In all cases we consider bottom to top information for a reliable profile generation (e.g., tracking errors are accounted for);

- the validation of the proposed hierarchical control through extensive simulations based on realistic load, renewable power and electricity price profiles.

7.3. DC microgrid modeling using Port-Hamiltonian formalism

7.3.1. Modeling methodology

In microgrids, a decomposition into subsystems leads to a structure simplification. To achieve this decomposition and to express in detail the power interconnections among the components, a modeling methodology based on PH representation is considered. A schematic view (Bond Graph) for PH representation multiphysical systems is given in Figure 7.2. It is seen as the interconnection of three types of elements: i) energy sources; ii) storing elements (capacitors, inductors); iii) dissipative elements (resistors). These three components are connected through a generalized interconnection structure (Dirac structure(A. v. d. Schaft, Jeltsema, et al. 2014) denoted by D in Figure 7.2) which accounts for the power continuous energy balance equations (e.g. Kirchoff's laws, ideal transformers equations in an electrical network). The general PH state−space representation of a system is (A. v. d. Schaft, Jeltsema, et al. 2014; Duindam, Macchelli, Stramigioli, and Bruyninckx 2009):

$$\begin{aligned} \dot{x}(t) &= [J(t) - R]\partial_x H(x) - Gu(t), \\ y(t) &= G^\top Q x(t) + Du(t), \end{aligned} \tag{7.1}$$

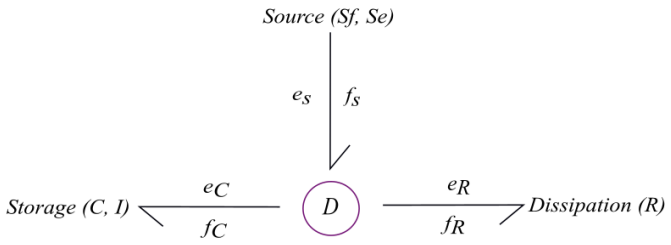


Figure 7.2.: Schematic view for the PH formulation of a multiphysics system, where the arrows describe the power flow direction, considered as the product of associated pairs of efforts e and flows f variables. In the case of electrical circuits, e is the voltage and f is the current.

where $x(t) \in \mathbb{R}^n$ is the state vector of the extensive energy variables $p(t)$ and $q(t)$, which, in the case of electrical circuits, $p(t)$ is the magnetic flux of the inductors and $q(t)$ is the electrical charge of the capacitors. Furthermore, $u(t) \in \mathbb{R}^m$ is the input vector and $y(t) \in \mathbb{R}^m$ is the output vector. In the following, we explain the matrices of the PH representation: i) $J(t) \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{n\times n}$ are skew−symmetric and symmetric positive semi definite matrices which represent the system's interconnection structure power continuity (structural balance equations which typically contain no numerical parameters) and dissipation; ii) $Q \in \mathbb{R}^{n\times n}$ is a positive definite matrix which is, in the linear case, with one−port[1] storage elements considered here, a diagonal matrix with the capacitance values C for the capacitors and the inductance values I for the inductors. More complex nonlinear constitutive equations may also be considered; iii) $D \in \mathbb{R}^{m\times m}$ describes the direct interconnection of the input variables and $G \in \mathbb{R}^{n\times m}$ is the control matrix. [2] , in the linear case with one−port storage elements like it is considered here, a diagonal matrix with the capacitance values for the capacitors and the inductance values for the inductors. More complex non linear constitutive equations may also be considered , capacitance C for the capacitors and inductance I for the inductors, This is a typical PH representation and each of the components has a clear physical justification:

The collocated output $y(t)$ in (7.1) results from the inputs selection and the input map G, in a way that the inner product among input and output vectors minus the dissipative energy gives always the external power supplied to the system (Escobar, J. v. d. Schaft, and Ortega 2015):

i) The Hamiltonian H is the total energy stored in the storage elements within the system, which in the linear case is equal to the following:

$$H(x) = \frac{1}{2}x(t)^\top Qx(t); \tag{7.2}$$

ii) From (7.1) and (7.2) we obtain the external power supplied to the system:

$$\frac{d}{dt}H = u(t)^\top y(t) - x(t)^\top Q^\top RQx(t) - u(t)^\top Du(t). \tag{7.3}$$

The power balance equation (7.3) is equivalent to the Tellegen's theorem, which describes the energy conservation in an electrical network. Note that the Hamiltonian H in an electrical circuit is defined as $H = \frac{1}{2}\frac{q^2}{C} + \frac{1}{2}\frac{p^2}{I}$, where q is the charge of the capacitor and p is the magnetic flux of the inductor.

[1] One-port storage elements are the simplest components of a Bond Graph as they have only two terminals (typical examples are the capacitors and the inductors).

[2] One−port storage elements are the simplest components of a Bond Graoh which have only two terminals, like the capacitors or the inductors.

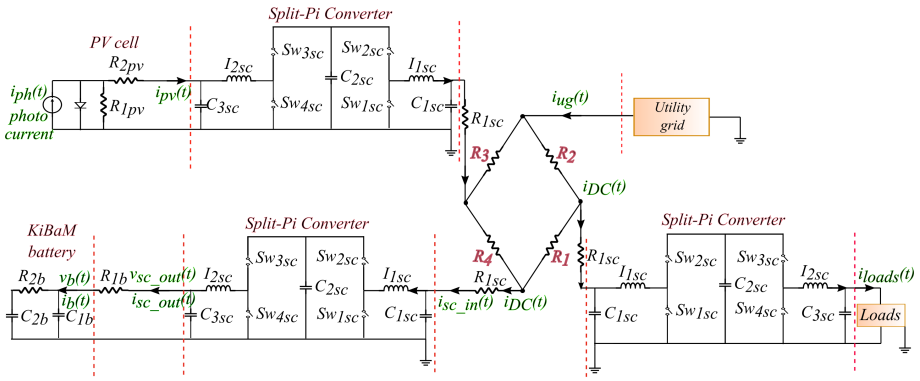


Figure 7.3.: Corresponding electrical circuit of the DC microgrid presented in Figure 7.1a.

7.3.2. Meshed DC microgrid dynamical representation

This section concentrates on presenting the PH model of the meshed DC microgrid architecture in Figure 7.1a). A similar model is also developed for the DC elevator system architecture in Figure 7.1b. Here we will present only the electromechanical elevator's model.

A meshed topology allows the electricity transmission through a variety of sources and transmission lines. Consequently, a possible interruption of the power transmission can be avoided and the safe operation of the system can be ensured. The system is composed by a set of PVs, an ES and an ensemble of loads as in Figure 7.1a. The global system dynamics is separated into different timescales. Primarily, the existence of the DC/DC converters useful for the voltage regulation creates a fast dynamics which needs to be stabilized around a set-point. Secondly, the slow dynamics is related to the battery and the PV system. At the same time, we cope with variable profiles and costs and obey to a set of constraints related to the different characteristics of the system components like the battery's capacity or the permissible UG power.

Figure 7.3 illustrates the corresponding electrical circuit of the DC microgrid (Figure 7.1a): its components and their links. Priority is given to the analysis of the ES since its proper operation is indispensable for the continuous power distribution and cost minimization. For the PV system, profiles are provided generated by the PV model proposed in **C37**, taking into account specific temperature and irradiation data. Additionally, we use specific profiles for the consumer's demand.

Dynamical representation of the Split-Pi/ES system

Figure 7.4 illustrates the lead−acid battery/Split−Pi electrical circuit. According to the Kinetic Battery Model (KiBaM) (J. F. Manwell and J. G. McGowan 1993), the battery consists of two capacitors (with storage capacities q_{1b} and q_{2b}) and a resistor which links them. In the following, the associated PH model is presented:

$$\begin{cases} \dot{x}_{es}(t) &= [J_{es}(d(t)) - R_{es}]Q_{es}x_{es}(t) + G_{es}u_{es}(t), \\ y_{es}(t) &= G_{es}^{\top}Q_{es}x_{es}(t) + D_{es}u_{es}(t), \end{cases} \tag{7.4}$$

where $x_{es}(t) = \begin{bmatrix} p_{1sc}(t) & p_{2sc}(t) & q_{1sc}(t) & q_{2sc}(t) & q_{3sc}(t) & q_{1b}(t) & q_{2b}(t) \end{bmatrix}^{\top} \in R^{7\times 1}$, $u_{es}(t) = \begin{bmatrix} -v_{DC}(t) & -i_{R_{1b}}(t) \end{bmatrix}^{\top} \in R^{2\times 1}$, $y_{es}(t) = \begin{bmatrix} i_{DC}(t) & v_{R_{1b}}(t) \end{bmatrix}^{\top} \in R^{1\times 2}$, where $i_{DC}(t)$ is the current during charging mode. Additionally, the diagonal matrix Q_{es} is equal to diag($1/I_{1sc}$, $1/I_{2sc}$, $1/C_{1sc}$, $1/C_{2sc}$, $1/C_{3sc}$, $1/C_{1b}$, $1/C_{2b}$) $\in R^{7\times 7}$. The skew-symmetric matrix $J_{es}(t) \in R^{7\times 7}$, the

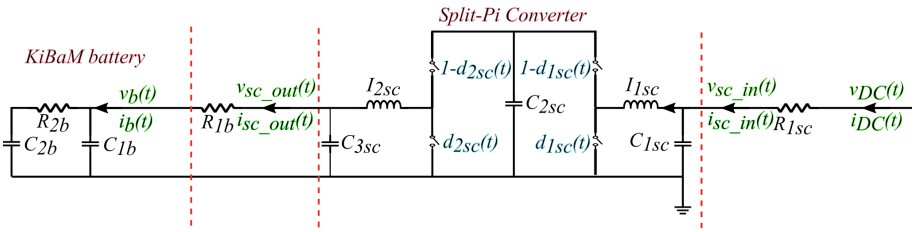


Figure 7.4.: The ES system during charging mode. The notations of the switches (Sw_{1sc}, Sw_{2sc}, Sw_{3sc}, Sw_{4sc}) have been replaced by the corresponding duty cycles d_{1sc} and d_{2sc}.

dissipation matrix $R_{es} \in R^{7\times 7}$, the $G_{es}^{\top} \in R^{2\times 7}$ and $D_{es} \in R^{2\times 2}$ are equal to:

$$J_{es}(t) = \begin{bmatrix} 0 & 0 & 1 & -(1-d_{1sc}(t)) & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-d_{2sc}(t)) & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1-d_{1sc}(t) & -(1-d_{2sc}(t)) & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{7.5}$$

$$R_{es} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{R_{1_{sc}}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{R_{1b}} + \frac{1}{R_{2b}} & -\frac{1}{R_{2b}} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{R_{2b}} & \frac{1}{R_{2b}} \end{bmatrix}, \tag{7.6}$$

$$G_{es}^{\top} = \begin{bmatrix} 0 & 0 & \frac{-1}{R_{1_{sc}}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}, D_{es} = \begin{bmatrix} \frac{1}{R_{1sc}} & 0 \\ 0 & 0 \end{bmatrix} \tag{7.7}$$

where $d_{1sc}(t)$, $d_{2sc}(t)$ are the control variables of our system. All the unknown variables and parameters of the battery model can be found in the literature and depend on the type of the lead−acid battery. The Split−Pi converter switches control the charging and the discharging of the battery.

Dynamical representation of the central transmission network of the meshed DC microgrid

The transmission network model (4-line transmission network, see Figure 7.3) is also represented by using the Bond Graph method **C37**. Its dynamical representation is presented below:

$$P_{ug}(t)+P_{pv}(t) - P_{es}(t) - P_{loads}(t) - P_{R1}(t) - P_{R2}(t) \quad -P_{R3}(t) - P_{R4}(t) = 0, \tag{7.8}$$

where $P_{ug}(t) = i_{ug}(t) \cdot v_{ug}(t)$ and $P_{es}(t) = i_b(t) \cdot v_b(t)$. At first, we simplify and ignore the power losses within the transmission central network:

$$P_{R1}(t) = P_{R2}(t) = P_{R3}(t) = P_{R4}(t) = 0. \tag{7.9}$$

Combining (7.8) and (7.9) the power conservation equation becomes:

$$P_{ug}(t) + P_{pv}(t) - P_{es}(t) - P_{loads}(t) = 0. \tag{7.10}$$

7.3.3. Dynamical model of the multi-source elevator system

The dynamical model of the DC microgrid elevator system illustrated in Figure 7.1b is similarly developed using the PH formalism. The system is equivalently represented by the electrical DC circuit in Figure 7.5 where we denote at the circuit node 1 the common ground. Hereinafter we present briefly each component.

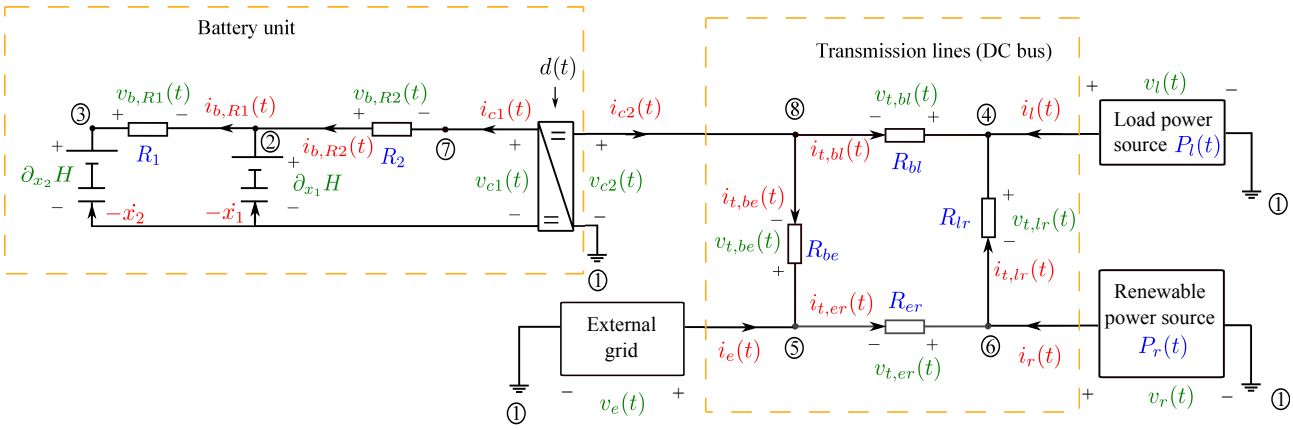


Figure 7.5.: Electrical circuit of the multi-source elevator system.

External grid: As illustrated in Figure 7.5 the DC microgrid is connected to the AC external grid which is modeled here as a controllable current source $i_e(t) \in \mathbb{R}$ with the following physical limitations:

$$i_{e,min} \leq i_e(t) \leq i_{e,max}, \tag{7.11}$$

with the upper and lower bounds $i_{e,max}, i_{e,min} \in \mathbb{R}$.

Load unit: The load component of the DC microgrid represents a combination of the electromechanical elevator and an AC/DC converter. Here, we simply model this as a power source $P_l(t) \in \mathbb{R}$ under current, $i_l(t) \in \mathbb{R}$, and voltage, $v_l(t) \in \mathbb{R}$, constraint:

$$i_l(t)v_l(t) = -P_l(t). \tag{7.12}$$

Renewable source: The DC microgrid system contains a solar panel and the corresponding DC/DC converter. Similarly, we model the distributed energy resource as a power source $P_r(t) \in \mathbb{R}$ satisfying the following relation:

$$i_r(t)v_r(t) = P_r(t), \tag{7.13}$$

with $i_r(t), v_r(t) \in \mathbb{R}$ the renewable source current and voltage as illustrated in Figure 7.5.

Electrical storage unit: We consider here a lead-acid type of battery which is modelled as two electronic "wells", a bridge connecting them described by R_1 and an internal resistor R_2 (see also Figure 7.5) (J. Manwell and J. McGowan 1993), (Lifshitz and G. Weiss 2015). Therefore, hereinafter, we denote by $\mathbf{x}(t) \in \mathbb{R}^2$ the battery charges. The Hamiltonian representing the stored energy in the battery is given by:

$$H(\mathbf{x}) = \mathbf{x}(t)^T \mathbf{Q}_1 + \frac{1}{2}\mathbf{x}(t)^T \mathbf{Q}_2 \mathbf{x}(t), \tag{7.14}$$

with $\mathbf{Q}_1 = E\mathbf{1}_2 \in \mathbb{R}^2$, $\mathbf{Q}_2 = diag\{C_1, C_2\} \in \mathbb{R}^{2\times 2}$. E is the battery internal voltage; C_1 and C_2 are the battery internal capacitances.

Via Ohm's law, we derive the current and voltage relation for the battery resistors:

$$\mathbf{R}_b \mathbf{i}_{bR}(t) + \mathbf{v}_{bR}(t) = \mathbf{0}, \tag{7.15}$$

with

$$\mathbf{i}_{bR}(t) = \begin{bmatrix} i_{b,R1}(t) \\ i_{b,R2}(t) \end{bmatrix}, \ \mathbf{v}_{bR}(t) = \begin{bmatrix} v_{b,R1}(t) \\ v_{b,R2}(t) \end{bmatrix} \in \mathbb{R}^2, \tag{7.16}$$

and $\mathbf{R}_b = diag\{R_1, R_2\} \in \mathbb{R}^{2\times 2}$ a positive diagonal matrix (see also Fig. 7.5).

The DC/DC converter: As illustrated in Figure 7.5, the battery has an associated DC/DC converter which is assumed to be an ideal transformer described by the following relations:

$$\begin{cases} d(t) i_{c1}(t) &= -i_{c2}(t), \\ v_{c1}(t) &= d(t) v_{c2}(t), \end{cases} \tag{7.17}$$

where $d(t) \in \mathbb{R}$ represents the positive duty cycle:

$$d(t) > 0. \tag{7.18}$$

While here we assume no loss at charge/discharge in the DC/DC converter for simplicity, non zero charge/discharge losses can be taken into account by the resistor series connections to the two sides of the converter in Figure 7.5 with appropriate values. These additional resistors are, then, fused with the resistors of the battery or the DC bus. Consequently, with suitable resistor values of the battery and/or the DC bus, the electrical circuit of the DC microgrid in Figure 7.5 is still valid.

Transmission lines and resistor network: The DC bus, i.e., the transmission lines, are illustrated in the electrical circuit of Figure 7.5 where its model is reduced to a resistor network. Using Ohm's law, we derive the current and voltage relation for the resistor network respect:

$$\mathbf{R}_t \mathbf{i}_{tR}(t) + \mathbf{v}_{tR}(t) = \mathbf{0}, \tag{7.19}$$

with

$$\begin{array}{ll} \mathbf{i}_{tR}(t) & = [i_{t,bl}(t) \ \ i_{t,be}(t) \ \ i_{t,er}(t) \ \ i_{b,rl}(t)]^T \quad \in \mathbb{R}^4, \\ \mathbf{v}_{tR}(t) & = [v_{t,bl}(t) \ \ v_{t,be}(t) \ \ v_{t,er}(t) \ \ v_{t,rl}(t)]^T \quad \in \mathbb{R}^4, \end{array} \tag{7.20}$$

and $\mathbf{R}_t = diag\{R_{bl}, R_{be}, R_{er}, R_{lr}\} \in \mathbb{R}^{4\times 4}$ a positive diagonal matrix (see also Fig. 7.5). Using the definition of the Dirac structure we provide the interconnections of the DC microgrid network through a closed graph. For more details we refer to **J12**.

Next, we go back to the meshed DC microgrid model and we provide its flat representation. Such a representation can also be found for the multi-source elevator system DC microgrid architecture.

7.3.4. Flat representation of the ES connected to the Split-Pi converter

The ES state−space representation model (7.4), (7.5), (7.6), (7.7), (7.10) is rewritten in function of the flat outputs of the system using the differential flatness notion. Differential flatness allows to describe the system's states and inputs as algebraic combinations of the flat outputs and a finite number of their derivatives. In turn, the flat output is an algebraic combination of states and input derivatives (Fliess, Lévine, P. Martin, and Rouchon 1995). A nonlinear system (Fliess, Lévine, P. Martin, and Rouchon 1995):

$$\dot{x} = f(x, u), \tag{7.21}$$

with $x \in R^n$ the state vector and $u \in R^m$ the input vector , for which $f(0,0) = 0$ and $rank \dfrac{\partial f}{\partial u} = m$ are verified, can be characterized as differentially flat, if there exists a flat output vector $z = [z_1 \ z_2 \ ... \ z_m]^\top$ which satisfies the following conditions: i) the flat output z is presented in function of the states and the inputs of the system and their derivatives, $z = \Phi(x, u, \dot{u}, \ddot{u}, ...)$; ii) the states and the inputs of the

system (7.4), (7.5), (7.6), (7.7) are described in terms of the flat outputs and a finite number of their derivatives, $x = \Phi_x(z, \dot{z}, \ddot{z}, ...)$ and $u = \Phi_u(z, \dot{z}, \ddot{z}, ...)$; iii) the flat outputs z and their derivatives are differentially independent.

Flatness and controllability are two associated properties. In (Fliess, Lévine, P. Martin, and Rouchon 1995) it is demonstrated that a nonlinear system is flat if and only if it is controllable. Finding the flat outputs set for nonlinear systems is convoluted. The literature provides useful approaches, such as the algorithm proposed by (Franke and Robenack 2013), which is a methodical computation of flat outputs for nonlinear control systems. The algorithm uses symbolic linearization of the system to generate flat outputs. A subsequent calculation of matrices, nullspaces and inverses leads to a set of the corresponding flat outputs through integration. The algorithm simultaneously verifies if the system is not controllable and, consequently, not flat (Levine 2009).

The dynamical PH model presented in (7.4)), (7.5), (7.6) and (7.7) has two inputs ($v_{DC}(t)$, $i_{R_{1b}}(t)$). The duty cycles ($d_{1sc}(t)$, $d_{2sc}(t)$), as control variables, are considered also as inputs to the system. According to (Fliess, Lévine, P. Martin, and Rouchon 1995), the number of flat outputs is equal to the number of inputs, which means that four flat outputs need to be found. Therefore, the four flat outputs provided by the algorithm are the following:

$$z_1(t) = \frac{1}{I_{1sc}}\frac{p_{1sc}(t)^2}{2} + \frac{1}{I_{2sc}}\frac{p_{2sc}(t)^2}{2} + \frac{1}{C_{2sc}}\frac{q_{2sc}(t)^2}{2}, \tag{7.22a}$$

$$z_2(t) = q_{3sc}(t) + q_{1b}(t), \tag{7.22b}$$

$$z_3(t) = q_{2b}(t), \tag{7.22c}$$

$$z_4(t) = q_{2sc}(t). \tag{7.22d}$$

The flat outputs are in function of the states $p_{1sc}(t)$, $p_{2sc}(t)$, $q_{3sc}(t)$, $q_{1b}(t)$, $q_{2b}(t)$. Substituting (7.22a)−(7.22d) into the PH model (7.4)-(7.7), the remaining states and inputs are written in function of the flat outputs. Appendix A presents the detailed flat representation of the system. Below, the general flat representation of the system is written where the states and the inputs are described in function of the flat outputs (7.22a)-(7.22d) and their derivatives[3]:

$$p_{1sc}(t) = \Phi_1(z_1, \dot{z}_2, z_3, \dot{z}_3, z_4), \tag{7.23a}$$

$$p_{2sc}(t) = \Phi_2(\dot{z}_2, z_3, \dot{z}_3), \tag{7.23b}$$

$$q_{1sc}(t) = \Phi_3(z_1, \dot{z}_1, z_2, \dot{z}_2, \ddot{z}_2, z_3, \dot{z}_3, \ddot{z}_3, z_4, \dot{z}_4), \tag{7.23c}$$

$$q_{2sc}(t) = \Phi_4(z_4), \tag{7.23d}$$

$$q_{3sc}(t) = \Phi_5(z_2, z_3, \dot{z}_3), \tag{7.23e}$$

$$q_{1b}(t) = \Phi_6(z_3, \dot{z}_3), \tag{7.23f}$$

$$q_{2b}(t) = \Phi_7(z_3), \tag{7.23g}$$

$$d_{1sc}(t) = \Phi_8(z_1, z_2, \dot{z}_2, \ddot{z}_2, z_3, \dot{z}_3, \ddot{z}_3, z_4), \tag{7.23h}$$

$$d_{2sc}(t) = \Phi_9(z_2, \ddot{z}_2, z_3, \dot{z}_3, \ddot{z}_3, z_4), \tag{7.23i}$$

$$v_{DC}(t) = \Phi_{10}(z_1, \dot{z}_1, \ddot{z}_1, z_2, \dot{z}_2, \ddot{z}_2, \dddot{z}_2, z_3, \dot{z}_3, \ddot{z}_3, \dddot{z}_3, z_4, \dot{z}_4, \ddot{z}_4), \tag{7.23j}$$

$$i_{R_{1b}}(t) = \Phi_{11}(\dot{z}_3, \ddot{z}_3). \tag{7.23k}$$

Next, the B−splines curves are employed, an appropriate tool for flat output parametrization due to its properties of convexity, smoothness and differentiability used for continuous−time constraints validation. The B−splines degree depends on the highest order derivative where the continuity needs

[3]Wherever it is straightforward implied by the text, we discard the time dependence.

to be ensured. Hence, in the following the flat output $z(t)$ is projected over N B−splines of order d:

$$z(t) = \sum_{i=1}^{N} p_i \cdot B_{i,d}(t) = \mathcal{P}\mathcal{B}_d(t), \tag{7.24}$$

where $p_i \in \mathbb{R}^3$ is gathered into the matrix $\mathcal{P} \in \mathbb{R}^{3\times N}$ of N control points, $\mathcal{P} = \begin{bmatrix} p_1 & p_2 & ... & p_N \end{bmatrix}$. In (7.24) $\mathcal{B}_d(t) = \begin{bmatrix} B_{1,d}(t) & B_{2,d}(t) & ... & B_{N,d}(t) \end{bmatrix}^\top$ is the B−spline vector. Furthermore, the B−splines are defined over a knot−vector $T = \{\tau_0 \leq \tau_1 \leq ... \leq \tau_m\} \in \mathbb{R}^{N+d}$ which is a set of non−decreasing time instants with $m = N + d + 1$. More details on the theory and the properties of B−splines, employed in the following section, can be found in **C30** and in Chapter 3.

7.4. Hierarchical constrained optimization-based control

This section analyzes the hierarchical control problem. The main goal is to reduce the electricity cost by minimizing the energy consumption of the UG, hence taking advantage of the PV power production and the ES system capacity. The control variables of the ES system (7.4), (7.5), (7.6), (7.7) are the duty cycles of the Split−Pi converter (7.5), $d_{1sc}(t)$, $d_{2sc}(t)$, and the power generated from the UG, $P_{ug}(t)$. Considering all the elements introduced in the previous section, the hierarchical control approach is presented in Figure 7.6:

i) *high level* (power flow optimization): optimal profiles for the battery current i_b and voltage v_b are generated by an optimization problem with continuous-time constraint validation ensured by the B−splines properties;

ii) *middle level* (battery scheduling): a tube-MPC tracking mechanism (Langson, Chryssochoos, Raković, and Mayne 2004) is employed which provides an efficient battery scheduling under current and voltage additive noises;

iii) *low level* (switching activity in the converter): an explicit control law for the duty cycles of the converter is provided for tracking the a priori given battery current and voltage profiles.

7.4.1. High level control

Hereinafter, the high level considers the minimization of the power generated by the UG (with $P_{ug}(t) = -P_{pv}(t) + P_{es}(t) + P_{loads}(t)$ as in (7.10)) while satisfying the ES system dynamics ((7.4), (7.5), (7.6), (7.7)) and the constraints:

$$\min_{i_b(t),v_b(t)} \int_{t_0}^{t_f} e(t)(\underbrace{P_{es}(t)}_{i_b(t)v_b(t)} + P_{loads}(t) - P_{pv}(t))dt, \tag{7.25a}$$

$$subject\ to: \ the\ system\ dynamics\ (7.23a) - (7.23k), \tag{7.25b}$$

$$v_b^{min,h} \leq v_b(t) \leq v_b^{max,h}, \tag{7.25c}$$

$$i_b^{min,h} \leq i_b(t) \leq i_b^{max,h}, \tag{7.25d}$$

$$q_{2b}^{min,h} \leq q_{2b}(t) \leq q_{2b}^{max,h}, \tag{7.25e}$$

$$P_{ug}^{min,h} - P_{loads}(t) + P_{pv}(t) \leq P_{es}(t) \leq P_{ug}^{max,h} + P_{loads}(t) - P_{pv}(t), \tag{7.25f}$$

with the control variables being the battery's voltage, v_b, and the battery's current, i_b. Replacing in (7.25a)−(7.25f) the ES power in terms of the control variables, i_b, v_b, we obtain a nonlinear optimization problem. Furthermore, considering $q_{1b}(t)$ and $i_{R_{1b}}(t)$ from (7.4) (which verify $i_b(t) = i_{R_{1b}}(t)$ and $v_b(t) = \dfrac{q_{1b}(t)}{C_{1b}}$) in function of the flat outputs as in (7.23f) and (7.23k), the voltage and current

profile references are expressed as:

$$v_b(t) = \frac{1}{C_{2b}} z_3(t) + R_{2b}\dot{z}_3(t), \tag{7.26a}$$

$$i_b(t) = (1 + \frac{C_{1b}}{C_{2b}})\dot{z}_3(t) + C_{1b}R_{2b}\ddot{z}_3(t). \tag{7.26b}$$

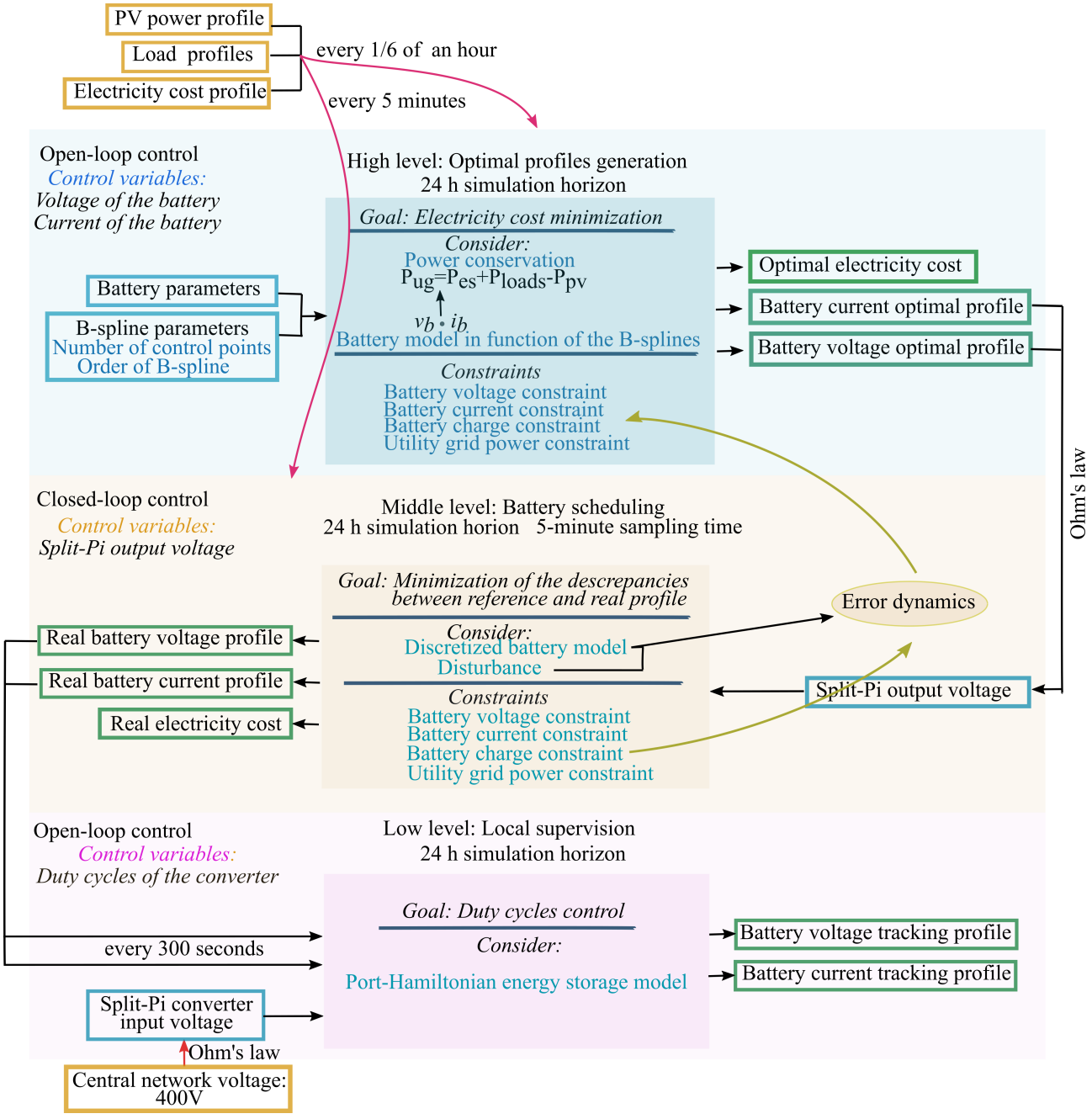


Figure 7.6.: Flowchart of the hierarchical control for the DC microgrid.

Next in (7.26a) and (7.26b), we continue with the B−spline parametrization of $z_3(t)$:

$$v_b(t) = \sum_{i=1}^{N} \left(\frac{1}{C_{2b}} \mathcal{P} + R_{2b} \mathcal{P} \right) \mathcal{B}_d^{(r)}(t), \tag{7.27a}$$

$$i_b(t) = \sum_{i=1}^{N} \left[\left(1 + \frac{C_{1b}}{C_{2b}} \right) \mathcal{P} + C_{1b} R_{2b} \mathcal{P} \right] \mathcal{B}_d^{(r)}(t). \tag{7.27b}$$

In (7.27a) and (7.27b) the differentiation property of the B−splines is employed. The $r-$order derivatives of $d-$order B−splines can be expressed as $d-r$ order B−splines which, in turn, can be expressed as $d-$order B−splines over each knot sub−interval:

$$\mathcal{B}_d^{(r)}(t) = M_{d,d-r} \mathcal{B}_{d-r}(t) = M_{d,d-r} S_{\kappa,d-r,d} \mathcal{B}_d(t), \ \ \forall t \in [\tau_\kappa, \tau_{\kappa+1}). \tag{7.28}$$

From (7.24), (7.26b), (7.26a) and (7.28), the battery's output current and voltage are derived in function of the B−splines:

$$v_b(t) = \sum_{i=1}^{N} \left[\frac{1}{C_{2b}} p_i + R_{2b} \left(\mathcal{P} M_{d,d-1} S_{\kappa,d-1,d} \right)_i \right] \mathcal{B}_{i,d}(t), \tag{7.29a}$$

$$\begin{aligned} i_b(t) = \sum_{i=1}^{N} & \left[\left(1 + \frac{C_{1b}}{C_{2b}} \right) \left(\mathcal{P} M_{d,d-1} S_{\kappa,d-1,d} \right)_i + C_{1b} R_{2b} \cdot \left(\mathcal{P} M_{d,d-2} S_{\kappa,d-2,d} \right)_i \right] \cdot \\ & \cdot \mathcal{B}_{i,d}(t), \ \ \forall t \in [\tau_\kappa, \tau_{\kappa+1}) \, . \end{aligned} \tag{7.29b}$$

The representation obtained in (7.29a) and (7.29b) is introduced in the cost function and the constraints of the optimization problem (7.25a)−(7.25f). Hence, it is rewritten in function of the B−splines and a finite number of control variables represented by the control points. The reference profiles obtained at the high level for the battery current and voltage will be denoted in the middle level as i_b^{ref} and v_b^{ref}, respectively.

7.4.2. Middle level control

In this section, a tube−MPC controller is developed to track the output voltage reference profile, $v_{sc\_out}^{ref}$, of the Split−Pi converter under bounded noise. Note that the output voltage reference of the converter can be written in function of the battery current and voltage reference profiles obtained at the high level from (7.25a)−(7.25f) following:

$$v_{sc\_out}^{ref}(t) = v_b^{ref}(t) + i_b^{ref}(t) R_{1b}. \tag{7.30}$$

Using the Euler explicit method, the battery dynamics is discretized with the charges of the battery, q_{1b} and q_{2b}, as state variables, the output voltage from the Split−Pi converter, v_{sc_out}, as input variable and the current and voltage of the battery, v_b and i_b, as output variables, considered as $\tilde{x}(k) = \begin{bmatrix} \tilde{q}_{1b}(k) & \tilde{q}_{2b}(k) \end{bmatrix}^\top$, $\tilde{u}(k) = \tilde{v}_{sc\_out}(k)$ and $\tilde{y}(k) = \begin{bmatrix} \tilde{i}_b(k) & \tilde{v}_b(k) \end{bmatrix}^\top$, where $\tilde{i}_b(k) = \tilde{i}_{sc}(k)$ and $\tilde{v}_b(k) = \dfrac{\tilde{q}_{1b}(k)}{C_{1b}}$ (see the ES circuit in Figure 7.4). Therefore, the obtained discretized system is presented as follows:

$$\begin{cases} \tilde{x}(k+1) & = A\tilde{x}(k) + B\tilde{u}(k), \\ \tilde{y}(k) & = C\tilde{x}(k) + D\tilde{u}(k), \end{cases} \tag{7.31}$$

with $A = \begin{bmatrix} 1 - \frac{T_s}{C_{1b}}\left(\frac{1}{R_{1b}} + \frac{1}{R_{2b}}\right) & \frac{T_s}{C_{2b}R_{2b}} \\ \frac{T_s}{C_{1b}R_{2b}} & 1 - \frac{T_s}{C_{2b}R_{2b}} \end{bmatrix}$, $B = \begin{bmatrix} \frac{T_s}{R_{1b}} \end{bmatrix}$, $C = \begin{bmatrix} -\frac{1}{C_{1b}R_{1b}} & 0 \\ \frac{1}{C_{1b}} & 0 \end{bmatrix}$, $D = \begin{bmatrix} \frac{1}{R_{1b}} \\ 0 \end{bmatrix}$ and T_s the sampling time.

Next, a tracking MPC formulation is proposed in which the cost penalizes the tracking error (the difference between actual and reference output profiles) over a finite prediction horizon N_p:

$$\min_{\tilde{u}(k)} \sum_{i=k}^{k+N_p-1} (\tilde{y}(i) - \tilde{y}^{ref}(i))^\top Q_{\tilde{y}} (\tilde{y}(i) - \tilde{y}^{ref}(i)) + \\ + (\tilde{u}(i) - \tilde{u}^{ref}(i))^\top R_{\tilde{u}} \left(\tilde{u}(i) - \tilde{u}^{ref}(i)\right) \tag{7.32a}$$

$$subject\ to: \ the\ system\ dynamics\ ((7.31)), \tag{7.32b}$$

$$\tilde{v}_b^{min,m} \leq \tilde{v}_b(k) \leq \tilde{v}_b^{max,m}, \tag{7.32c}$$

$$\tilde{i}_b^{min,m} \leq \tilde{i}_b(k) \leq \tilde{i}_b^{max,m}, \tag{7.32d}$$

$$\tilde{q}_{2b}^{min,m} \leq \tilde{q}_{2b}(k) \leq \tilde{q}_{2b}^{max,m}, \tag{7.32e}$$

$$\tilde{P}_{ug}^{min,m} \leq \tilde{P}_{ug}(k) \leq \tilde{P}_{ug}^{max,m}, \tag{7.32f}$$

with $\tilde{y}^{ref}(k) = \begin{bmatrix} \tilde{i}_b^{ref}(k) & \tilde{v}_b^{ref}(k) \end{bmatrix}^\top$, the current and voltage references of the battery, and $\tilde{u}^{ref}(k) = \tilde{v}_{sc\_out}^{ref}(k)$, the output voltage reference of the Split−Pi converter, taken at T_s sampling time. The control variable of the tracking problem is the output voltage of the Split-Pi converter, v_{sc_out}. The last constraint $P_{ug}(t)$ is rewritten below:

$$\tilde{P}_{ug}^{min,m} - \tilde{P}_{loads}(k) + \tilde{P}_{pv}(k) \leq \tilde{P}_{es}(k) \leq \tilde{P}_{ug}^{max,m} - \tilde{P}_{loads}(k) + \tilde{P}_{pv}(k), \tag{7.33}$$

where $\tilde{P}_{es}(k) = \tilde{i}_b(k)\tilde{v}_b(k)$ as aforementioned. The optimization problem has nonlinear constraints and a variable electricity cost, considers PV, load demands and electricity price profiles and assumes the existence of perturbations.

At the high level, a desired profile is generated for the voltage, v_b, and the current, i_b, of the battery. The dynamics considered at the middle level has to follow these profiles (replaced by the output voltage of the Split−Pi converter v_{sc_out}) as best as possible. The considered approach is the so−called *tube−MPC* (Langson, Chryssochoos, Raković, and Mayne 2004) where an MPC law provides the nominal input (based on the nominal, noise−free dynamics) and the actual input adds to the nominal value a corrective term which counteracts the noise.

The tracking error, under certain assumptions, can be bounded by an RPI set. Since the profile to be tracked is generated at the high level we can tighten the constraints considered in its design so as to guarantee reliability under noises (with the tightening factor being defined by the aforementioned RPI set). The pair of nominal input and nominal state $(\tilde{u}(k), \tilde{x}(k)$ are considered generated by the repeated application of a MPC law over the nominal dynamics of the battery (7.34b). Furthermore, the real dynamics (7.34a) is affected by the bounded noise $\tilde{w}(k)$:

$$\tilde{x}_w(k+1) = A\tilde{x}_w(k) + B\tilde{u}_w(k) + \tilde{w}(k), \tag{7.34a}$$

$$\tilde{x}(k+1) = A\tilde{x}(k) + B\tilde{u}(k), \tag{7.34b}$$

where $\tilde{w}(k)$ is the perturbation, $\tilde{x}(k)$ is the nominal state and $\tilde{x}_w(k)$ is the real, noise−affected, state. Linking the nominal and actual inputs through the relation $\tilde{u}_w(k) = \tilde{u}(k) + K(\tilde{x}(k)\text{-}\tilde{x}_w(k))$ allows us to write the tracking error dynamics as:

$$s(k+1) = (A + BK)s(k) + \tilde{w}(k). \tag{7.35}$$

For any controllable pair $(A,\ B)$ in (7.35), there exists a static feedback K such that $(A+BK)$ is stable which means that there exists an RPI set $\mathcal{S}$ for which $s(k) \in S\ \forall\ k \geq k_0$ holds. Such a set can be computed with the ultimate bounds method (Kofman, Haimovich, and M. M. Seron 2007) or iterative procedures (Olaru, J. A. De Doná, M. M. Seron, and Stoican 2010).

Having $s(k) \in S$ is equivalent with $\tilde{x}_w(k) \in \{\tilde{x}(k) \oplus S\}$ (note that $\oplus$ is the Minkowski sum). In other words, the nominal $\tilde{x}(k)$ has to be chosen more conservatively than $\tilde{x}_w(k)$. Thus, to ensure that $\tilde{x}_w(k) \in \tilde{X} = \{x^{min,m} \leq x \leq x^{max,m}\}$, $\tilde{x}(k)$ is limited as follows:

$$\tilde{x}(k) \in \tilde{X} \ominus S \tag{7.36}$$

Note that the restriction on $\tilde{x}(k)$ translates to a similar restriction on $\tilde{y}(k)$:

$$\tilde{y}(k) \in \tilde{Y} \ominus \mathcal{C}S, \tag{7.37}$$

where $\tilde{Y}$ is a shorthand notation for constraints (7.32c), (7.32d) and (7.32f). The $\ominus$ symbol refers to the Pontryagin difference. The same tightening term $\mathcal{C}S$ is considered in the profile generation implemented at the high level as in (7.29a)−(7.29b).

7.4.3. Low level control

The low level control focuses on the fast dynamics of the system caused by the switching activity within the Split−Pi converter. The tracking profiles obtained in the middle level are taken into account as references for the duty cycles supervision. In order to proceed to the analysis of the Split−Pi converter system, we follow the patent of United States Patent and Trademark Office No: US 6914420 B2 published on July 2005 (Crocker 2005). The patent provides the relations among the input/output voltage (v_{sc_in}, v_{sc_out}) and the duty cycles (d_{1sc}, d_{2sc}) of the converter. Notice that the output voltage v_{sc_out} is always between 12-13 V, according to the battery's parameters considered later in the simulations, and the input voltage v_{sc_in} is always $\approx 400\ V$. Therefore, the Split−Pi converter always operates in down−conversion during battery's charging (positive direction) and up−conversion during battery's discharging (negative direction). Consequently, only the duty cycle d_{2sc} takes values from 0 to 1 (on/off switching between Sw_3 and Sw_4) and the duty cycle d_{1sc} is always equal to 0 (Sw_2 is always off and Sw_1 is always on). Thus, the control variable is d_{2sc} and:

$$\frac{v_{sc\_out}(t)}{v_{sc\_in}(t)} = 1 - d_{2sc}(t) \tag{7.38}$$

where, from Ohm's law,

$$v_{sc\_in}(t) = v_{DC}(t) - i_{DC}(t)R_{1sc}, \tag{7.39}$$
$$v_{sc\_out}(t) = v_b(t) + i_b(t)R_{1b}. \tag{7.40}$$

Moreover, since there is no dissipation within the converter, at the equilibrium point, where $\dot{p}_{isc} = 0$ and $\dot{q}_{isc} = 0$, the total energy contained in the capacitors and inductors is preserved, thus:

$$u_{sc\_in}(t)i_{sc\_in}(t) = u_{sc\_out}(t)i_{sc\_out}(t), \tag{7.41}$$
$$i_{DC}(t)v_{sc\_in}(t) = i_b(t)v_{sc\_out}(t). \tag{7.42}$$

Substituting equations (7.39), (7.40), (7.42) in (7.38) concludes in the relation below:

$$d_{2sc}(t) = 1 - \frac{v_{DC}(t) - \sqrt{v_{DC}^2(t) - 4(v_{sc\_out}(t) - v_b(t))(v_{sc\_out})}}{2(v_{sc\_out}(t) - v_b(t))}. \tag{7.43}$$

The duty cycle d_{2sc} is the control variable of the low level. The last equation (7.43) is valid only if $v_{sc\_out}(t) \neq v_b(t)$ as in (7.30) and (7.26a).

7.5. Simulation results

This section presents briefly some simulation results using real numerical data, hence validating the proposed approaches applied for the two DC microgrid arhitectures in Figure 7.1a and Figure 7.1b.

7.5.1. Hierarhical control for the meshed DC microgrid

In Table 7.1 and Table 7.2, the parameters of the DC microgrid and the simulation settings are indicated for the the high and the middle levels.

Table 7.1.: Model parameters for the simulations

Variable	Values	Units
R_{1sc}, R_{1b}, R_{2b}	0.1, 0.025, 0.088	$[\Omega]$
I_{1sc}, I_{2sc}	0.25, 0.25	$[H]$
C_{1sc}, C_{2sc}, C_{3sc}	0.0008, 0.0008, 0.0008	$[F]$
C_{1b}, C_{2b}	86400, 21600	$[F]$

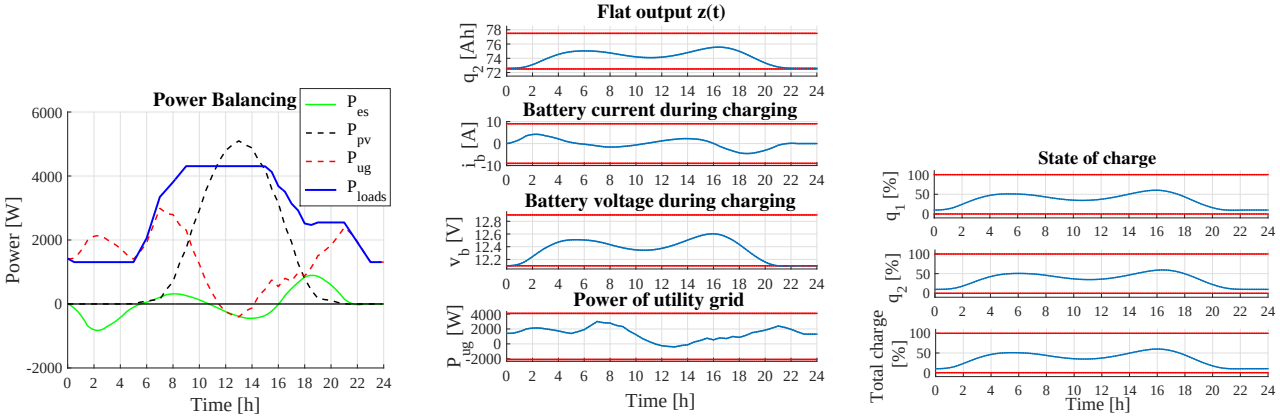
Table 7.2.: System settings for the simulations

	Variable	Values	Units
High level	N as in (7.29a),(7.29b)	18	
	d as in (7.29a),(7.29b)	4	
Constraints	$v_b^{min,h} \leq v_b(t) \leq v_b^{max,h}$	$12.1 \leq v_b(t) \leq 12.9$	$[V]$
	$i_b^{min,h} \leq i_b \leq i_b^{max,h}$	$-9 \leq i_b(t) \leq 9$	$[A]$
	$q_{1b}^{min,m} \leq q_{1b}(t) \leq q_{1b}^{max,m}$	$290 \leq q_{1b}(t) \leq 307$	$[Ah]$
	$q_{2b}^{min,h} \leq q_{2b}(t) \leq q_{2b}^{max,h}$	$72.5 \leq q_{2b}(t) \leq 77.5$	$[Ah]$
	$P_{ug}^{min,h} \leq P_{ug}(t) \leq P_{ug}^{max,h}$	$-2100 \leq P_{ug}(t) \leq 4200$	$[W]$
Middle level	N_p as in (7.32a)	10	$[h]$
	T_s as in (7.31)	300	$[s]$
	Q_y as in (7.32a)	$diag(1,1)$	
	R_u as in (7.32a)	800	
Constraints	$v_b^{min,m} \leq \tilde{v}_b(k) \leq v_b^{max,m}$	$12 \leq \tilde{v}_b(k) \leq 13$	$[V]$
	$i_b^{min,m} \leq \tilde{i}_b(k) \leq i_b^{max,m}$	$-10 \leq \tilde{i}_b(k) \leq 10$	$[A]$
	$q_{1b}^{min,m} \leq \tilde{q}_{1b}(k) \leq q_{1b}^{max,m}$	$288 \leq \tilde{q}_{1b}(k) \leq 308$	$[Ah]$
	$q_{2b}^{min,m} \leq \tilde{q}_{2b}(k) \leq q_{2b}^{max,m}$	$72 \leq \tilde{q}_{2b}(k) \leq 78$	$[Ah]$
	$P_{ug}^{min,m} \leq \tilde{P}_{ug}(k) \leq P_{ug}^{max,m}$	$-2100 \leq \tilde{P}_{ug}(k) \leq 4200$	$[W]$

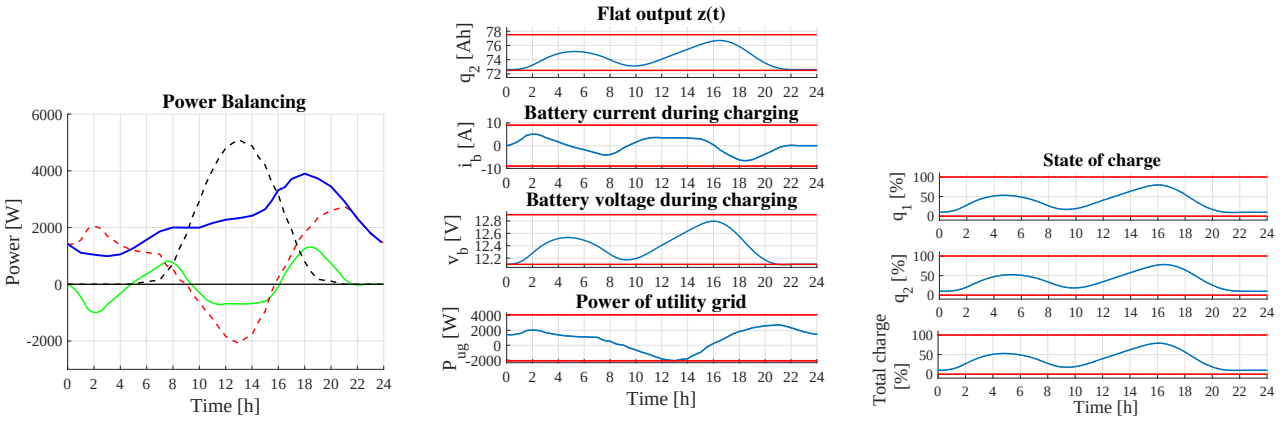
For the simulations we use a set of DS−100 PV modules (180 W peak PV generation) with external temperature and irradiation profiles gathered for a whole summer day (CIAT entreprise 2014). Through the model developed in **C37** implemented in MATLAB/Simulink, we obtain the power profiles for the PV system. For the ES system in (7.4), a collection of AGM 12−165 lead acid batteries (165 Ah battery capacity) is considered. Additionally, for the load profiles, two types of loads are provided, one for commercial use (4308 W peak demand) (National Renewable Energy Laboratory 2016), where the demand is higher during the day, and one for domestic use (3901 W peak demand) (Department of Energy Office of Energy Efficiency & Renewable Energy 2017), where the demand increases after 4 p.m.. Concerning the electricity price, the cost varies between 0.147 $[euros/kWh]$ from 4 p.m. to 10 p.m. and 0.116 $[euros/kWh]$ for the rest of the day. The DC microgrid is connected

to the UG (4200 W maximum UG generation) through a DC breaker, as shown in Figure 7.1a. MATLAB 2015a version is used for the simulations.

High level: First, the simulation results of the high level in Figure 7.7 are presented for both commercial (Figure 7.7a) and domestic load (Fig.7.7b) profiles generated through a B−spline parametrization, as in section 4.1, with $N = 18$ control points. The simulation is based on a constrained open−loop dynamics implementation in continuous−time over a horizon of 24 hours. The profiles of the PV and the loads are imported at the beginning with a sampling time equal to 600 s. Note that the power positive sign indicates the power supplied to the microgrid. Figure 7.7a and Figure 7.7b depict the power profiles generated by the ES and the UG within 24 hours (power balancing), taking into account the PV and the consumers' demand profiles for commercial and domestic use. For the commercial load profile, the demand is high during the day from 6 a.m. to 4 p.m.. On the other hand, the domestic load demand increases during the afternoon after 4 p.m..



(a) commercial load profile



(b) domestic load profile

Figure 7.7.: Power balancing, optimal reference profiles and state of charge of the ES system

In Table 7.3 percentages of the power produced or consumed by the sources (UG, PV, ES) and loads are presented as percentages of the total generated power. In the case of the commercial load demand (Fig.9), the PV generates 47% of the total power giving priority to the consumers' demand, while the rest is sold to the UG. While, in the afternoon, both the UG and ES contribute to the loads' supply especially after 4 p.m. when the energy generated from the PV is decreasing. Overall, only 1% is sold to the UG and 93% of the total energy produced is used by the consumers. On the other hand, for the domestic use profile, when the electricity price is high, the demand is low. The remained PV power (almost 12% of the total power consumed) is either sold to the UG or is used to charge the

batteries (also charged during the night, when the electricity cost is lower). In general, in both cases, whenever the PV power fully covers the consumers' demand, the remaining power is either stored in the ES system or sold to the UG.

Table 7.3.: Percentage of power with respect to the total power produced or consumed.

Load profile	Power	Power produced [%]	Power consumed [%]
Commercial	P_{ug}	46%	1% sold to the UG
	P_{es}	7%	6% for ES charging
	P_{pv}	47%	–
	P_{loads}	–	93% for load usage
Domestic	P_{ug}	40%	12% sold to the UG
	P_{es}	9%	9% for ES charging
	P_{pv}	51%	–
	P_{loads}	–	79% for load usage

Middle level: Afterwards, the results for the middle level are introduced using as reference the optimal profiles generated at the high level. As previously mentioned, in the middle level, we use MPC for reference tracking with a prediction horizon, N_p, equal to 10 h and a sampling time, T_s, equal to 300 s. The power profiles of the PV and the loads are updated following the sampling time T_s. According to Table 7.4, where the power produced and the power consumed are illustrated in respect to the total power, the optimal profiles obtained at the high level are very closely followed.

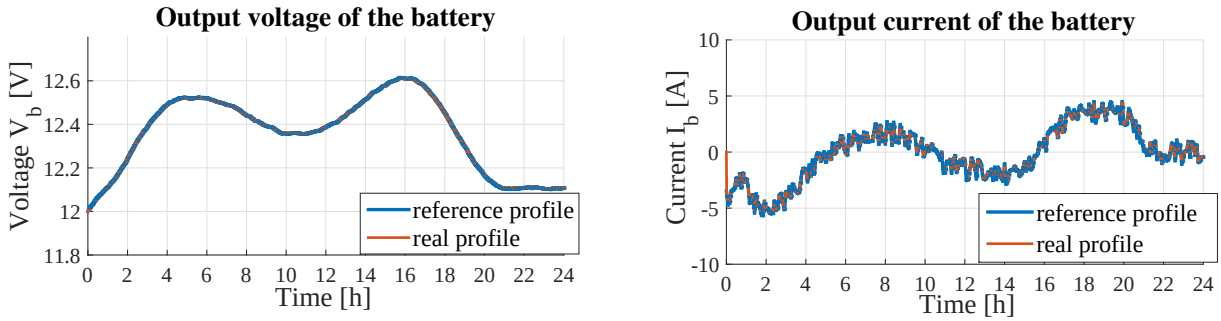
Table 7.4.: Percentage of power with respect to the total power produced or consumed. Comparison with high level optimal profiles.

Load profile	Power	Power produced [%]	Power consumed [%]	Power production difference from high level [%]	Power consumption difference from high level [%]
Commercial	P_{ug}	47%	1% sold to the UG	1%	0%
	P_{es}	6%	7% for ES charging	-1%	1%
	P_{pv}	47%	–	0%	–
	P_{loads}	–	92% for load usage	–	-1%
Domestic	P_{ug}	40%	12% sold to the UG	0%	0%
	P_{es}	9%	11% for ES charging	0%	−2%
	P_{pv}	51%	–	0%	–
	P_{loads}	–	77% for load usage	–	−2%

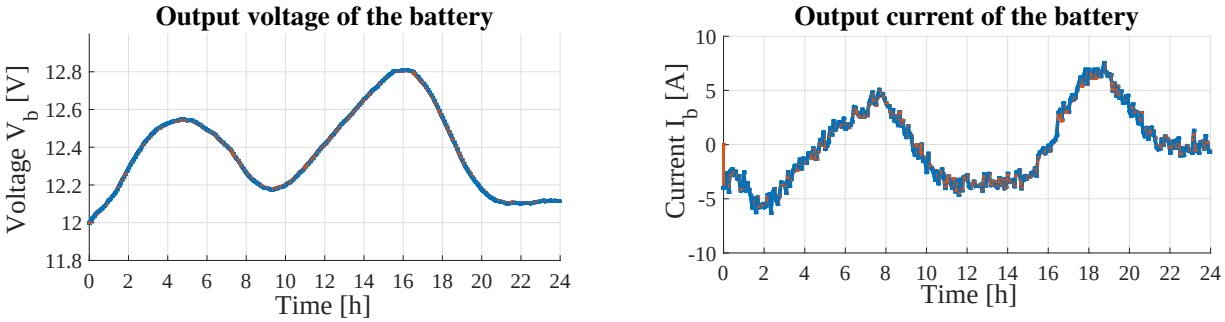
For the commercial load profile, a slight difference of 1% is observed in battery's charging and discharging and in the UG power production. The cost of the electricity increases at about 1% and from 4.090 raises to 4.140 *euros* for the real noise−affected profile. A similar case is also observed for the domestic load demand regarding the discharging of the battery. The computational time of the simulation lasts around 180 s for each load profile.

Low level: In the following, the results obtained at the low level are presented following the tracking profiles of the middle level for the battery current, i_b, and voltage, v_b, under perturbation (Fig.7.8). For the simulations, we develop the model of the ES (7.4, 7.6) in MATLAB/Simulink. The continuous−time simulation lasts about 10 s and demonstrates the proper operation of the converter which regulates very well the current and the voltage (Figure 7.8a, Figure 7.8b). The control variable

d_{2sc} is updated continuously (i.e., in simulation, this means, that the values are refreshed at each simulation sampling time equal to 300 s).

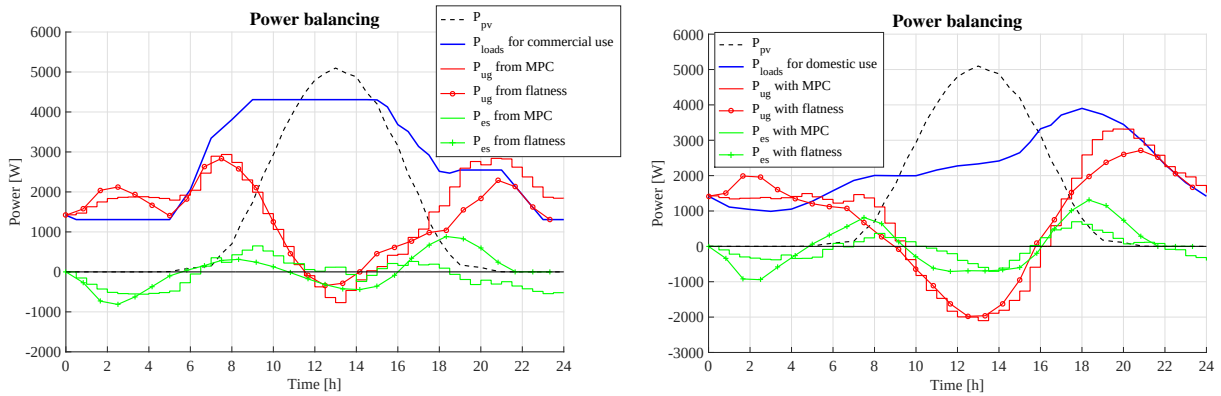


(a) commercial load profile



(b) domestic load profile

Figure 7.8.: Voltage and current tracking profiles.



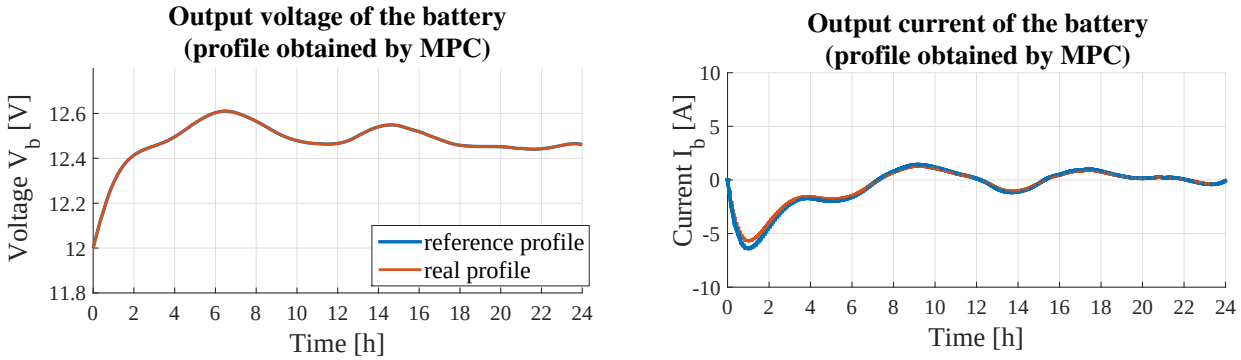
(a) commercial load profile

(b) domestic load profile

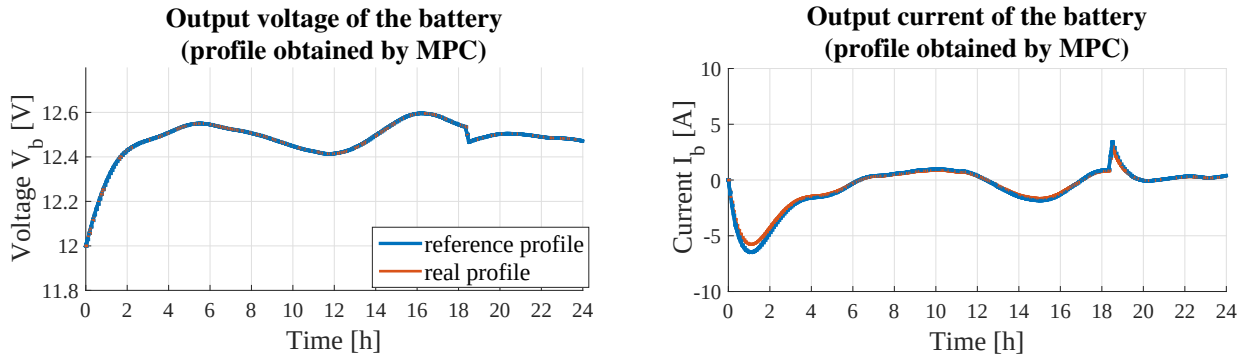
Figure 7.9.: Power balancing using commercial/domestic load profile. Comparisons of P_{es} and P_{ug} with optimal profiles obtained by MPC.

Comparisons: As previously described, the reference trajectories were obtained through differential flatness and B−spline parametrization. As a next step, the reference trajectories generation obtained through differential flatness and B−spline parametrization are compared with MPC, as presented also in (Parisio, Rikos, and Glielmo 2014) and (Iovine, Rigaut, Damm, De Santis, and Di

Benedetto 2018). The simulation results are presented in Figures 7.9a and 7.9b taking into account that Np is equal to 24 h with a sampling time T_s equal to 1800 s. The obtained trajectories are similar. In Figure 7.10a for the commercial load profile and in Figure 7.10b for the domestic load profile, we take the optimal profiles generated from MPC in order to follow them directly in the low level and calculate the electricity cost. As a result, the electricity cost for the commercial use profile is equal to 4.657 *euros* and in case of the domestic use is equal to 2.912 *euros*, which is higher than the electricity cost obtained from the flatness−based optimization problem. In both profiles with MPC, it is observed that the P_{ug} generates more power to satisfy the consumers' demand instead of exploiting the use of the battery. Consequently, although the power sold to the UG is higher in the MPC approach, the electricity cost remains higher than the one obtained from flatness, see Table 7.5.



(a) commercial load profile



(b) domestic load profile

Figure 7.10.: Voltage and current tracking profile at the low level obtained from MPC.

Table 7.5.: Total electricity cost for the UG-related power, in the MPC and flatness implementation.

	Load profile	MPC	Flatness
Electricity cost [euros]	Commercial	4.657	4.090
	Domestic	2.912	2.534

7.5.2. Economic MPC for the multi-source elevator system

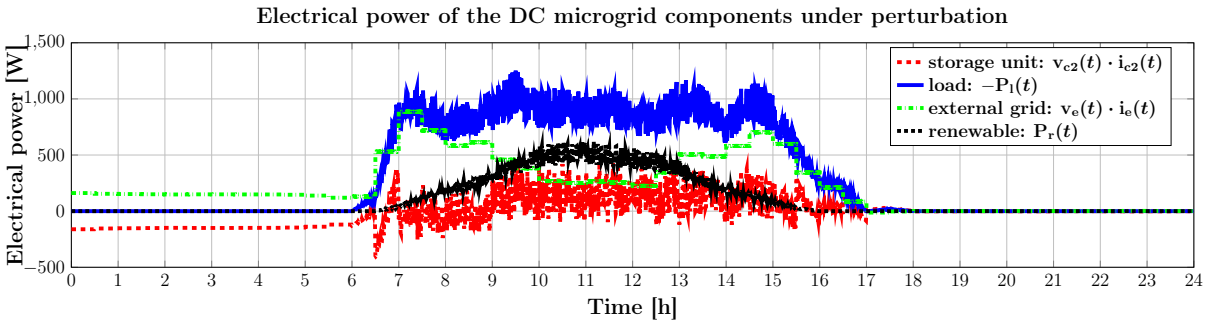
The numerical optimization problem for the load balancing detailed in **J12** is solved by using Yalmip (Löfberg 2004) and IPOPT (Wächter 2002) in Matlab 2013a. The constrained closed-loop dynamics implementation are done by using the *fsolve* function in Matlab 2013a with a fixed sampled time of 36 seconds over a horizon of 24 hours. Note that this sampling time corresponds to the discretization

of the continuous nonlinear dynamics. The update of the power profiles happens every 30 minutes, which is described by setting the value of the scheduling time step, h, at 0.5 hours, see Table 7.6.

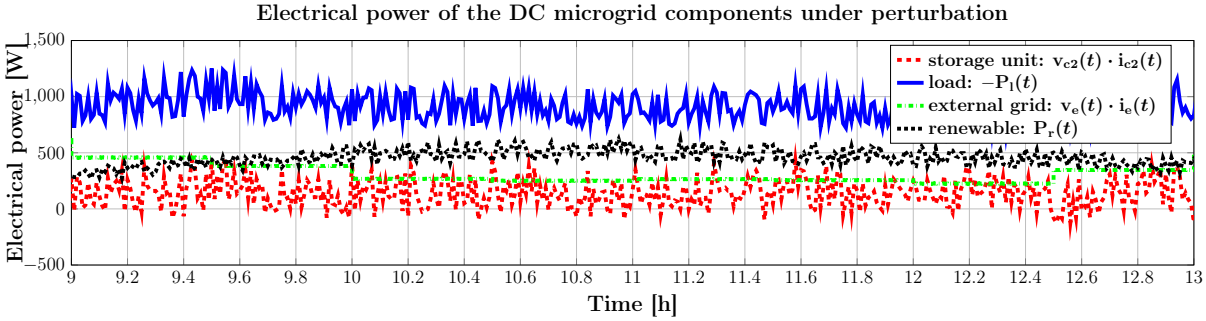
Table 7.6.: Numerical data for the multi-source elevator system components

Name	Notation	Value
Battery parameters	$\mathbf{Q}_1$ $[V]$	$[\ 13 \quad 13\ ]^T$
	$\mathbf{Q}_2$ $[V/C]$	diag $\{0.3036, 0.2024\}$
Battery constraints	$\mathbf{x}_{max}$ $[Ah]$	$[\ 73.2 \quad 109.8\ ]^T$
	$i_{b,min}$ $[A]$	-20
	$i_{b,max}$ $[A]$	20
Grid constraints	$i_{e,min}$ $[A]$	-8
	$i_{e,max}$ $[A]$	8
Bus voltage reference	v_{ref} $[V]$	380
Resistors	R_1 $[\Omega]$	0.012
	R_2 $[\Omega]$	0.015
	R_{bl} $[\Omega]$	0.31
	R_{be} $[\Omega]$	0.29
	R_{er} $[\Omega]$	0.23
	R_{rl} $[\Omega]$	0.19
Scheduling time step	h $[hour]$	0.5
Prediction horizon	N	48
Weighting parameter	$\gamma \in (0, 1)$	0.5

Note that, for this particular DC microgrid system the maximum supplied power (the sum of the PV and the battery) is always less than the load power. This means that with the numerical data we have at our disposal the microgrid cannot operate in islanded mode, i.e., disconnected from the external grid. Also, the maximum amount of power provided by the external grid is greater than the load power.



(a) power profiles



(b) zoom detail

Figure 7.11.: Electrical power charged/discharged by the DC components under disturbance.

Scenario 1: The perturbation is assumed to be bounded in a symmetrical tube. More precisely, the

electrical power of load and renewable source are within some uncertainty ranges:

$$P_l(t) \in \overline{P_l}(t)\,[1-\epsilon_{lmin}, 1+\epsilon_{lmax}]\,,\; P_r(t) \qquad \in \overline{P_r}(t)\,[1-\epsilon_{rmin}, 1+\epsilon_{rmax}]\,, \tag{7.44}$$

where $\epsilon_{(.)}$ are positive numbers taken here as $\epsilon_{lmin} = \epsilon_{lmax},\ \epsilon_{rmin} = \epsilon_{rmax}$ with the values set to 0.2.

Figure 7.11 describes the components, actual provided electrical power under the uncertainty-affected scenario. Since the current (and power) of the external grid is fixed, most of the fluctuation of the microgrid electrical power is absorbed by the battery. Further simulations under different scenarios and comparisons with a rule-based approach can be found in **J12**.

7.6. Discussions and conclusions

This chapter presented a multilevel supervision for DC microgrids. Two architectures were used as benchmarks: a meshed DC microgrid and a multi-source elevator system. First, the DC microgrid system has been presented in PH form, detailing in particular the ES system. Then, a constrained optimization based control approach was introduced which solved the power balancing problem. In a meshed topology, the optimization problem to solve becomes complicated since a model combines multiple sources, different timescales, nonlinearities and state/input constraints, to be satisfied simultaneously. Therefore, we built a controller that can manage and take into consideration all the aforementioned factors. The reference profile generation obtained with differential flatness and B−spline parametrization have been compared with the optimal profile provided by a MPC mechanism. This work has proven that differential flatness represents an accurate way to generate optimal profiles for subsequent use in the power balancing optimization problem.

As a short-term future work, further improvements in the constrained optimization problem are envisioned, e.g., energy dissipation minimization in the central transmission network by explicitly considering power losses in the cost. Furthermore, developing the transmission-line model will allow to analyze the robustness of the scheme under unexpected events, such as continuity of the system operation in case of a faulted line. In a long term future work, aspects, such as the proper sizing of the renewable sources or the batteries, addition of other elements (electrical vehicles, other battery models, other sources), will be studied. Additional properties of the PH formulation will be exploited from the viewpoint of stability and performance in the control scheme (passivity, energy conservation, the Hamiltonian as a candidate Lyapunov function).

We underline that electrical grid systems are challenging since within the same grid there may appear heterogeneous components functioning at wildly different time scales and with different constraints and requirements (Zafra-Cabeza, Velarde, and Maestre 2020). E.g., a well-studied but still challenging topic is the modeling of DC-DC converters (an actual issue due to the prominence of intermittent DC power sources such as solar panels). Approaches which employ flatness to characterize the nonlinear dynamics of the converter, (Gavagsaz-Ghoachani et al. 2016), are of great interest and will help provide a fully flat description of the grid dynamics.

8. Reliable power balancing for DC microgrid systems

This chapter presents a robust MPC (Model Predictive Control) approach for reserve balancing in DC microgrid systems under uncertainties like wind power and energy price variations and different types of fault events. The approach is based on a coherent framework of control tools, like mixed-integer programming and soft constrained MPC, for describing the microgrid components dynamics and the overall system control architecture. Fault tolerant strategies are inserted in order to ensure the proper amount of energy in the storage devices such that (together with the utility grid) the essential consumer demand is always covered. Furthermore, the robust MPC algorithm considers a variable-length prediction horizon which accounts for forecasts in energy price and renewable power over one day. The storage system is used to increase the utility of the demands and minimize the energy costs. The algorithm is tested for multiple fault types which affect the system (line and loss of power faults).

The chapter is based on the following papers:

C58 Kiebler, C., **I. Prodan**, F. Petzke, and S. Streif. "Reserve Balancing in a Microgrid System for Safety Analysis", pp. 1–6. In: *Proceedings of the 21th IFAC World Congress*. 2020. Berlin, Germany.

J7 **Prodan, I.**, E. Zio, and F. Stoican. "Fault tolerant predictive control design for reliable microgrid energy management under uncertainties". In: *Energy*, pp. 20–34, 2015. ISSN: 0360-5442. Elsevier.
DOI: 10.1016/j.energy.2015.08.009. WOS: 000365362700003. EID: 2-s2.0-84946026960

J6 **Prodan, I.** and E. Zio. "A model predictive control for reliable microgrid energy management". In: *International Journal of Electrical Power and Energy Systems*, pp. 399–409, 2014. ISSN: 0142-0615. Elsevier.
DOI: 10.1016/j.ijepes.2014.03.017. WOS: 000337855600043. EID: 2-s2.0-84899105409

8.1. Brief literature review

The flexible configuration and operation of microgrids helps to avoid cascading failures and, thus, blackouts and instabilities. Microgrids can be attached to the utility (grid-connected) and isolated (islanded) easily in case of faults or instability in the external grid. Once the disturbance in the main grid has cleared, microgrids can be connected again and system reliability is improved (Lo Prete et al. 2012), (Bracco, Delfino, Pampararo, Robba, and Rossi 2014). Hence, realistic modeling and optimization for efficient, reliable and economic planning, operation and control of microgrids are very important and still open issues.

Proposed techniques for the minimization of microgrid operating costs include heuristics (Sanseverino, Di Silvestre, Ippolito, De Paola, and Lo Re 2011), mathematical programming (Hawkes, Brett, and Brandon 2009) and priority rules (Tsikalakis and Hatziargyriou 2011). A method based on the combination of empirical mean, dynamic programming and MPC has been used in (Hooshmand, Poursaeidi, Mohammadpour, Malki, and Grigoriads 2012) for solving a power management problem within a microgrid in islanded mode operation. In (Parisio and Glielmo 2011), a preliminary study has addressed the application of a receding horizon approach for optimizing microgrid operations while satisfying time-varying demands and operational constraints. In (Pérez, Beltran, Aparicio, and Rodriguez 2013), an approach based on MPC has been designed to manage in real-time the power production of a grid-tied photovoltaic plus energy storage power plant with a reduced energy storage system capacity. It is important to mention that all these papers do not consider explicitly the detailed modeling of the microgrid components, the constraints description and the interaction between the independent components of the micrigrid system. Instead, abstract mathematical models are used to embody the practical and functional behavior of the components and the focus is on the formulation of the optimization problem for minimizing costs. In (R. Negenborn, Houwing, De Schutter, and Hellendoorn 2009) a model predictive controller is applied for controlling the energy flows inside a household system equipped with a "micro" combined heat and power unit. In addition, the household can buy and sell electricity from/to the energy supplier; heat and electricity can be stored in specific storage devices. In (Zervas, Sarimveis, Palyvos, and Markatos 2008), MPC is used for energy scheduling on a hydrogen-based microgrid without batteries. In (Qi, J. Liu, and P. Christofides 2012), predictive control is applied for renewable energy power management with battery storage in a water desalination plant. In (Khalid and Savkin 2010), MPC is performed for a plug-in renewable energy source with battery storage. The electrical power transferred to the network and that stored in the battery are the control variables considered. Finally, (Real, Arce, and Bordons 2007) presents the application of hybrid modeling control for a photovoltaic-fuel cell power plant.

In view of increasing availability of renewable energy sources and political efforts to encourage clean energy, an EMS (Energy Management System) capable to adapt the grid power distribution to the renewable energy variability is required. The technical literature provides an extensive number of robust smart grids controller implementation with different goals. In **J6** a MPC algorithm for reliable microgrid energy management regarding uncertainties in the forecast is proposed and extended to fault cases in **J7**. Herein the MPC algorithm considers soft constraints to enable feasibility under faults and disturbances. The algorithm enables robustness against one broken line (N-1 security). In (X. Wu and Conejo 2017) the most critical facilities were protected to minimize worst-case loads after physical attacks (faults), such that at least N-1 security holds. Here, a tri-level min-max-min problem is used, in order to minimize possible damages (the attacker tries to maximize the damage, while the operator handles the damage minimizing the cost). In (Khodabakhsh and Sirouspour 2016) the battery usage is optimized, while a multi-variant Gaussian distribution is used to model uncertainties in energy price and demands. In addition, (Wytock, Moehle, and Boyd 2017) proposed a scenario-based robust MPC approach, in which the worst case of all the generated scenarios is considered. (Khodaei 2014) emphasizes the resiliency for smart grid under islanded conditions, while in (C. Chen, J. Wang, Qiu, and D. Zhao 2016) the energy distribution within connected smart grids after natural disasters is analyzed. Furthermore (Rahimiyan, Baringo, and Conejo 2014) provides a simple EMS and data records for the energy price and wind power for a 24 hours interval. An approximated minimax robust

approach is proposed for improving the utility of demands under forecast uncertainties. Herein the approximation considers a reduced set of disturbance scenarios.

8.2. Contributions

The overall goal is to implement a realistic and flexible control scheme where:

- Costs, constraints, profiles are taken into account into a centralized constrained optimization problem (i.e., via a model predictive control design);
- Depending on external events and energy costs the user may receive only its essential demand or up to the entire desired demand;
- Faults at the generator output level are explicitly accounted in both robust (by managing the minimal storage requirements) and adaptive fashion (change of constraints and costs, as a function of system state – healthy, under fault, under recovery);
- The battery charge/discharge cycles are penalized in the cost function, in order to account for the battery wear and tear;
- The user load is partitioned into two components:
  - essential loads, that is, demands of power related to essential processes (e.g., electricity in an operating room, fail-safe modules in critical systems) which have to be met;
  - non-essential loads, that can be reduced or shed during supply constraints or emergency situations (e.g., standby devices, day-time lighting and the like).
- Reliable functioning of the microgrid is maintained under parameter variations, noises and fault events (e.g., generator output outages). Especially for the latter case, we provide an adaptive control which can handle fault events via subsequent control reconfiguration.
- A robust economic MPC scheme with variable prediction horizon length is implemented to handle profile variations and maximize demand utility.

8.3. Microgrid system description using mixed-integer formulations

Any microgrid, regardless on the particular constructive details, will contain several types of components as illustrated in Figure 8.1: power generators (e.g., hydro, wind turbine, photovoltaic panels and the like) denoted as

$$\{G_i\}_{i=1\cdots N_g}, \text{ where } N_g \text{ represents the number of generators,} \tag{8.1}$$

storage elements (e.g., batteries)

$$\{S_j\}_j = 1\cdots N_s, \text{ where } N_s \text{ represents the number of electrical storages,} \tag{8.2}$$

consumers (e.g., large cooling houses),

$$\{D_k\}_{k=1,\cdots,N_d}, \text{ where } N_d \text{ represents the number of consumers,} \tag{8.3}$$

internal interconnections between these elements and (if not in islanded mode) external connections to the utility power grid denoted as E.

With reference to Figure 8.1, let us define in Table 8.1 the generic interconnection signals which appear in a typical microgrid system. Formally, the microgrid system is topologically characterized by a directed graph, as illustrated in Figure 8.1. The nodes are the previously described components

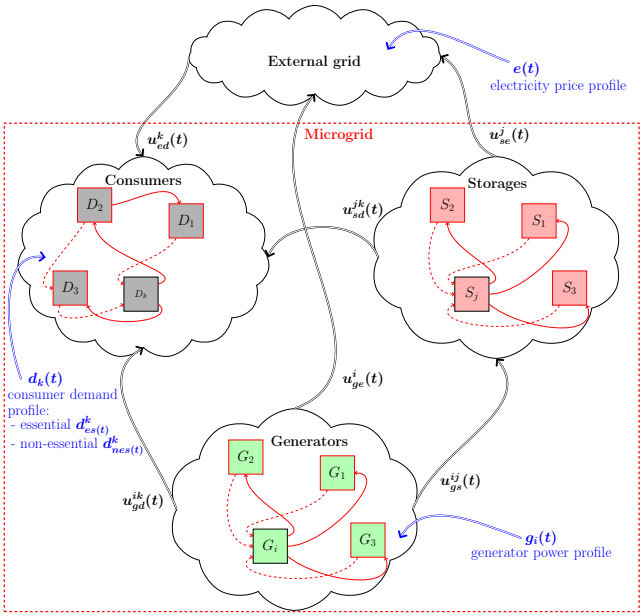


Figure 8.1.: Interconnections in a typical microgrid system.

and the edges are the links between these components. For the sake of keeping compact the system equations, let us introduce the adjacency matrices of Table 8.2 to characterize the links between components, e.g., $M_{gs}(i,j) = 1$ means that there exists a connection between the generator G_i and the storage unit S_j (i.e., the link u^{ij}_{gs}) and $M_{gs}(i,j) = 0$ denotes the absence of the connection.

Notation	Description
$u^{ij}_{gs}(t)$ $[W]$	electrical power transmitted by the generator G_i to the electrical storage S_j at time step t.
$u^{ik}_{gd}(t)$ $[W]$	electrical power transmitted by the generator G_i to the consumer D_k at time step t.
$u^{i}_{ge}(t)$ $[W]$	electrical power transmitted by the generator G_i to the external grid E at time step t.
$u^{jk}_{sd}(t)$ $[W]$	electrical power transmitted by the electrical storage S_j to the consumer D_k at time step t.
$u^{j}_{se}(t)$ $[W]$	electrical power transmitted by the electrical storage S_j to the external grid E at time step t.
$u^{k}_{ed}(t)$ $[W]$	electrical power transmitted by the external grid E to the consumer D_k at time step t.

Table 8.1.: General interconnection signals in a typical microgrid.

Notation	Description
$M_{gs} \in \{0,1\}^{N_g \times N_s}$	characterizes the existence of a link from the generator G_i to the electrical storage S_j.
$M_{gd} \in \{0,1\}^{N_g \times N_d}$	characterizes the existence of a link from the generator G_i to the consumer D_k.
$M_{ge} \in \{0,1\}^{N_g \times 1}$	characterizes the existence of a link from the generator G_i to the external grid E.
$M_{sd} \in \{0,1\}^{N_s \times N_d}$	characterizes the existence of a link from the electrical storage S_j to the consumer D_k.
$M_{se} \in \{0,1\}^{N_s \times 1}$	characterizes the existence of a link from the electrical storage S_j to the external grid E.
$M_{ed} \in \{0,1\}^{1 \times N_d}$	characterizes the existence of a link from the external grid E to the consumer D_k.

Table 8.2.: Adjacency matrices specifying the links between the microgrid components.

For future use, let us also define the number of non-zero entries in an adjacency matrix[1]:

$$N_{gs} = \sum_{M_{gs}(i,j)\neq 0} M_{gs}(i,j), \qquad N_{gd} = \sum_{M_{gd}(i,k)\neq 0} M_{gd}(i,k), \tag{8.4}$$

$$N_{ge} = \sum_{M_{ge}(i)\neq 0} M_{ge}(i), \qquad N_{sd} = \sum_{M_{sd}(j,k)\neq 0} M_{sd}(j,k), \tag{8.5}$$

$$N_{se} = \sum_{M_{se}(j)\neq 0} M_{se}(j), \qquad N_{ed} = \sum_{M_{ed}(k)\neq 0} M_{ed}(k), \tag{8.6}$$

for all $i = 1, \ldots, N_g$, $j = 1, \ldots, N_s$, $k = 1, \ldots, N_d$.

In the following, we detail the elements introduced from the point of view of control optimization: dynamics governing the internal functioning, profiles to be followed, constraints and costs affecting the components. Note that, since each of the components of a given type behaves similarly, we describe as much as possible generic rules of functioning and discuss particularities if and when it is the case.

8.3.1. Electrical storage units

Storage dynamics

In the proposed microgrid setting of Figure 8.1, the components that we assume to be directly controllable are the storage units (through the rates of charge and discharge). To model the operation in the charge and discharge modes, we introduce the following Linear-Time Invariant (LTI) dynamics[2]:

$$x_j(t+1) = (1-\sigma_j)x_j(t) + \sum_{M_{gs}(i,j)\neq 0} u_{gs}^{ij}(t) - \sum_{M_{sd}(i,j)\neq 0} u_{sd}^{jk}(t) - \sum_{M_{se}(j,k)\neq 0} u_{se}^{j}(t) + w_j(t), \tag{8.7}$$

[1]For example, $N_{gs} \in \mathbb{R}$ denotes the number of non-zero entries in M_{gs}. In other words, N_{gs} represents the number of interconnections between the components G_i and S_j of the microgrid.

[2]While $x_j(t)$ denotes the energy stored at time step t, $u_{gs}^{ij}(t)$, $u_{sd}^{jk}(t)$, $u_{se}^{j}(t)$ and $w(t)$ are electrical powers multiplied by the sampling time $\Delta t = 1$ hour. For a streamlined presentation, Δt is hidden in (8.7) and in the following equations.

with the *mixed-integer* conditions:

$$\begin{cases} 0 \leq u_{gs}^{ij}(t) \leq M\alpha_j(t), & \forall i \text{ with } M_{gs}(i,j) \neq 0, \\ 0 \leq u_{sd}^{jk}(t) \leq M(1-\alpha_j(t)), & \forall k \text{ with } M_{sd}(j,k) \neq 0, \\ 0 \leq u_{se}^{j}(t) \leq M(1-\alpha_j(t)), & \text{if } \exists j \text{ with } M_{se}(j) \neq 0, \end{cases} \tag{8.8}$$

where $x_j(t) \in \mathbb{R}$ represents the amount of energy stored in S_j at time step t, $u_{gs}^{ij}(t) \in \mathbb{R}$ and $u_{sd}^{jk}(t) \in \mathbb{R}$, $u_{se}^{j}(t) \in \mathbb{R}$ denote the charging and respectively the discharging components, σ_j denotes the hourly self-discharge decay and the additive noise $w_j(t)$ accounts for the various sources of variation appearing in the storage dynamics (e.g., temperature influences, inverter performances and the like).

While presumably for certain storage units both charge and discharge operations are possible simultaneously, here we consider batteries which have a switched behavior (they function either in charge or discharge mode). This is modeled in the supplementary mixed-integer conditions (8.8): $\alpha_j(t) \in \{0,1\}$ is an auxiliary binary variable which switches between the modes of functioning. Specifically, when $\alpha_j(t) = 1$ the battery is in discharge mode (since the input $u_{gs}^{ij}(t)$ is forced into zero) whereas, for $\alpha_j(t) = 0$ the battery is in charge mode (since the outputs $u_{sd}^{jk}(t)$, $u_{se}^{j}(t)$ are forced into zero). The scalar M is an appropriately chosen constant (i.e., significantly larger than the rest of the variables and playing the role of a relaxation constant).

Storage constraints

Each storage unit has bounds on the quantity of stored energy

$$B_{min}^{j} \leq x_j(t) \leq B_{max}^{j}, \tag{8.9}$$

where $B_{min}^{j}, B_{max}^{j} \in \mathbb{R}$ and on the rate of charge/discharge at time t

$$V_{min}^{j} \leq \Delta x_j(t) \leq V_{max}^{j}, \tag{8.10}$$

where $V_{min}^{j}, V_{max}^{j} \in \mathbb{R}$ and $\Delta x_j(t) = x_j(t) - x_j(t-1)$ represents the charge/discharge variation.

Batteries have additional particularities, which stem from their physical characteristics. In particular, the minimum capacity, B_{min}^{j}, is determined from the Depth of Discharge (DoD), which is used to describe how deeply the battery can be discharged (S. Diaf, D. Diaf, Belhamel, Haddadi, and Louche 2007):

$$B_{min}^{j} = (1 - DoD)B_{max}^{j}. \tag{8.11}$$

The battery capacity B_{max} decreases as a function of time, usage and conditions of work, as given by:

$$B_{max}^{j} = \beta_j(t)\bar{B}_{max}^{j}, \tag{8.12}$$

where $\bar{B}_{max}^{j}$ represents the nominal capacity of the battery and parameter $\beta_j(t) \in (0,1)$ models the capacity decay and its value is generally empiric.

Storage cost

Depending on the type of storage unit, we may need to take into account wear and tear issues. While in the short-medium time point of view it makes sense to exploit aggressively the storage unit (i.e., the energy flows to and from the storage unit at all times, to ensure that energy costs are minimized) in the longer time view it is counter-productive to over-use a component if the price of replacing it is larger than the actual gains from its use. To account for this, depending on the nature of the energy storage component, different terms are considered as penalization in the cost function:

i) penalize repeated discharging modes:

$$C_s(t) = \sum_{j=1}^{N_s} \alpha_j(t); \tag{8.13}$$

ii) penalize repeated charging modes:

$$C_s(t) = \sum_{j=1}^{N_s} (1 - \alpha_j(t)); \tag{8.14}$$

iii) penalize the switch between charge/discharge modes:

$$C_s(t) = \sum_{j=1}^{N_s} (\alpha_j(t) - \alpha_j(t-1)). \tag{8.15}$$

E.g., for a hidro pumping storage unit it would make sense to penalize the charging mode (as it takes power to operate the pump); on the other hand, for a battery we may wish to minimize repeated cycles of charge/discharge[3].

8.3.2. Power generators

Generator dynamics

Consider the generator system G_i in Figure 8.1, whose dynamics can be described by:

$$g_i(t+1) = f(g_i(t), v_i(t)), \quad i = 1, \ldots, N_g \tag{8.16}$$

where $f(\cdot,\cdot) : \mathbb{R} \to \mathbb{R}$ represents the possibly nonlinear dynamics of the generator, $g_i(t) \in \mathbb{R}$ is the electrical power given by the generator and $v_i(t) \in \mathbb{R}$ denotes the internal and external parameters which influence the generator output.

In our control scheme we consider renewable generators and as such we may ignore the control aspects, the generator is assimilated to an external source of power which provides as much as possible given the external environment, e.g., wind velocity will decide the actual power output for a wind turbine generator. The addition of photovoltaic dynamic systems, electrical vehicles (EV) and the like would entail some modifications in the constraints formulation, the optimization problem and the related simulation validation, with no conceptual changes from the methodological viewpoint.

Generator constraints

The generator output can be sent to other various microgrid components: to the storage unit for further use, directly to the users or even to the external grid for selling. Since the electrical power sent to these components cannot be larger than the actual output, the following constraint is imposed:

$$0 \leq \sum_{M_{gs}(i,j)\neq 0} u_{gs}^{ij}(t) + \sum_{M_{gd}(i,k)\neq 0} u_{gd}^{ik}(t) + \sum_{M_{ge}(i)\neq 0} u_{ge}^{i}(t) \leq g_i(t), \tag{8.17}$$

[3]The battery cycle life is defined as the number of complete charge - discharge cycles a battery can perform before its nominal capacity falls below 80% of its initial rated capacity (Divya and J. Østergaard 2009). Key factors affecting cycle life are time t and the number of charge-discharge cycles completed. It is also important to mention that the cycle life depends on the Depth of Discharge (DoD) (8.11) and that it is assumed that the battery is fully charged and discharged each cycle. If the battery is only partially discharged each cycle, then, the cycle life can be larger: it is, therefore, important that DoD be clearly defined when specifying the cycle life.

with $\sum_{M_{gs}(i,j)\neq 0} u_{gs}^{ij}(t) \geq 0$, $\sum_{M_{gd}(i,k)\neq 0} u_{gd}^{ik}(t) \geq 0$ and $\sum_{M_{ge}(i)\neq 0} u_{ge}^{i}(t) \geq 0$.

Generator profile

As stated earlier, the wind turbine generator output mainly depends on external factors (in particular, the wind velocity). In order to have a good estimation of the output, we make use of wind profiles, as the ones obtained from real numerical data of a reliability test system (Grigg et al. 1999).

For a wind turbine generator, $f(\cdot,\cdot)$ in (8.16) is nonlinear, that is, the output of the generator depends nonlinearly on the wind speed and can be roughly partitioned into three regions of functioning (starting-up, nominal functioning and dangerous wind levels) as described by the following equation (K. Z. Østergaard, Stoustrup, and Brath 2009) and also shown in Figure 8.2:

$$g_i(t) = \begin{cases} 0, & if\ v_i(t) < v_{ci}, \\ P_r \cdot \frac{v_i(t)-v_{ci}}{v_r-v_{ci}} \cdot \Delta t, & if\ v_{ci} \leq v_i(t) < v_r, \\ P_r \cdot \Delta t, & if\ v_r \leq v_i(t) < v_{co}, \\ 0, & if\ v_i(t) > v_{co}, \end{cases} \tag{8.18}$$

where v_{ci} [m/s], v_r [m/s] and v_{co} [m/s] are the *cut-in*, *rated* and *cut-out* wind speeds, respectively, and P_r [W] is the rated power of the wind turbine.

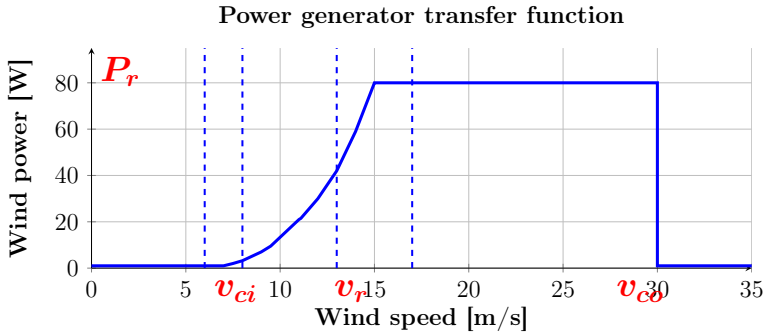


Figure 8.2.: Reference power curve for the generator as a function of the wind speed.

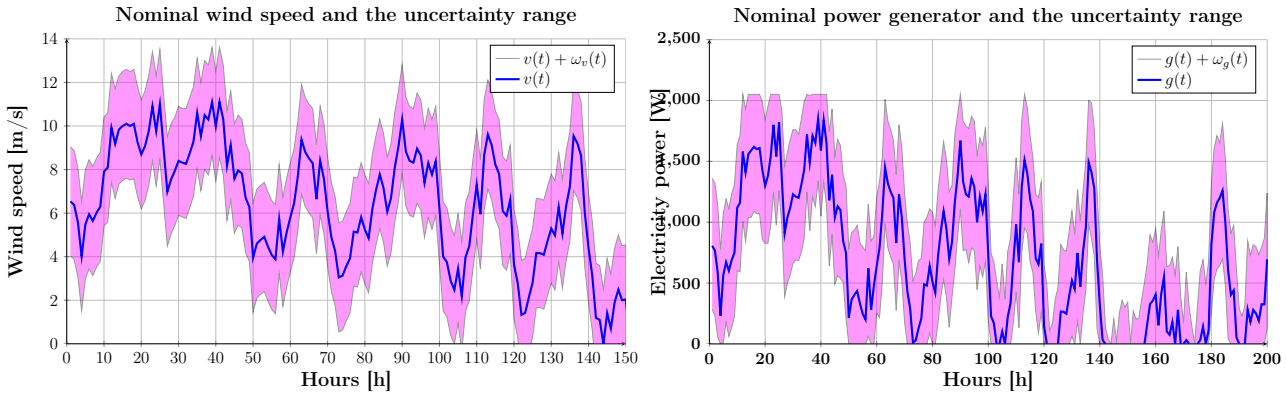
Note that the wind profile is sampled at one hour interval which from the viewpoint of wind variation is quite large. For this reason and due to measurement uncertainties we have considered along the nominal wind profile a band of disturbances, Figure 8.3a. Passing these values through the dynamics (8.16) of the generator we obtain a band of possible power output values, Figure 8.3b.

8.3.3. Consumers

Consumer constraints

A user D_k in Figure 8.1 can consume electricity from three sources, i.e., the electrical storage, the power generator and the external grid. The user demand of electricity, denoted as $d_k(t)$, need not be inelastic: that is, we may partition the consumer demand $d_k(t)$ into essential, $d_{es}^k(t) \in \mathbb{R}$ and non-essential demand, $d_{nes}^k(t) \in \mathbb{R}$, respectively[4]. Therefore, for a reliable management of the energy system it is necessary to ensure that at time t the electricity purchased by the consumers from the

[4]Note that when and if a power shortcoming occurs, we want to be able to safely cut from the non-essential part while still covering the essential part.



(a) Wind speed profile and uncertainty band

(b) Generator power profile and uncertainty band

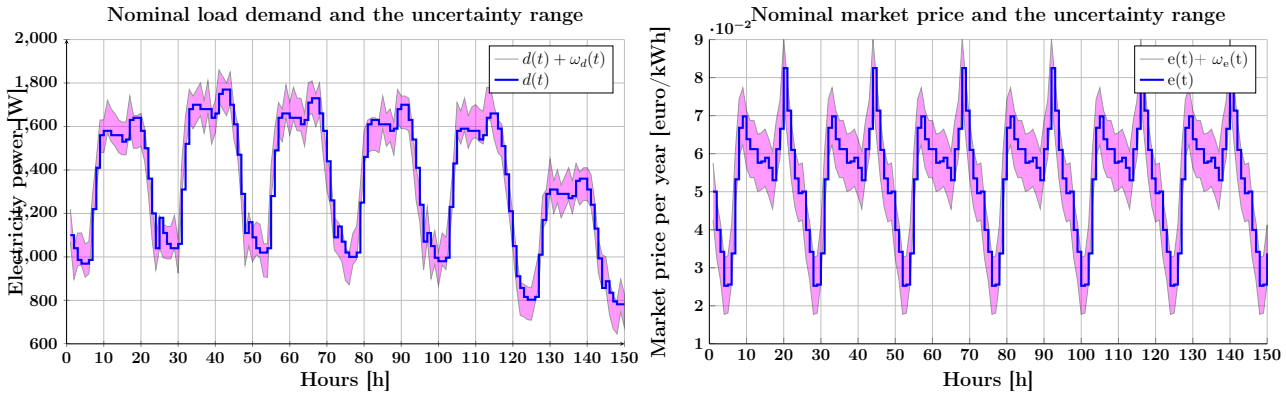
Figure 8.3.: Reference profiles.

three sources will satisfy at least the essential demands. This translates into the following constraint:

$$d_{es}^{k}(t) < \sum_{M_{gd}(i,k)\neq 0} u_{gd}^{ik}(t) + \sum_{M_{sd}(j,k)\neq 0} u_{sd}^{jk}(t) + \sum_{M_{ed}(k)\neq 0} u_{ed}^{k}(t) \leq d_{es}^{k}(t) + d_{nes}^{k}(t). \tag{8.19}$$

Consumer demand profile

The consumer demand profile usually exhibits periodicity (both daily, weekly and seasonal) and can therefore be predicted with a good degree of accuracy as presented in detail in **J6**. The real profiles may differ from the nominal values due to external factors, uncertainty and so forth, as shown in Figure 8.4a which illustrates both the nominal profiles and the bands in which they lie.



(a) Consumer load profile and uncertainty band

(b) Electricity market price profile and uncertainty band

Figure 8.4.: Reference profiles.

Consumer cost

In the cost function, we penalize for the difference between actual and required demand:

$$C_d(t) = \sum_{k=1}^{N_d} d_{es}^{k}(t) + d_{nes}^{k}(t) - \left[\sum_{M_{sd}(j,k)\neq 0} u_{sd}^{jk}(t) + \sum_{M_{gd}(i,k)\neq 0} u_{gd}^{ik}(t) + \sum_{M_{ed}(k)\neq 0} u_{ed}^{k}(t) \right]. \tag{8.20}$$

8.3.4. External grid

Electricity price profile

If the microgrid is not in islanded mode, the price of electricity in the external grid, denoted as $e(t) \in \mathbb{R}$, is relevant in taking sell (when expensive) and buy (when cheap) decisions. Depending on the differing country rules, the electricity price might be pre-scheduled by an external authority or be decided by market pressures. In any case, we may make use of price profiles in order to optimize the energy usage. Such a profile, taken from (Grigg et al. 1999) is illustrated in Figure 8.4b.

External grid cost

The electricity price will be used to penalize buying and encourage selling with the cost described by the following relation:

$$C_e(t) = e(t) \cdot \left(\sum_{M_{ed}(k)\neq 0} u^k_{ed}(t) - \sum_{M_{ge}(i)\neq 0} u^i_{ge}(t) - \sum_{M_{se}(j)\neq 0} u^j_{se}(t) \right), \tag{8.21}$$

where we make the simplifying assumption that both sell and buy prices are equal, without loss of generality.

8.4. Optimization-based control for electrical storage scheduling

The goal of this section is to provide a realistic control strategy for the microgrid system, and the storage scheduling in particular. Taking into account the dynamics, constraints, costs and profiles discussed in the previous section, we can now write a constrained optimization problem for the global microgrid system. To do so let us define first the global state $\mathbf{x}(t) = \left[x_1(t)^T, \quad \ldots, \quad x_{N_s}(t)^T\right]^T \in \mathbb{R}^{N_s}$, which concatenates the storage units' states[5], the additive disturbance $\mathbf{w}(t) = \left[w_1(t), \ldots w_{N_s}(t)\right]^T \in \mathbb{R}^{N_s}$ and the input $\mathbf{u}(t) = \begin{bmatrix} u^{ij}_{gs}(t) & u^{ik}_{gd}(t) & u^i_{ge}(t) & u^{jk}_{sd}(t) & u^j_{se}(t) & u^k_{ed}(t) & \alpha_j(t) \end{bmatrix}^T \in \mathbb{R}^{N_u} \times \{0,1\}^{N_s}$, which concatenates in lexicographical order all the control inputs appearing throughout the microgrid system[6], with $N_u = N_{gs} + N_{gd} + N_{ge} + N_{sd} + N_{se} + N_{ed}$ denoting the number of real valued control variables and $N_{gs}, N_{gd}, N_{ge}, N_{sd}, N_{se}, N_{ed} \in \mathbb{R}$ as defined in (8.4)–(8.6).

Hence, with the above notation and the dynamic model of the storage unit detailed in (8.7), (8.8) we formulate the centralized system:

$$\mathbf{x}(t+1) = \underbrace{\begin{bmatrix} 1-\tau_1 & & \\ & \ddots & \\ & & 1-\tau_{N_s} \end{bmatrix}}_{\mathbf{A}} \mathbf{x}(t) + \underbrace{\begin{bmatrix} \delta^b_{gs} & \mathbf{0} \end{bmatrix}}_{\mathbf{B}_{ch}} \mathbf{u}(t) + \underbrace{\begin{bmatrix} \mathbf{0} & \delta^b_{sd} & \delta^b_{se} & \mathbf{0} \end{bmatrix}}_{\mathbf{B}_{disch}} \mathbf{u}(t) + \mathbf{w}(t), \tag{8.22}$$

with matrices $\mathbf{A} \in \mathbb{R}^{N_s \times N_s}$ $\mathbf{B}_{ch} \in \mathbb{R}^{N_s \times (N_u + N_s)}$, $\mathbf{B}_{disch} \in \mathbb{R}^{N_s \times (N_u + N_s)}$ decomposed in $\delta^b_{gs} \in \mathbb{R}^{N_s \times N_{gs}}$, $\delta^b_{sd} \in \mathbb{R}^{N_s \times N_{sd}}$, $\delta^b_{se} \in \mathbb{R}^{N_s \times N_{se}}$ denoting hereafter the corresponding inputs from $\mathbf{u}(t)$ (such that (8.22) is equivalent with (8.7)) and the matrices $\mathbf{0}$ chosen with appropriate dimensions.

The dynamics (8.22) are subject to charge/discharge mixed-integer conditions (as in (8.8)), magnitude constraints (as in (8.9)), power variation (as in (8.10)), generator constraints (as in (8.17)),

[5] Since these are the only microgrid units with internal dynamics.

[6] Note that the auxiliary binary variables $\alpha_j(t)$ appear as control inputs, since they model the charge and discharge operations of the storage units (see also Section 8.3.1).

bounds on the energy transfer[7], user demand validation (as in (8.19)):

$$\mathbf{0} \leq \mathbf{B}_{ch}\mathbf{u}(t) \leq M\begin{bmatrix}\mathbf{0} & I_{N_s}\end{bmatrix}\mathbf{u}(t), \tag{8.23}$$

$$\mathbf{0} \leq \mathbf{B}_{disch}\mathbf{u}(t) \leq M\begin{bmatrix}\mathbf{0} & I_{N_s}\end{bmatrix}(\mathbf{1}-\mathbf{u}(t)), \tag{8.24}$$

$$\mathbf{B}_{min} \leq \mathbf{x}(t) \leq \mathbf{B}_{max}, \tag{8.25}$$

$$\mathbf{V}_{min} \leq \Delta\mathbf{x}(t) \leq \mathbf{V}_{max}, \tag{8.26}$$

$$\mathbf{0} \leq \mathbf{G}\mathbf{u}(t) \leq \mathbf{g}(t), \tag{8.27}$$

$$\mathbf{d}_{es}(t) \leq \mathbf{D}\mathbf{u}(t) \leq \mathbf{d}_{es}(t) + \mathbf{d}_{nes}(t), \tag{8.28}$$

$$\mathbf{0} \leq \mathbf{u}(t) \leq \bar{\mathbf{u}}(t). \tag{8.29}$$

More to the point, (8.23)–(8.24) describe in a compact form the battery charge/discharge conditions and (8.25)–(8.26) describe the battery magnitude/variation constraints with $\mathbf{B}_{min} = \begin{bmatrix}\dots & B^j_{min} & \dots\end{bmatrix}^T$, $\mathbf{B}_{max} = \begin{bmatrix}\dots & B^j_{max} & \dots\end{bmatrix}^T$, $\mathbf{V}_{min} = \begin{bmatrix}\dots & V^j_{min} & \dots\end{bmatrix}^T$, $\mathbf{V}_{max} = \begin{bmatrix}\dots & V^j_{max} & \dots\end{bmatrix}^T$, for all $j = 1, \dots, N_s$. Next, condition (8.27) describes the power distribution of the generator towards different sources with matrix $\mathbf{G} = \begin{bmatrix}\delta^g_{gs} & \delta^g_{gd} & \delta^g_{ge} & \mathbf{0}\end{bmatrix} \in \mathbb{R}^{N_s\times(N_u+N_s)}$ decomposed in $\delta^g_{gs} \in \mathbb{R}^{N_s\times N_{gs}}$, $\delta^g_{gd} \in \mathbb{R}^{N_s\times N_{gd}}$, $\delta^g_{ge} \in \mathbb{R}^{N_s\times N_{ge}}$ and $\mathbf{g}(t) = \begin{bmatrix}\dots & g_i(t)^T & \dots\end{bmatrix}^T$, for all $i = 1, \dots, N_g$, (8.28) describes the necessary power distribution to the users with matrix $\mathbf{D} = \begin{bmatrix}\mathbf{0} & \delta^d_{gd} & \mathbf{0} & \delta^d_{sd} & \mathbf{0} & \delta^d_{ed} & \mathbf{0}\end{bmatrix} \in \mathbb{R}^{N_s\times(N_u+N_s)}$, $\delta^d_{gd} \in \mathbb{R}^{N_s\times N_{gd}}$, $\delta^d_{sd} \in \mathbb{R}^{N_s\times N_{sd}}$, $\delta^d_{ed} \in \mathbb{R}^{N_s\times N_{ed}}$, $\mathbf{d}_{es}(t) = \begin{bmatrix}\dots & d^k_{es}(t)^T & \dots\end{bmatrix}^T$, $\mathbf{d}_{nes}(t) = \begin{bmatrix}\dots & d^k_{es}(t)^T & \dots\end{bmatrix}^T$, for all $k = 1, \dots, N_d$. Finally, (8.29) describes the physical limitations on the energy transfer between the different microgrid components with $\bar{\mathbf{u}}(t) = \begin{bmatrix}\bar{u}^{ij}_{gs} & \bar{u}^{ik}_{gd} & \bar{u}^{i}_{ge} & \bar{u}^{jk}_{sd} & \bar{u}^{j}_{se} & \bar{u}^{k}_{ed} & \mathbf{1}\end{bmatrix}^T$.

The goal is to control the storage units such that the electricity taken from the external grid is minimized, the users demand is fulfilled, the electrical storages utilization is minimized and the generators utilization is maximized. For a reliable microgrid energy management we consider the recursive construction of an optimal open-loop control sequence $u = \{\mathbf{u}(t), \mathbf{u}(t+1), \cdots, \mathbf{u}(t+N_{\mathrm{p}}-1)\}$ over a *finite* constrained receding horizon N_p, which leads to a feedback control policy by the effective application of the first control action as system input:

$$\mathbf{u}^* = \arg \min_{\mathbf{u}(t),\mathbf{u}(t+1),\cdots,\mathbf{u}(t+N_{\mathrm{p}}-1)} \sum_{l=0}^{N_{\mathrm{p}}-1} \gamma^l \cdot C(t+l), \tag{8.30}$$

subject to the set of constraints defined in (8.23)–(8.29) for $t = 0, \dots, N_p - 1$.

In (8.30) the length of the prediction horizon is denoted by N_p, $\gamma \in (0, 1)$ represents a weighting parameter and the cost function is the sum of the cost functions in (8.21), (8.20) and (8.15):

$$C(t) = \gamma_e C_e(t) + \gamma_d C_d(t) + \gamma_s C_s(t), \tag{8.31}$$

with $\gamma_e, \gamma_d, \gamma_s \in \mathbb{R}$ some constant parameters whose values reflect the importance given to the different terms of the cost. Using the above notations, the cost function can be written in a general form as:

$$C(t) = (e(t)\mathbf{F} - \mathbf{D})\,\mathbf{u}(t) + \mathbf{1}^T(\mathbf{d}_{es}(t) + \mathbf{d}_{nes}(t)) + \mathbf{H}\Delta\mathbf{u}(t), \tag{8.32}$$

with $\mathbf{F} = \begin{bmatrix}\mathbf{0} & -\delta^f_{ge} & \mathbf{0} & \delta^f_{se} & \delta^f_{ed} & \mathbf{0}\end{bmatrix}^T \in \mathbb{R}^{1\times(N_u+N_s)}$ decomposed in $\delta^f_{ge} \in \mathbb{R}^{N_g\times N_{ge}}$, $\delta^f_{se} \in \mathbb{R}^{N_s\times N_{se}}$, $\delta^f_{ed} \in \mathbb{R}^{1\times N_{ed}}$ denoting the corresponding inputs from $\mathbf{u}(t)$ as previously mentioned, $\mathbf{H} = \begin{bmatrix}\mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & 1\end{bmatrix}^T$ and $\Delta\mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}(t-1)$. The last term in (8.32) is chosen such that it penalizes the switch between charge/discharge modes as detailed in (8.15).

[7]The constraints in (8.29) stand from physical limits on the energy transfer for each control signal, with $\bar{u}^{ij}_{gs}, \bar{u}^{ik}_{gd}, \bar{u}^{i}_{ge}, \bar{u}^{jk}_{sd}, \bar{u}^{j}_{se}, \bar{u}^{k}_{ed} \in \mathbb{R}$.

Note that it remains a matter of choice on how to adjust the weights in order to decide the optimum usage of the storages. For example, a large γ_s can lead to the battery being never used and hence its role being insignificant. On the other hand, too small a value leads to a premature capacity loss for the battery.

The profiles introduced in Section 8.3 appear as parameters here (e.g., the consumer load $d(t)$, the energy cost $e(t)$, etc.). The cost is variable in the sense that the weight matrix may change with time due to the variation in energy price[8], but otherwise is linear in the input values. Therefore, we may denote this problem as one of Mixed-Integer Linear Programming (MILP). For this type of problems, efficient solvers exist and a reasonably large prediction horizon can be used **J3**.

Note also that the increase of the prediction horizon length in (8.30) means that the optimization problem minimizes the cost along this entire horizon. The cost function, in turn, may be affected by uncertainties in a way that cost values further away along the prediction horizon are less reliable than the ones closer to the present. A solution is to assign in the optimization problem less importance to the cost values which are further in the future. This can be done by varying the weight $\gamma \in (0, 1)$ in (8.30) associated to each cost value over the prediction horizon. The weight value decreases exponentially, with speed depending on the value taken by the rate γ (see, for example (Hovd and Braatz 2001)).

Next, we discuss the integration of fault tolerant strategies within the optimization-based control approach for microgrid energy management.

8.4.1. Fault tolerant control strategies

The fault events considered in this microgrid system are external, in the sense that some of the renewable generators may fail to provide power to the battery, user and external grid (i.e., total output failure, $g_i(t) \to 0$). This is modeled by modifying the generator constraint (8.27) into:

$$0 \leq \mathbf{G}\mathbf{u}(t) \leq \mathbf{B}_f \mathbf{g}(t), \tag{8.33}$$

where matrix $\mathbf{B}_f = \text{diag}(\{0, 1\}^{N_g})$ characterizes the functioning of the generators. If all elements on the diagonal are '1' it means that all the generators are healthy and provide the expected power output. Whenever a fault occurs, one or more of the diagonal elements becomes '0'.

Since the fault detection and isolation are evident, what remains is to design a fault tolerant strategy. We aim at ensuring that the necessary quantity of energy is always stored in the electrical storage units such that, together with the external grid, the essential user demand for the k-th user is covered during the fault:

$$\sum_{M_{sd}(j,k)\neq 0} B^j_{min}(t) \geq \sum_{\tau=t}^{t+MTTR_i-1} \max \left[0, d^k_{es}(\tau) - \overline{u}^k_{ed} - \sum_{M_{gd}(i,k)\neq 0, \mathbf{B}_f(i,i)\neq 0} u^{ik}_{gd}(\tau) \right], \tag{8.34}$$

with $MTTR_i$ (Mean Time to Repair) representing the expected fault duration for the i-th generator[9]. Relation (8.34) states that we expect to have at all times in the storage units enough energy to cover whatever power deficit remains after we consider the maximum amount given by the external grid $\overline{u}^k_{ed}$ (see also (8.29)) and by the remaining healthy generators $\sum_{M_{gd}(i,k)\neq 0, \mathbf{B}_f(i,i)\neq 0} u^{ik}_{gd}(\xi)$. In fact, during this time we may consider that the remaining healthy generators send all their power to the users:

$$\sum_{k=1}^{N_d} u^{ik}_{gd}(\tau) = g_i(\tau), \ \forall i = 1, \ldots, N_g, \forall \tau = t, \ldots, t + MTTR_i - 1. \tag{8.35}$$

Using (8.34) and (8.35) and imposing capacity bounds specific to the storage units we formulate an

[8] Note that we consider the same price for electricity selling and buying.

[9] If condition (8.34) can be interpreted as a passive (robust) approach, then satisfying only the user essential demands as in (8.19) can be considered as an active measure.

optimization problem whose output are the profiles of minimum storage ensuring essential demand:

$$\begin{aligned} B^1_{h,min}(t)\dots B^{N_s}_{h,min}(t) = & \min_{B^1_{min}(t)\dots B^{N_s}_{min}(t)} \sum_{j=1}^{N_s} B^j_{min}(t) \\ & \text{such that (8.34) and (8.35) are verified and} \\ & (1-DOD)B^j_{max} \leq B^j_{min}(t) \leq B^j_{max},\ \forall j = 1,\dots,N_s. \end{aligned} \tag{8.36}$$

Furthermore, denoting $\mathbf{B}_{h,min}(t) = \left[B^1_{h,min}(t)\dots B^{N_s}_{h,min}(t)\right]$ the optimization problem (8.36) becomes:

$$\begin{aligned} \mathbf{B}_{h,min}(t) = & \min_{\mathbf{B}_{h,min}(t)} \mathbf{B}_{min}(t)\mathbf{1}^T \\ & \text{such that (8.34) and (8.35) are verified and} \\ & (1-DOD)B^j_{max} \leq B^j_{min}(t) \leq B^j_{max},\ \forall j = 1,\dots,N_s. \end{aligned} \tag{8.37}$$

Remark 8.1. Note that we made a series of simplifying assumptions. Firstly, we assume that there is a single generator in fault and that the fault is persistent. Secondly, we assume that the energy to be stored will never surpass the actual storage capacity (i.e., $B^j_{min}(t) \leq B^j_{max}$). Since ensuring that the storage units are large enough is a design choice, we chose not to discuss it here.

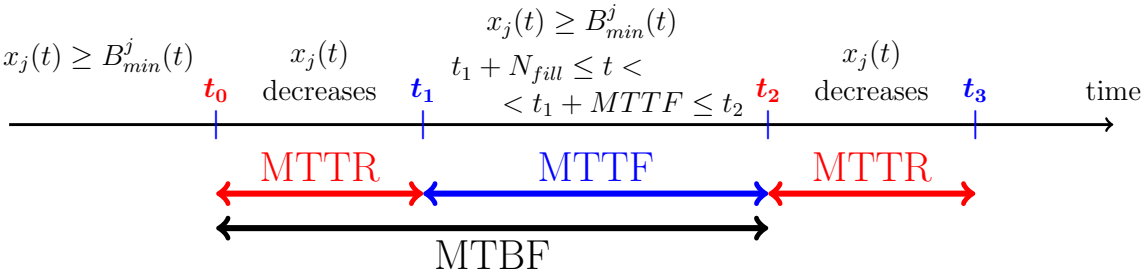


Figure 8.5.: Fault tolerant strategy illustration.

Condition $\mathbf{x}(t) > \mathbf{B}_{min}(t) = \mathbf{B}_{h,min}(t)$ is the capacity bound we wish to ensure at all times but it cannot be respected when under fault or immediately after. Hence, three modes of functioning are identified: nominal functioning after a previous fault, steady state nominal functioning and under fault. To thoroughly analyze the microgrid modes of functioning under generator failures, we make use of the proof of concept illustration in Figure 8.5 where a typical functioning scenario is depicted: at t_0 the microgrid enters into faulty functioning due to the i-th generator having an outage. This continues up to time t_1 when the fault is solved and the microgrid regains its healthy functioning. The next time instants t_3 and t_4 retrace the same scenario (fault and subsequent recovery). Therefore, the three modes of functioning which provide the solution of safe stopping the microgrid system are:

- Nominal functioning after a previous fault: the battery is empty or close to empty due to its usage during the recovery phase (ended at t_1 as in Figure 8.5); the goal is, then, to charge it as fast as possible in order to reach the safe storage quantities obtained in (8.36). This cannot be done instantaneously due to practical limitations (charge variation for the battery for example). Hence, we need first to determine a feasible recharging interval denoted as $N_{fill} \in \mathbb{R}$. This can be done e.g., by solving a minimal time problem:

$$\begin{aligned} N_{fill} = \min \quad & \tau \\ & \text{such that } \mathbf{x}(t_1+\tau) \geq \mathbf{B}_{h,min}(t_1+\tau),\ \forall j = 1,\dots,N_s \text{ and} \\ & \text{equations (8.22)} - \text{(8.29) are verified for } t = t_1,\dots,t_1+\tau. \end{aligned} \tag{8.38}$$

We either provide condition $\mathbf{x}(t_1 + N_{fill}) \geq \mathbf{B}_{h,min}(t_1 + N_{fill})$ as a terminal constraint (not obvious since $t_1 + N_{fill}$ will probably be beyond the prediction horizon) or with a suboptimal approach, by forcing $\mathbf{B}_{min}(t)$ to increase towards the expected safe value $\mathbf{B}_{h,min}(t)$ in the filling time N_{fill}:

$$\mathbf{B}_{r,min}(\tau) = \mathbf{B}_{min}(t_1) + \frac{\tau - t_1}{N_{fill}}(\mathbf{B}_{h,min}(t_1 + N_{fill}) - \mathbf{B}_{min}(t_1)), \tag{8.39}$$

for all $\tau = t_1, \ldots, t_1 + N_{fill}$.
Thus, in the transitional period we will increase the minimum storage requirement step by step[10]:

$$\mathbf{x}(t) > \mathbf{B}_{min}(t) = \mathbf{B}_{r,min}(t), \ \forall t = t_1, \ldots, t_1 + N_{fill}. \tag{8.40}$$

- Steady state nominal functioning: sufficient time has passed since the last fault. The battery already verifies (8.34) at the current instant and has to verify it for all future instances of the prediction horizon:

$$\mathbf{x}(t) > \mathbf{B}_{min}(t) = \mathbf{B}_{h,min}(t), \ \forall t > t_1 + N_{fill} \tag{8.41}$$

- Under fault: we enforce only the physical minimal storage requirements (8.11) for the interval $t = t_0, \ldots, t_1$.

Algorithm 5: Fault tolerant scheme

```
Input: tstop
mode='healthy';
while t ≤ tstop do
    if checkFault=true & mode='healthy' then
        construct signature fault matrix B_f;
        update constraint (8.33);
        mode='faulty';
    end
    if checkFault=false & mode='faulty' then
        update constraint (8.33);
        compute N_fill as in (8.38);
        lastFault = t;
        mode='healthy';
    end
    if mode='healthy' then
        if t < lastFault + N_fill then
            B_min = B_r,min(t) given as in (8.39);
        else
            B_min = B_h,min(t) given as in (8.41);
        end
    end
    if mode='faulty' then
        give B_min as in (8.11);
    end
    solve the optimization problem (8.30);
end
```

The above remarks and constructions can be synthesized in algorithm Algorithm 5 which is to be followed by a supervisor unit. Since we wish that in general the load demands are respected, we transform constraint (8.19) into a "soft constraint" through the addition of a slack variable.

8.4.2. Soft constraints

The primordial requirement of the microgrid scheme is to fulfill the user demands as best as possible. To this end, we made use of constraint (8.21) to force at all times the coverage of the essential demand

[10] If $N_{fill} < \min MTTF_i$ we have leeway to choose a larger N_{fill}. This will soften the storage requirements.

and of the cost (8.28) to tilt the control towards providing complete user coverage (both essential and non-essential). The issue in this approach is that choosing the weight γ^d associated to the cost in (8.31) makes the difference between covering only the essential demand or both essential and non-essential demand (i.e., if too low relatively to the other weights, the optimization problem chooses to provide only essential demand, if it is too large, numerical issues appear).

To correctly characterize the weight γ^d, we make use of the "hard/soft constraints" notion. Basically, we consider that the original "hard constrained" problem forces the validation at all times of equality

$$\mathbf{Du}(t) = \mathbf{d}_{es}(t) + \mathbf{d}_{nes}(t). \tag{8.42}$$

In this case, there may be initial conditions such that there exists no input sequence for which the problem is feasible ((8.42) cannot be validated). In such a situation the optimization solver will find no solution and consequently will not provide any input for the microgrid. This is in general considered unacceptable and practical MPC implementations therefore include some way of relaxing the constraints to ensure that the optimization problem is always feasible. We proceed here by adding slack variables $\epsilon^k(t) \geq 0$, which make the constraints (8.42) "soft":

$$\mathbf{d}_{es}(t) + \mathbf{d}_{nes}(t) \leq \mathbf{Du}(t), \tag{8.43}$$

$$\mathbf{Du}(t) + \epsilon(t) = \mathbf{d}_{es}(t) + \mathbf{d}_{nes}(t), \quad \epsilon(t) \geq 0. \tag{8.44}$$

Using soft constraints, means relaxing some of the constraints appearing in the MPC formulation by the addition of $\epsilon(t)$, while the cost includes terms which penalize the constraint violation:

$$C_d(t) = \gamma^d ||\epsilon(t)||_1 \tag{8.45}$$

The conditions of exact correspondence between "hard" and "soft" constraints follow classic optimization results (Fletcher 1987) and express conditions on the Lagrangian multipliers of the original optimization problem (due to the fact that we have "softened" only constraints (8.42) we consider only the associated Lagrangians):

$$\gamma^d \geq \max_{\mu_d} ||D^T \mu_d||_\infty. \tag{8.46}$$

Finding the minimum for problem (8.46) can be done by solving a bi-level optimization problem, e.g., via a MILP formulation as in (Hovd and Stoican 2013).

8.5. Simulation results

The forthcoming simulations use the numerical data of a test system (IEEE RTS–96) developed for bulk power system reliability evaluation studies (Grigg et al. 1999). Consider the microgrid of Figure 8.6 which contains $N_g = 2$ generators, $N_s = 2$ electrical storage units and $N_d = 2$ consumers. All components are characterized by the dynamic models, profiles, constraints and costs presented in detail in Section 8.3. The microgrid system is connected to the main utility grid through a transformer. Here, the transformer provides electrical power from the external grid as well as information about the electrical market price, which plays an important role as the storage units may decide to keep or use the stored energy depending on the external price. The consumers have also the possibility to take electrical power from the external grid when the renewable resources are not available (or sufficient).

The interactions between the independent components of the microgrid are most important for accomplishing the consumers objectives. As shown in Figure 8.6 there are various links between the components of the microgrid, which determine the energy flow. The adjacency matrices characterizing the links between the microgrid components (see also Section 8.3) are:

$$M_{gs} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}; \quad M_{gd} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}; \quad M_{ge} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \quad M_{sd} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad M_{se} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \quad M_{ed} = \begin{bmatrix} 1 & 1 \end{bmatrix}; \tag{8.47}$$

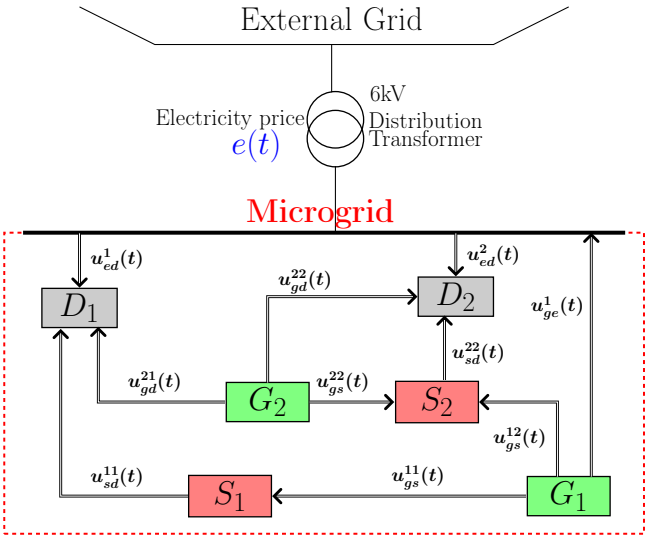


Figure 8.6.: Microgrid architecture

with the number of non-zero entries $N_{gs} = 3$, $N_{gd} = 2$, $N_{ge} = 1$, $N_{sd} = 2$, $N_{se} = 0$, $N_{ed} = 2$ as defined in (8.4) – (8.6). The global state of the microgrid system $\mathbf{x}(t) = \left[x_1(t)^T, x_2(t)^T\right]^T \in \mathbb{R}^2$ concatenates the states of the two storage units and the input
$\mathbf{u}(t) = [u^{11}_{gs}(t)\ u^{12}_{gs}(t)\ u^{22}_{gs}(t)\ u^{21}_{gd}(t)\ u^{22}_{gd}(t)\ u^{1}_{ge}(t)\ u^{11}_{sd}(t)\ u^{22}_{sd}(t)\ u^{2}_{ed}(t)\ \alpha_1(t)\ \alpha_2(t)]^T \in \mathbb{R}^{N_u} \times \{0,1\}^{N_s}$, with $N_u = 10$, $N_s = 2$ concatenates all the control inputs of the microgrid system.

Using the relation in (8.22) we obtain the matrices which characterize the centralized system:

$$\mathbf{A} = \begin{bmatrix} 1-\sigma_1 & 0 \\ 0 & 1-\sigma_2 \end{bmatrix} \in \mathbb{R}^{2\times 2}; \quad \mathbf{B}_{ch} = \begin{bmatrix} \delta^b_{gs} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2\times 12}; \tag{8.48}$$

$$\mathbf{B}_{disch} = \begin{bmatrix} \mathbf{0} & \delta^b_{sd} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2\times 12} \text{ with } \delta^b_{gs} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad \delta^b_{sd} = -I_2.$$

The constraints (8.27), (8.28) are described by:

$$\mathbf{G} = \begin{bmatrix} \delta^g_{gs} & \delta^g_{gd} & \delta^g_{ge} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2\times 12}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{0} & \delta^d_{gd} & \mathbf{0} & \delta^d_{sd} & \mathbf{0} & \delta^d_{ed} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2\times 12}, \tag{8.49}$$

with $\delta^g_{gs} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\delta^g_{gd} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$, $\delta^g_{ge} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\delta^d_{gd} = \delta^d_{sd} = \delta^d_{ed} = I_2$.

The cost function (8.32) is described by:

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & -\delta^f_{ge} & \mathbf{0} & \delta^f_{ed} & \mathbf{0} \end{bmatrix}^T \in \mathbb{R}^{1\times 12}, \quad \mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \in \mathbb{R}^{1\times 12}, \tag{8.50}$$

with $\delta^f_{ge} = \begin{bmatrix} -1 \end{bmatrix}$, $\delta^f_{ed} = \begin{bmatrix} 1 & 1 \end{bmatrix}$.

Considering the discretization step $\Delta t = 1$ hour and the reference profiles described in Section 8.3 we provide in Table 8.7 the rest of the numerical values of the parameters used for the simulation.

Figure 8.8 illustrates the charge stored in battery $x_1(t)$ along the simulation horizon (i.e., 200 hours) and also the charge variation $\Delta x_1(t)$. Note that whenever there is a discharge the value of the variation becomes negative, which corresponds to the binary variable codification $\alpha_1(t) = 1$). The reverse holds for the charging process. Also, note that the binary variable $\alpha_1(t)$ was scaled and offset such that it is well represented in the down plot of Figure 8.8. Hence, as it can be observed in Figure 8.8 the battery load and charge constraints are always verified. Similarly, Figure 8.9 illustrates the charge stored in battery $x_2(t)$ along the same simulation horizon (i.e., 200 hours) as well as the charge variation $\Delta x_1(t)$,

Batteries parameters			
$\sigma_1 = \sigma_2 = 13 \cdot 10^{-4}$	$M = 9 \cdot 10^3$	$B^1_{min} = B^2_{min} = 12 \cdot 10^2\ [Wh]$	$B^1_{max} = B^2_{max} = 6 \cdot 10^3\ [Wh]$
$V^1_{min} = V^2_{min} = -1.5 \cdot 10^3\ [W]$		$V^1_{max} = V^2_{max} = 1.5 \cdot 10^3\ [W]$	
Power generators parameters			
$P_r = 6 \cdot 10^3\ [W]$	$v_{ci} = 3\ [m/s]$	$v_r = 12\ [m/s]$	$v_{co} = 20\ [m/s]$
Prediction horizon	Control input constraints		
$N_p = 7$	$\bar{\mathbf{u}} = [1.2\ 1.5\ 1.2\ 1.5\ 1.5\ 1.2\ 1.7\ 2\ 1.5\ 1.5\ 10^{-3}\ 10^{-3}]^T \cdot 10^3\ [W]$		
Weight parameters in (30) and (31)			
$\gamma_b = 1$	$\gamma_d = 0.5$	$\gamma_e = 1$	$\gamma = 1$

Figure 8.7.: Numerical data for the microgrid components.

correspondingly to the binary variable codification $\alpha_2(t)$.

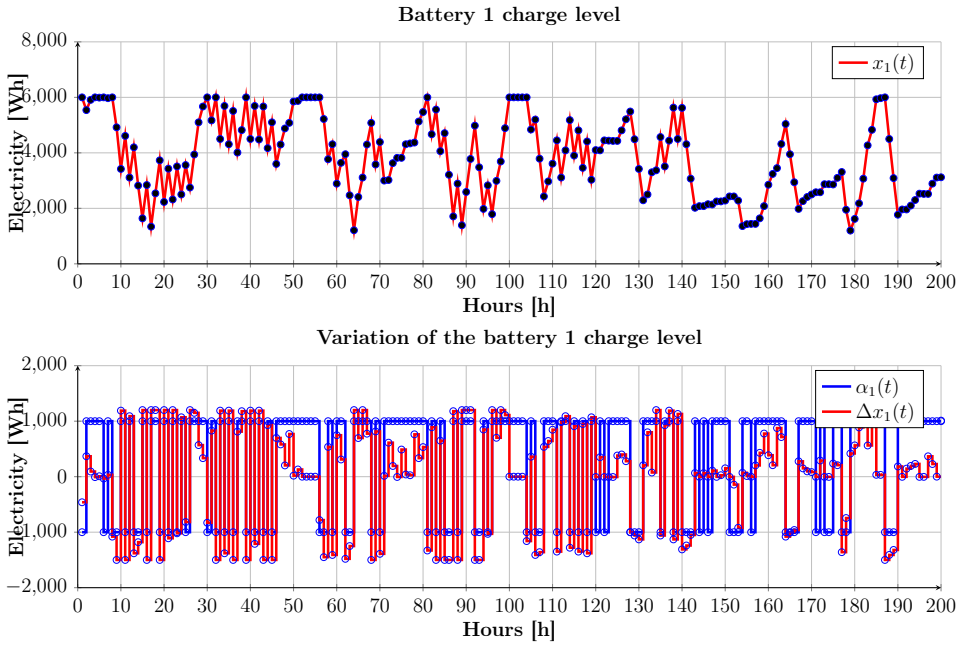


Figure 8.8.: Battery 1 charge level and variation.

Figures 8.10, 8.11, 8.13 and 8.14 illustrate the various control signals of the microgrid. It can be seen that there is a complex interplay between the storage devices and the rest of the microgrid components. Particularly, at some instants, it appears that it is more efficient to give and take energy from the external grid rather than send it to the users.

Figure 8.10 illustrates the electrical power sent by the generators to the storage units. Note that the 1st generator sends to both storage units whereas the 2nd sends only to the second storage unit.

In Figure 8.11 we illustrate the same behavior but this time for the electrical power sent from the generators to the users and the external grid. In Figures 8.10 and 8.11 we have shown different types of power flow between generators and the other microgrid components. In Figure 8.12 we now show which is the partitioning of the power flow of a generator (the first) between all the other components in order to make clear that a generator may decide (depending on cost and constraints) to partition its power accordingly.

In Figure 8.13 we depict the power flows sent from the storage units to the users. As it can be seen,

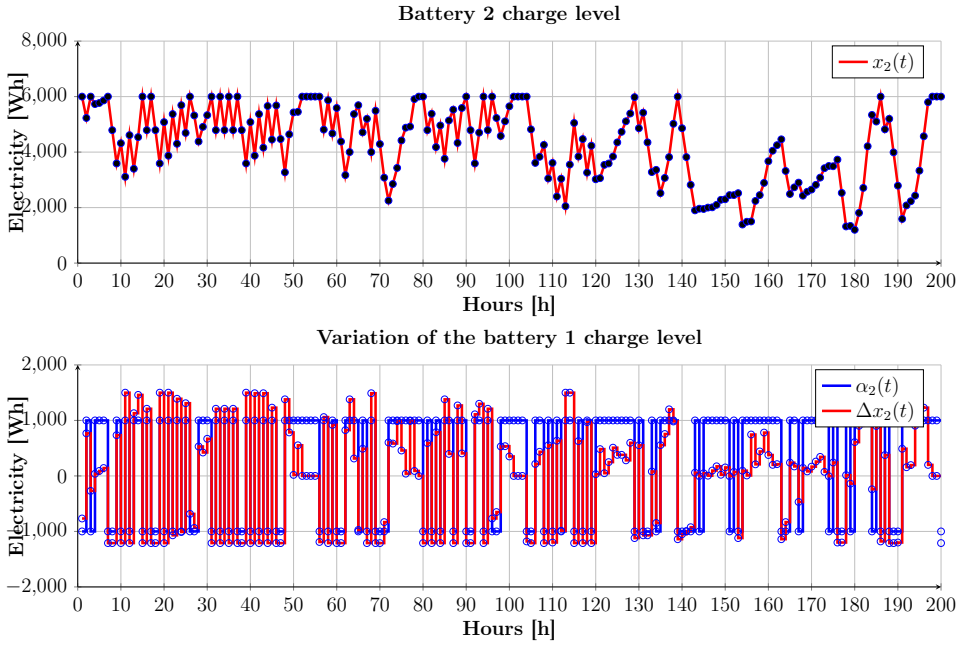


Figure 8.9.: Battery 2 charge level and variation.

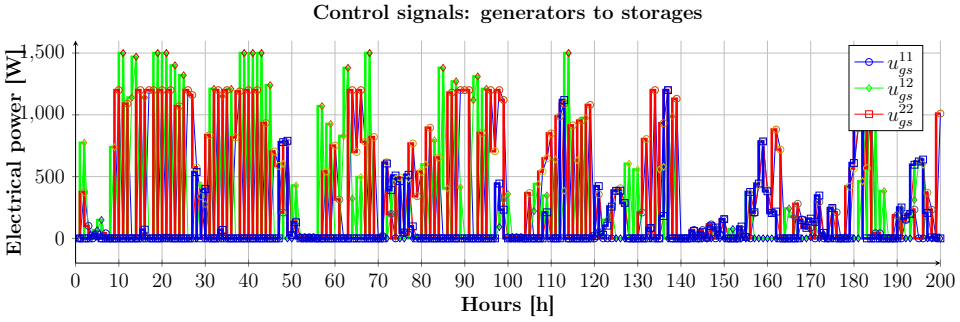


Figure 8.10.: Electrical power transmitted by the generators to electrical storage components.

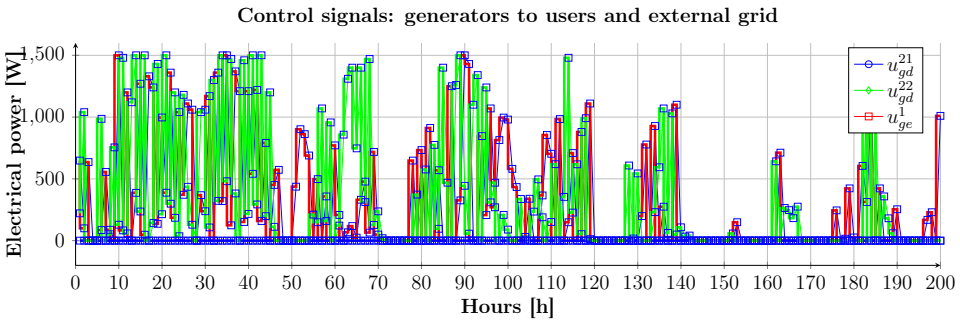


Figure 8.11.: Electrical power transmitted by the generators to users.

the behavior appears to be “on-off”: either the batteries are discharging to the maximum or they are in charge mode (there are relatively few charging values when the power sent is not close to maximal). Figure 8.14 depicts the power received by both users from the external grid.

To characterize the fulfillment of the demands of the users we depict in Figure 8.15 the demand

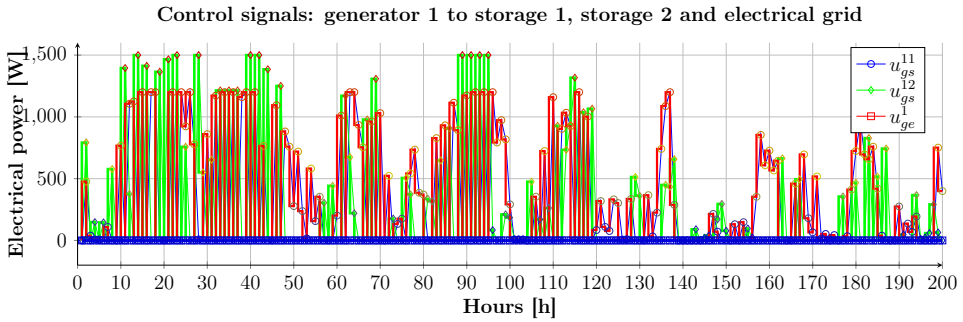


Figure 8.12.: Electrical power transmitted by the generator 1 to external grid and the two storage units.

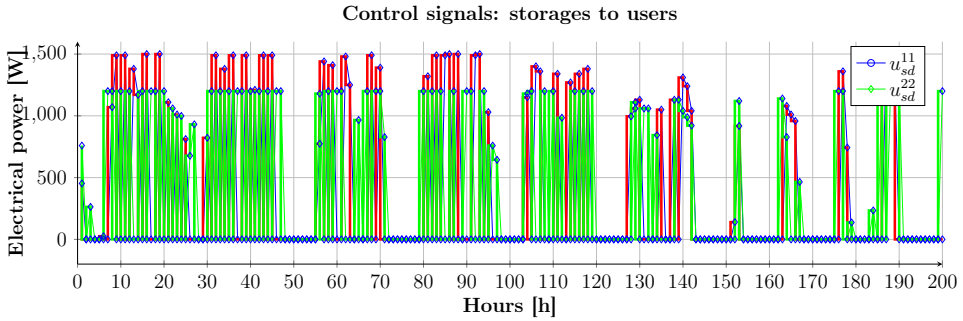


Figure 8.13.: Electrical power transmitted by the storage units to users.

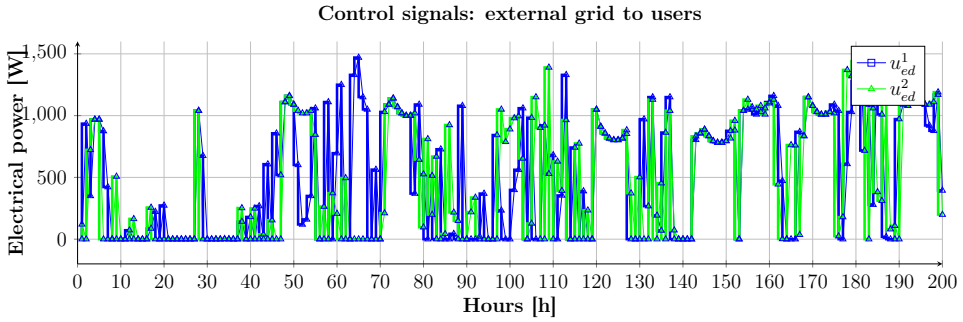


Figure 8.14.: Electrical power transmitted by the external grid to users.

profile (essential and total) of the first user against the actual received power. In the case of user 1 this means that the power flows from the 2nd generator, 1st storage unit and the external grid. The essential demand is ensured at all times and, conditions permitting, the total demand is also covered.

Furthermore, to clarify the usage of the generators we illustrate in Figure 8.16 the power flows of interest for the first generator. The generator profile is depicted as a tube around the real power output ($g_1(t)$ in solid black) and the actual power used in the migrogrid is shown as a dashed blue line (in the case of the first generator, the power output is partitioned between both storage units and the external grid). Lastly, in Figure 8.17 we show the minimal capacity of the storage units, B_{min} values computed as in Section 8.4.1 for a MTTR value of 5 hours and a total loss of power output in the first generator ($g_1(t) \to 0$). The first battery requires a large B_{min} value to cope with this fault scenario (it is getting in fact close to the $B_{max} = 6000$ W value) whereas the second battery requires no extra charge ($B_{min}^2 = 0$). These values are justified by the structure of the microgrid and by the fault's

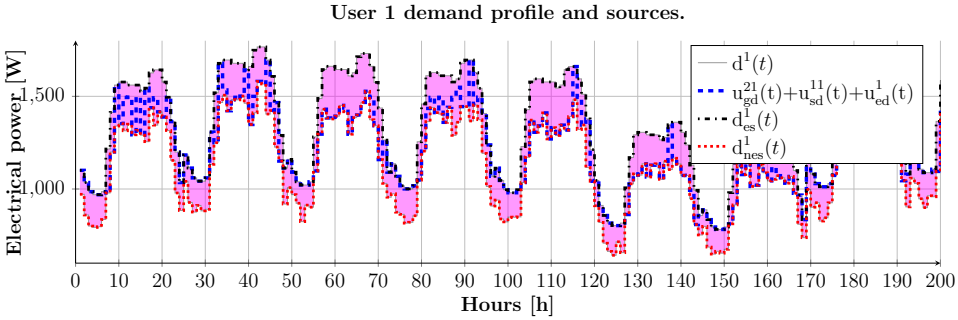


Figure 8.15.: Electrical power transmitted by three sources to user 1.

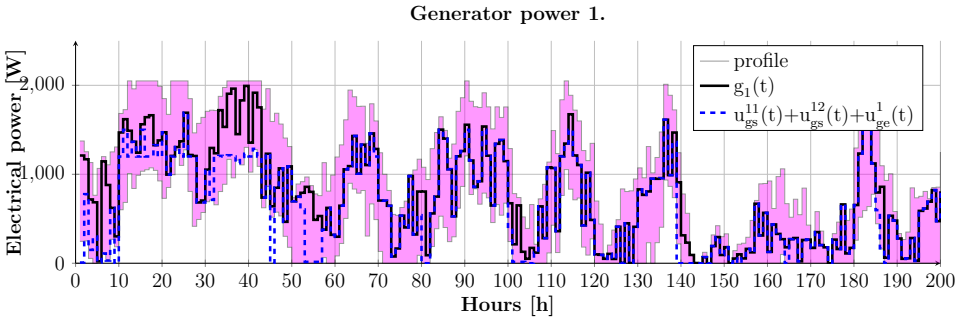


Figure 8.16.: Electrical power consumed by user 1.

nature: a fault in the first generator cuts all the sources of charge for the first storage unit whereas the second unit can still draw power from the second generator which continues to function correctly.

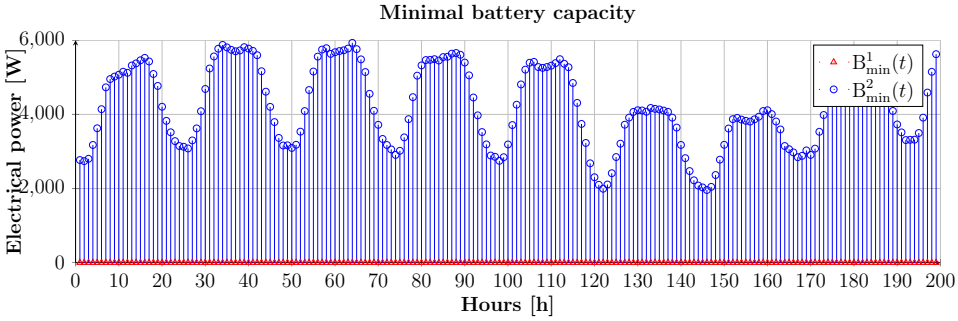


Figure 8.17.: Minimal capacity of the battery storage units.

8.6. Extension to robust MPC with shrinking prediction horizon

This section extends the MPC implementation presented above and in **J7**, uses the data provided in (Rahimiyan, Baringo, and Conejo 2014) and further applies robust approaches and deals with fault events. The goal is to propose a robust, optimization-based control implementation and reconfiguration for the reserve balancing in a microgrid system influenced by various types of uncertainties like wind and energy price variations or fault events.

The detailed model of the grid is given in **C58** and is similar with the one presented in Section 8.3. The particularity here resides in that the consumers have to meet a minimum daily energy consumption

$\boldsymbol{e}^{\text{day}}$ while, simultaneously, through an optimal scheduling, increase the grid utility. Thus, at the current time instant the sum of "already" and "to be"-consumed energies has to respect the constraint:

$$\sum_{h=1}^{t-1} \boldsymbol{e}_h^{\text{C}} + \sum_{h=0}^{24-t} \boldsymbol{e}_\tau^{\text{C}} \geq \boldsymbol{e}^{\text{day}}. \tag{8.51}$$

The bus injection depends on the wind, storage charge and discharge, main grid power, and the demand, where $\boldsymbol{\Omega}_{\text{B}}^{(\cdot)}$ are the adjacency matrices:

$$\boldsymbol{P}^{\text{B}} = \boldsymbol{\Omega}_{\text{B}}^{\text{W}} \boldsymbol{P}^{\text{W}} - \boldsymbol{\Omega}_{\text{B}}^{\text{C}} \boldsymbol{P}^{\text{C}} + \boldsymbol{\Omega}_{\text{B}}^{\text{ST}} (\boldsymbol{P}^{\text{ST},-} - \boldsymbol{P}^{\text{ST},+}) + \boldsymbol{\Omega}_{\text{B}}^{\text{S}} \boldsymbol{P}^{\text{S}}. \tag{8.52}$$

Each power demand is weighted by utility $\boldsymbol{u}^\top \in \mathbb{R}^{N_{\text{C}}}$, to value the worth of the consumption in comparison to the energy price.

8.6.1. Energy Management System (EMS)

The EMS solves, at each step, a constrained optimization problem with an economic objective function, which minimizes the cost of wind λ^{W} and external grid energy, while maximizing the utility $\boldsymbol{u}$.

$$J_t = \lambda_t^{\text{S}} e_t^{\text{S}} - \boldsymbol{u}_t^{\text{T}} \boldsymbol{e}_t^{\text{C}} + \lambda^{\text{W}} e_t^{\text{W}} + \sum_{h=1}^{24-t} \lambda_{t+h}^{\text{S}} e_{t+h}^{\text{S}} - \boldsymbol{u}_{t+h}^{\text{T}} \boldsymbol{e}_{t+h}^{\text{C}} + \lambda^{\text{W}} e_{t+h}^{\text{W}}. \tag{8.53}$$

The whole optimization problem is given by

$$\begin{aligned}
\min \quad & J_t \quad (8.53) && (8.54a)\\
\text{s.t.} \quad & \text{power balancing conditions} \quad (\textbf{C58}), (8.52) && (8.54b)\\
& \text{consumer constraints} \quad (8.51), \textbf{C58}) && (8.54c)\\
& \text{storage unit dynamics, storage constraints, trapezoidal rule} \quad (\textbf{C58}) && (8.54d)\\
& \text{main grid constraints, power lines constraints} \quad (\textbf{C58}) && (8.54e)
\end{aligned}$$

Using the utility as a time-varying weight, the value describes the worth of a particular consumption. Note that the objective function (8.53) utilizes a shrinking prediction horizon. For the current time step t, the objective function predicts $24-t$ steps into future (covers the remainder of the day). The constraints include the microgrid model with $\tau = t+h$, $h = 0, ..., 24-t$ and $t = 1, ..., 24$. The minimum generator power $\boldsymbol{P}^{\text{W,min}}$ appears for completeness reasons but in the scheme is set to zero.

In general, the EMS is not robust under forecast disturbances. Therefore a robust control approach is required. We propose two different control approaches: the robust EMS (8.55) taken from (Rahimiyan, Baringo, and Conejo 2014) and the minimax EMS (8.56), with the same cost J_t as in (8.53).

The robust EMS allows to set the degree of robustness by the parameters Γ^{S} and Γ^{W} (for further

information see (Rahimiyan, Baringo, and Conejo 2014)):

$$\min \quad J_t + \beta^{\mathrm{S}}\Gamma^{\mathrm{S}} + \sum_{h=1}^{24-t} \xi^{\mathrm{S}}_{t+h} \tag{8.55a}$$

$$\text{s.t.} \quad \text{constraints} \quad (8.54) \tag{8.55b}$$

$$\beta^{\mathrm{S}} + \xi^{\mathrm{S}}_{t+h} \geq (\lambda^{\mathrm{S,max}}_{t+h} - \lambda^{\mathrm{S,min}}_{t+h}) y^{\mathrm{S}}_{t+h} \tag{8.55c}$$

$$-y^{\mathrm{S}}_{t+h} \leq e^{\mathrm{S}}_{t+h} \leq y^{\mathrm{S}}_{t+h} \tag{8.55d}$$

$$P^{\mathrm{W}}_{t+h+1} - \frac{P^{\mathrm{AW,max}}_{t+h+1} + P^{\mathrm{AW,min}}_{t+h+1}}{2} + \beta^{\mathrm{W}}_{t+h+1}\Gamma^{\mathrm{W}}_{t+h+1} + \xi^{\mathrm{W}}_{t+h+1} \leq 0 \tag{8.55e}$$

$$\beta^{\mathrm{W}}_{t+h+1} + \xi^{\mathrm{W}}_{t+h+1} \geq \frac{P^{\mathrm{AW,max}}_{t+h+1} - P^{\mathrm{AW,min}}_{t+h+1}}{2} y^{\mathrm{W}}_{t+h+1} \tag{8.55f}$$

$$1 \leq y^{\mathrm{W}}_{t+h+1} \tag{8.55g}$$

$$P^{\mathrm{W}}_{t+1} \leq P^{\mathrm{AW}}_{t+1} \tag{8.55h}$$

$$\beta^{\mathrm{W}}_{t+h+1}, y^{\mathrm{W}}_{t+h+1}, \xi^{\mathrm{W}}_{t+h+1}, \xi^{\mathrm{S}}_{t+h}, y^{\mathrm{S}}_{t+h}, \beta^{\mathrm{S}} \geq 0. \tag{8.55i}$$

The minimax EMS considers the worst-case scenario by using the minimax MPC algorithm (Löfberg 2003), where the available wind power and the energy price are parametrized by the minimum and maximum forecast bound under the bounded uncertainty variables w.

$$\min \max_{w^{\mathrm{S}}, w^{\mathrm{AW}}} \quad J_t \tag{8.56a}$$

$$\text{s.t.} \quad \text{constraints} \quad (8.54) \tag{8.56b}$$

$$P^{\mathrm{AW}}_{t+h+1} = \frac{P^{\mathrm{AW,max}}_{t+h+1} + P^{\mathrm{AW,min}}_{t+h+1}}{2} + w^{\mathrm{AW}} \frac{P^{\mathrm{AW,max}}_{t+h+1} - P^{\mathrm{AW,min}}_{t+h+1}}{2} \tag{8.56c}$$

$$-1 \leq w^{\mathrm{AW}}_{t+h+1} \leq 1 \tag{8.56d}$$

$$\lambda^{\mathrm{S}}_{t+h+1} = \frac{\lambda^{\mathrm{S,max}}_{t+h+1} + \lambda^{\mathrm{S,min}}_{t+h+1}}{2} + w^{\mathrm{S}} \frac{\lambda^{\mathrm{S,max}}_{t+h+1} - \lambda^{\mathrm{S,min}}_{t+h+1}}{2} \tag{8.56e}$$

$$-1 \leq w^{\mathrm{S}}_{t+h+1} \leq 1 \tag{8.56f}$$

$$P^{\mathrm{W}}_{t+1} \leq P^{\mathrm{AW}}_{t+1}. \tag{8.56g}$$

Due to the good performance results of the minimax EMS, the next considerations are based on the minimax EMS implementation. Note that, when considering faults, or operating in islanded mode, the minimax EMS (8.56) computes no preemptive power and demand scheduling. To have reliability and feasibility, we analyze two preemptive schemes:

1. Soft constraints for power balancing (through the slack variables $\theta_{3,(\cdot)}, \theta_{4,(\cdot)}$), where the EMS is enforced to meet the demand by the wind and storage power.

$$\mathcal{O}^1 : \min \quad J_t + \sum_{h=0}^{24-t} Q_3\theta_{3,t+h+1} + Q_4\theta_{4,t+h+1} \tag{8.57a}$$

$$\text{s.t.} \quad \text{constraints} \quad (8.56) \tag{8.57b}$$

$$\sum_{i=1}^{N_{\mathrm{C}}} P^{\mathrm{C},i}_{t+h+1} \leq P^{\mathrm{ST},+}_{t+h+1} + P^{\mathrm{W}}_{t+h+1} + \theta_{3,t+h+1} \tag{8.57c}$$

$$P^{\mathrm{ST},-}_{t+h+1} \leq P^{\mathrm{W}}_{t+h+1} + \theta_{4,t+h+1} \tag{8.57d}$$

$$0 \leq \theta_{3,t+h+1}, \theta_{4,t+h+1}; \tag{8.57e}$$

2. Soft constraints for demand scheduling (through the slack variable $\theta^C_{(\cdot)}$), where the main grid power is considered as a bounded disturbance. Soft constraints are required to be feasible regarding the worst-case scenarios.

$$\mathcal{O}^2 : \min \max_{w^{\mathrm{PS}}} \quad J_t + \sum_{h=0}^{24-t} Q^{\mathrm{C}} \theta^{\mathrm{C}}_{t+h+1} \tag{8.58a}$$

$$\text{s.t.} \quad \text{constraints} \quad (8.56) \tag{8.58b}$$

$$P^{\mathrm{S,unc}}_{t+h+1} = \frac{P^{\mathrm{S,max}}_{t+h+1} + P^{\mathrm{S,zero}}_{t+h+1}}{2} + w^{\mathrm{PS}} \frac{P^{\mathrm{S,max}}_{t+h+1} - P^{\mathrm{S,zero}}_{t+h+1}}{2} \tag{8.58c}$$

$$P^{\mathrm{S,min}}_{t+h+1} \leq P^{\mathrm{S}}_{t+h+1} \leq P^{\mathrm{S,unc}}_{t+h+1} \tag{8.58d}$$

$$-1 \leq w^{\mathrm{PS}}_{t+h+1} \leq 1 \tag{8.58e}$$

$$\boldsymbol{P}^{\mathrm{C,min}}_{t+h+1} \leq \boldsymbol{P}^{\mathrm{C}}_{t+h+1} + \boldsymbol{\theta}^{\mathrm{C}}_{t+h+1} \leq \boldsymbol{P}^{\mathrm{C,max}}_{t+h+1} \tag{8.58f}$$

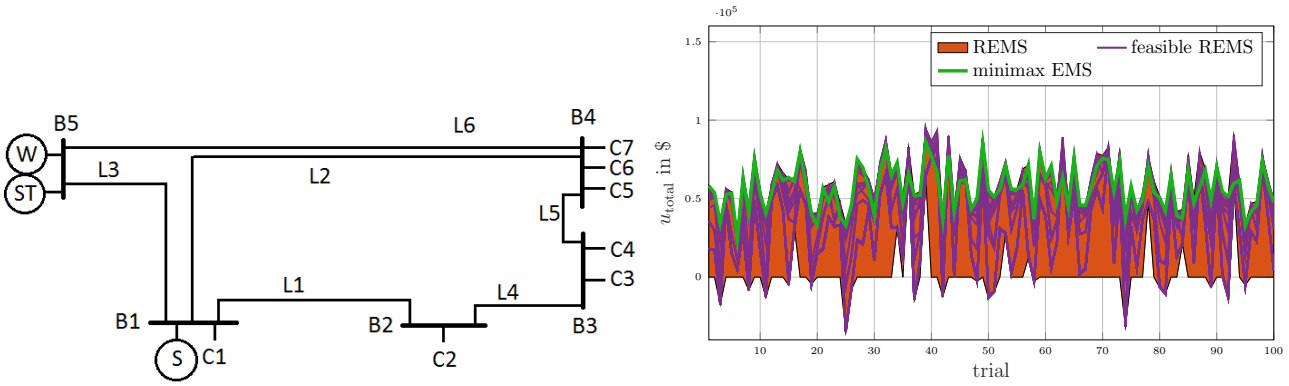
$$0 \leq \boldsymbol{\theta}^{\mathrm{C}}_{t+h+1} \tag{8.58g}$$

$$0 \leq \boldsymbol{P}^{\mathrm{C}}_{t+h+1} \tag{8.58h}$$

8.6.2. Simulations for various case studies

Performance over one day

For the undergoing case studies we use the 5-bus system in Figure 8.18. The wind power, energy price, utility and minimum demand profiles are taken from (Rahimiyan, Baringo, and Conejo 2014).



(a) 5-bus microgrid architecture

(b) Total utility of minimax MPC and robust EMS

Figure 8.18.: 5-bus microgrid architecture and total utility of minimax MPC and robust EMS.

$$J = \sum_{t=1}^{24} \lambda^{\mathrm{S}}_t e^{\mathrm{S}}_t - \boldsymbol{u}^{\mathrm{T}}_t \boldsymbol{e}^{\mathrm{C}}_t + \lambda^{\mathrm{W}} e^{\mathrm{W}}_t \tag{8.59}$$

Regarding the forecast disturbances, we simulate the performance of the robust EMS (REMS) and the minimax EMS, where we apply randomly the minimum or maximum forecast bound for the next time step in 100 scenarios. To analyze both approaches we use the total utility (8.59). The results in Fig. 8.18b indicate that the robust EMS has a performance range which is strongly parameter-dependent (e.g., there are parameter combinations that cause infeasibility, and parameter

combinations that are feasible for every scenario). Depending on the scenario, the robust EMS can lead to a better performance, than the minimax EMS. The latter is feasible in every scenario and leads to a good performance independent of the parameter choice. This is due to the good performance under disturbances that the schemes (8.57) and (8.58) have for the minimax EMS.

Considering line, storage, wind faults and the islanded mode, the overall goal is to charge preemptively the storage system, such that the necessary demand is covered. For the implementation, we use:

1. The "healthy-mode" controller (which uses internally, for prediction, the nominal grid model). This corresponds in fact to the results presented and discussed above (Figure 8.18b and the surrounding text).
2. The "faulty-mode" controller (the grid model is changed to characterize a fault[11]). This controller becomes active whenever a fault is detected (we assume no missed faults or false alarms). Once the grid is again under nominal functioning, the EMS switches back to the healthy-mode controller.

We are interested in the advantages and shortcomings of preemptive control schemes. These strongly depend on fault type, frequency of occurrence and operational constraints (e.g., line faults do not require a charged storage system; storage and wind faults depend on the external grid power variation constraints whenever, e.g., the EMS is bound to sell energy to the external grid).

Since each fault type is solved by specific and optimized architecture design, we hereinafter consider a limited test case: we focus on the islanded mode and consider temporary and unexpected lack of wind power as the fault. In general, the minimax EMS does not schedule the demands and charge the storage preemptively, thus we compare two preemptive schemes in Figure 8.19 under nominal conditions. The soft constrained power balancing scheme $\mathcal{O}^1$ causes a high battery usage, such that the storage charges and discharges frequently. In comparison to the minimax capacity profile $e^{\mathrm{ST}}_{\mathrm{nom}}$, the preemptive scheme charges earlier, while in this example the battery remains longer at the maximum capacity for the original minimax EMS. Hence the EMS schedules the demands preemptively. Scheme $\mathcal{O}^2$ charges earlier than the nominal case and remains close to being fully charged until the economic cost forces the discharge. In Tab. 8.3 Scheme $\mathcal{O}^1$ has a worse total utility (8.60) and profit (8.63), since the higher battery usage (8.62) is economically expensive.

$$u_{\mathrm{total}} = -\sum_{t=0}^{24}[\lambda^{\mathrm{S}}_t e^{\mathrm{S}}_t + \lambda^{\mathrm{W}} e^{W}_t - \boldsymbol{u}^{\mathrm{T}}_t \boldsymbol{e}^{\mathrm{C}}_t] \tag{8.60}$$

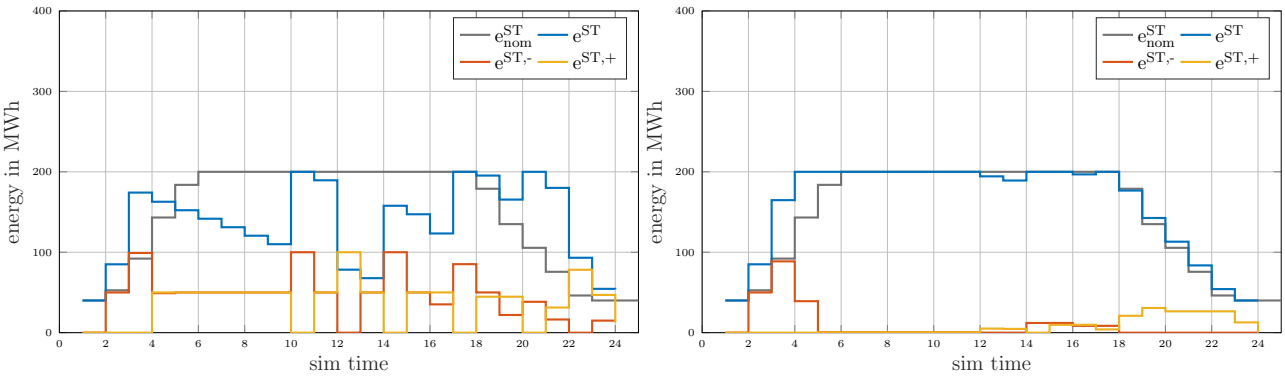
$$e_{\mathrm{total}} = \sum_{t=1}^{24}\sum_{k=1}^{N_{\mathrm{C}}} e^{\mathrm{C}}_{k,t} \tag{8.61}$$

$$V_0 = \frac{\sum_{t=1}^{24} e^{\mathrm{ST},-}_t}{e_{\mathrm{total}}} \tag{8.62}$$

$$V_1 = \frac{u_{\mathrm{total}}}{e_{\mathrm{total}}} \tag{8.63}$$

Under unexpected islanded mode the minimax EMS predicts at hour 7 no available wind power for the next 2 hours (worst-case-scenario). Therefore a higher storage energy is required due to variation constraints on the main grid power. Scheme $\mathcal{O}^1$ is infeasible for this scenario, because EMS follows economic objectives which lead to stored energy insufficient in covering the demands. Scheme $\mathcal{O}^2$ is feasible and chooses an optimal demand schedule under fault. E.g. for consumer 7 in Figure 8.20a the demand is lower before the fault appears and still receives the minimum demand under fault.

[11] For line, storage and wind faults the fault duration is unknown, so the fault is considered until the end of the day. For the islanded mode, we use a fixed fault duration of one hour.



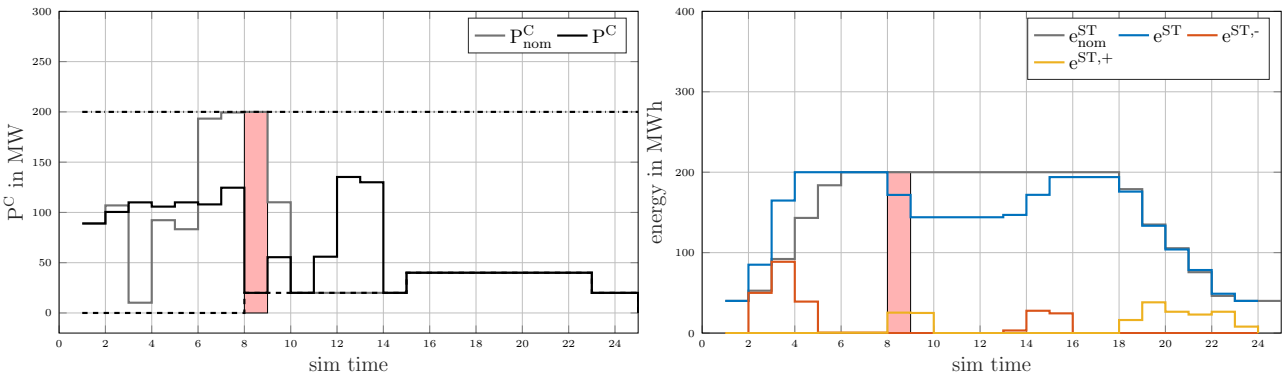
(a) Scheme $\mathcal{O}^1$.

(b) Scheme $\mathcal{O}^2$.

Figure 8.19.: Nominal storage profile.

Table 8.3.: 5-bus-system: nominal Performance of preemptive schemes for grid fault

Approach	u_{total} [\$ $\cdot 10^4$]	V_0	V_1 [$\frac{\$}{\text{MWh}}$]
minimax EMS	3.92	0.035	7.66
$\mathcal{O}^1$	2.08	0.214	4.14
$\mathcal{O}^2$	3.23	0.042	6.11



(a) Consumer 7 under fault

(b) Storage Energy under fault

Figure 8.20.: Analysis of the under-fault case.

In general, we cannot a priori provide guarantees about the feasibility of Schemes $\mathcal{O}^1$ or $\mathcal{O}^2$. Scheme $\mathcal{O}^1$ has a high battery usage, which leads to a worse overall cost. Scheme $\mathcal{O}^2$ is conservative, since it considers the worst-case scenario. The economic objectives and economic soft constraints can lead to infeasible steady-states. Thus, we consider these schemes (and similar variations) as tools to be used in Monte Carlo-like analysis: multiple faults, profiles and parameters variations are considered in order to assess which scheme, and under which circumstances, is better-suited to a particular case.

8.7. Discussions and conclusions

This chapter presented a reliable model predictive control approach for efficient energy storage and distribution in a microgrid system. Electricity costs, power consumption, generation profiles and power

Table 8.4.: 5-bus-system: performance under islanded mode.

Approach	u_{total} [\$ $\cdot 10^4$]	V_0	V_1 [$\frac{\$}{\text{MWh}}$]
Nominal	3.23	0.042	6.11
Unknown fault	2.95	0.046	5.72
Known fault	2.95	0.046	5.71

and energy constraints were taken into consideration in a Mixed-Integer Linear Programming optimization problem. Uncertainty due to variations in the environment (wind speed, load and electricity market price), failures and subsequent repairs of the generators as well as battery lifespan limitations are efficiently treated. Simulation results validate the proposed approach. The original contributions are:

- by minimizing the charge/discharge cycles the battery life is increased: this is achieved through the analysis of the expected battery life/versus electricity prices to find the optimum balance between battery usage and profit gained from electricity management;
- by using mixed-integer techniques, we can model different situations and behaviors of the microgrid components (switching between functioning modes of the storage elements and the microgrid in general, connected and islanded);
- fault tolerant strategies are proposed in order to ensure the necessary amount of energy in the storage devices such that (together with the utility grid) the essential consumer demand is always covered;
- by applying these improvements, we obtain a Mixed-Integer Linear Programming (MILP) optimization problem, which is difficult to solve but still with acceptable computational efforts.

For future work, we envision to analyze interconnected microgrid systems and discuss the combination between multi-agent approaches, mixed-integer programming and centralized/distributed/decentralized algorithms for scheduling control (Alraddadi, Conejo, and Lima 2020; Ananduta, Maestre, Ocampo-Martinez, and Ishii 2020; Bordons, Garcia-Torres, and Ridao 2020; Baros and Annaswamy 2019; Samad and Annaswamy 2017).

Part III.

Open issues and future directions

9. Spline representations for nonlinear formulations

Within my research, a recurring idea is to represent continuous and nonlinear signals in a finite spline space for further manipulation. In my collaborations, our usual choice are B-spline functions (Lyche, Manni, and Speleers 2018) due to their many nice properties (among which are local support, convexity and positivity). While powerful, this approach does have certain limitations:

i) non-linear constraints lead to non-convex feasible domains and numerically challenging optimization problems (especially when including binary variables in the mix as it leads to MINLP formulations);

ii) using B-splines leads usually to sufficient conditions based on their control points; this becomes conservative when handling nonlinear terms (the real curve lies far from its control points);

iii) from a practical viewpoint, it is troubling that the complexity of the representation depends more on the particularities of the problem than on parameters given at the initialization stage (e.g., trajectory planning, constrained profile generation in microgrid systems).

Not least, in my admittedly limited experience in the area, it feels that B-splines and spline theory in general are under-used or at least, used at a basic level, in the control community. Hence, I strongly believe that existing theoretical results from spline and approximation theory can be successfully grafted and particularized for the control topics in which I am interested.

I propose hence several directions of study. To illustrate these, consider a typical nonlinear system (the 2 degree of freedom dynamics of a fixed-wing aircraft):

$$\dot{x}(t) = V_{\mathrm{a}}(t)\cos\psi(t), \tag{9.1a}$$
$$\dot{y}(t) = V_{\mathrm{a}}(t)\sin\psi(t), \tag{9.1b}$$
$$\dot{\psi}(t) = \frac{g\ \tan\phi(t)}{V_{\mathrm{a}}(t)}. \tag{9.1c}$$

The state variables are represented by the position $(x(t), y(t))$ and the heading (yaw) angle $\psi(t) \in [0, 2\pi]$ rad. The input signals are the airspeed velocity $V_{\mathrm{a}}(t)$ and the roll angle $\phi(t)$, respectively. The flat output is taken as the position components of the state,

$$z(t) = \begin{bmatrix} z_1(t) & z_2(t) \end{bmatrix}^\top = \begin{bmatrix} x(t) & y(t) \end{bmatrix}^\top \tag{9.2}$$

which allows to rewrite the remaining state ($\psi(t)$) and inputs ($\phi(t), V_a(t)$) in terms of itself and its derivatives:

$$\psi(t) = \arctan\left(\frac{\dot{z}_2(t)}{\dot{z}_1(t)}\right), \qquad V_a(t) = \sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}, \tag{9.3a}$$
$$\phi(t) = \arctan\left(\frac{1}{g}\frac{\ddot{z}_2(t)\dot{z}_1(t) - \dot{z}_2(t)\ddot{z}_1(t)}{\sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}}\right). \tag{9.3b}$$

Further, assume that $z(t)$ is given as a B-spline curve of degree d and control points $\mathbf{P} = \{P_1, P_2, \ldots\}$:

$$z(t) = \sum_{i=1}^{n} P_i B_{i,d}(t). \tag{9.4}$$

9.1. Spline approximations

The goal is often to solve a constrained optimization problem where both the cost and the constraints are given in terms of the control points P. For example, rewriting $g\tan\phi(t)$ in (9.1) using (9.4):

$$g\tan\phi(t)=\frac{\ddot{z}_2(t)\dot{z}_1(t)-\dot{z}_2(t)\ddot{z}_1(t)}{\sqrt{\dot{z}_1^2(t)+\dot{z}_2^2(t)}}=\frac{\sum\limits_{i=1}^{n+2}\sum\limits_{j=1}^{n+1}\left(P''_{2,i}P'_{1,j}-P''_{1,i}P'_{2,j}\right)B_{i,d-2}(t)B_{j,d-1}(t)}{\sqrt{\sum\limits_{k=1}^{n+1}\sum\limits_{\ell=1}^{n+1}\left(P'_{1,k}P'_{1,\ell}+P'_{2,k}P'_{2,\ell}\right)B_{k,d-1}(t)B_{\ell,d-1}(t)}},\tag{9.5a}$$

with P', P'' obtained from the B-spline differentiation property, allows to constrain $g\tan\phi(t)$ in terms of the control points P. Since both the numerator and denominator are convex sums (the sets of functions $\{B_{i,d-2}B_{j,d-1}\}$ and $\{B_{k,d-1}B_{\ell,d-1}\}$ partition the unity), a magnitude condition such as:

$$g\tan\underline{\phi}\leq g\tan\phi(t)\leq g\tan\overline{\phi}\tag{9.6}$$

may be written in terms of its control points. Exploiting the convexity property we have that the numerator and nominator lie in the convex hull defined by the control points $\left\{P''_{2,i}P'_{1,j}-P''_{1,i}P'_{2,j}\right\}_{i,j}$ and, respectively, $\left\{P'_{1,k}P'_{1,\ell}+P'_{2,k}P'_{2,\ell}\right\}_{k,\ell}$. Hence, a sufficient condition for (9.6) is:

$$g\tan\underline{\phi}\leq\frac{P''_{2,i}P'_{1,j}-P''_{1,i}P'_{2,j}}{\sqrt{P'_{1,k}P'_{1,\ell}+P'_{2,k}P'_{2,\ell}}}\leq g\tan\overline{\phi},\quad\forall i,j,k,\ell.\tag{9.7}$$

While (9.7) can be introduced as a nonlinear constraint in any suitable solver, it is cumbersome and it may rightfully be asked whether the added complexity is actually worthwhile. An alluring approach is the use of spline approximations Lyche, Manni, and Speleers 2018. These tools are mature and have strong theoretical foundations. There is plethora of options but my first choice is the Schoenberg operator Beutel, Gonska, Kacsó, and Tachev 2002 as a quasi-interpolant due to its properties: i) it has a simple construction, based on the Greville points $\xi^\star_{i,d,\xi}$ of the knot used by the approximant spline basis; ii) it uses the ℓ_∞ norm to bound the approximation error; and iii) the complexity of the representation is controlled through the knot vector used by the approximant. Applying the Schoenberg operator for the scalar function $f(t)=g\tan\phi(t)$ we obtain its approximant in the spline space defined by $\{\tilde{B}_{i,d}(t)\}$:

$$\tilde{f}(t)=\sum_{i=1}^{n}f(\xi^\star_{i,d,\xi})\tilde{B}_{i,d}(t)\tag{9.8}$$

A sufficient condition to verify (9.6) is that $g\tan\underline{\phi}\leq f(\xi^*_{i,d,\xi})\leq g\tan\overline{\phi}$, $\forall i=1\dots n$ holds; the control points $f(\xi^*_{i,d,\xi})$ are still nonlinear combinations of the original control points P, as in (9.7), but there are many fewer constraints to be considered (n instead of the $n^2\cdot(n+1)^2$ appearing in (9.7)). Several elements are of interest for future research (either as positive outcomes or as downsides to be tackled):

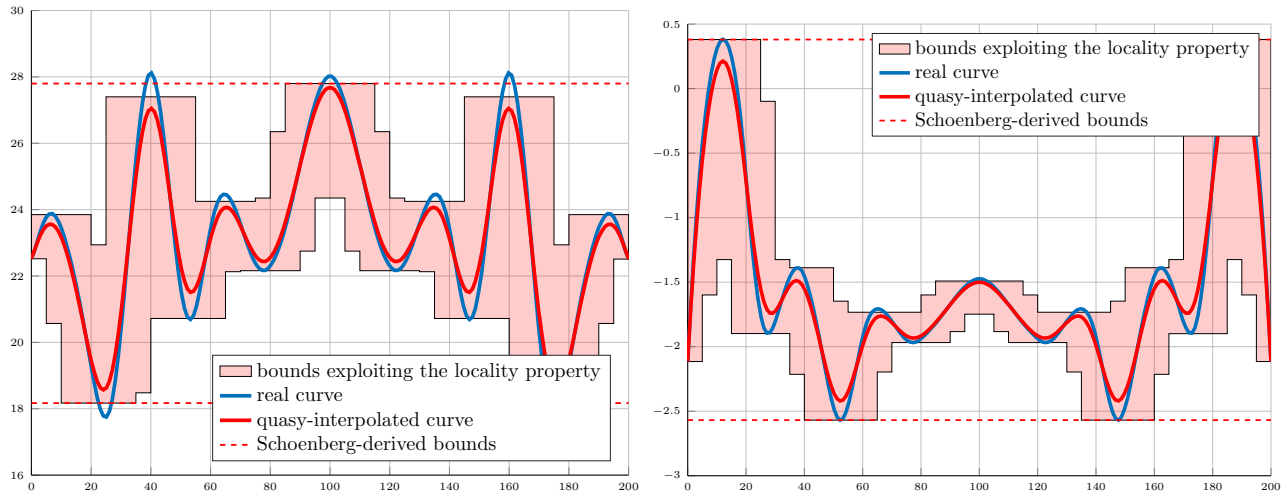
i) $\tilde{f}(t)$ is a quasi-interpolant and cannot describe exactly $f(t)$; still, the interpolation error $\|f(t)-\tilde{f}(t)\|_\infty$ is bounded by $\|D^2f\|^2_\infty$, a quantity related to the 2nd order derivative of $f(t)$.

ii) The performance of the approximation depends, quantifiably so, on the knot mesh size (if the size of the sub-intervals is small, the approximation is better).

iii) The approximation error bound depends on the degree of the B-spline basis. Relatively recent results provide bounds which exploit the second order smoothness modulus and apparently provide tighter bounds Zapryanova and Tachev 2012.

iv) While any nonlinear function can be approximated through a quasi-interpolant, I expect that when the function is a nonlinear combination of B-splines, the approximation bounds are sharper.

Figure 9.1 illustrates my initial research in this area. Applying the Schoenberg operator as in (9.8) for the velocity and roll angle, defined as in (9.3) I obtain tight curves (close to the bounds induced by their control points and close wrt to the curves computed with the exact forms). Two aspects are noteworthy: the actual curves may pass the bounds (since these were computed for the approximant) and the bounds can be computed for each knot sub-interval thus tightening the sufficient conditions.



(a) $V_a(t)$ (b) $\phi(t)$

Figure 9.1.: Approximations based on the Schoenberg operator for the control inputs.

9.2. Knot refinement techniques

Arguably, the main appeal of the B-spline parametrization is that we can use their control points to provide sufficient conditions for the curve that they characterize.

The issue is that the sufficient conditions may be conservative. The naive idea, to simply increase the number of control points is not necessarily adequate. Any increase in control points is in fact an increase in basis functions and a corresponding increase in complexity. Approaches such as the LMI-based formulation from Section 3.3.2 which gives necessary and sufficient conditions are complex both theoretically and implementation-wise.

I propose to use the knot refinement technique Piegl and Tiller 2012 to obtain tighter approximations of the B-spline curve. The catch is that I do not introduce new control points but rather 'fake' control points which depend linearly on the initial ones (the true variables of the optimization problem). Thus, the number of control variables is decoupled from the precision of the sufficient constraints.

Let us consider a time instant σ with $1 \leq k < n+1$ taken such that $t_k < \sigma < t_{k+1}$ (or, equivalently stated, $[t_k, t_{k+1}) = [t_k, \sigma) \cup [\sigma, t_{k+1})$). This allows to define a *refined* knot-vector $\overline{\boldsymbol{\zeta}} = \{\overline{\tau}_i\}_{i=1\ldots n+d+2}$:

$$\overline{\tau}_i = \begin{cases} \tau_i, & 1 \leq i \leq k \\ \sigma_i, & i = k+1 \\ \tau_{i-1}, & k+2 \leq i \leq n+d+2 \end{cases}, \tag{9.9}$$

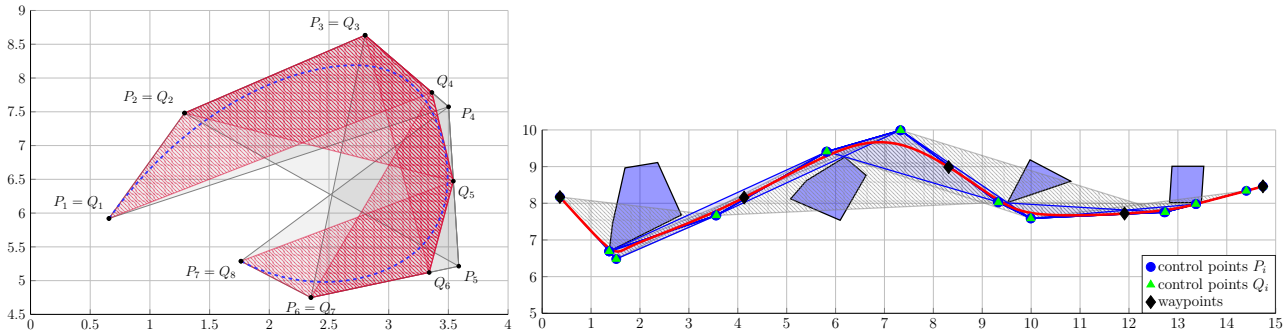
which permits to redefine the initial curve $z(t)$ constructed as in (9.13) in the novel basis $\{\overline{B}_{\ell,d,\overline{\boldsymbol{\zeta}}}(t)\}_{\ell=1\ldots n+1}$:

$$z(t) = \sum_{i=1}^{n} P_i B_{i,d,\boldsymbol{\zeta}}(t) = \sum_{i=1}^{n+1} Q_i \overline{B}_{i,d,\overline{\boldsymbol{\zeta}}}(t), \ \forall t \in [\overline{\tau}_{d+1}, \overline{\tau}_{n+2}], \tag{9.10}$$

where Piegl and Tiller 2012, p. 142,

$$Q_i = \alpha_i P_i + (1-\alpha_i)P_{i-1}, \quad \alpha_i = \begin{cases} 1, & i \leq k-d \\ \frac{\sigma-\tau_i}{\tau_{i+d}-\tau_i}, & k-d+1 \leq i \leq k \\ 0, & i \geq k+1. \end{cases} \tag{9.11a}$$

The idea, as illustrated in Figure 9.2 is straightforward. The curve is guaranteed to lie in a union of convex regions characterized by consecutive control points. By increasing the number of control points through a knot insertion/refinement procedure we arrive to increasingly accurate approximations of the curve. In fact, at the limit, the control polygon will converge towards the curve Boor 1987. Not in the least, the difference between the curve and its control polygon has a known bound Lutterkort and Peters 1999. The optimization problem involving these points keeps the same number of control points (since the new ones are linear combinations of them) and only the number of constraints increases (more regions in which the curve lies).



(a) knot insertion

(b) obstacle avoidance in a cluttered environment

Figure 9.2.: Illustration of motion planning in a cluttered environment - infeasibility of the initial implementation.

9.3. Extensions of the standard B-spline family

While the discussion until now was centered on standard B-splines, it does not necessarily have to be so. There are many variants and extensions which may be employed.

Particular knot vectors

Taking particular knot vectors we achieve basis families which exhibit specific properties. Taking for example the knot vector as the set of integers ($\tau_j = j$) we arrive at *cardinal B-spline functions* Milovanović and Udovičić 2010 which enjoy simplified properties since all the basis functions are offsets of each other (i.e., they have the same shape but start at different times along the time axis).

Going the other direction, are the non-uniform knot meshes. The requirement for such an implementation arises naturally when considering a heterogeneous environment. Take the motion planning problem: when the curve is passing through a convoluted region it is desirable to have many degrees of freedom but keeping the same density of splines in regions without obstacles is wasteful from a computational viewpoint. Hence, implementations such as truncated and/or hierarchical B-spline functions have, in my opinion, great potential Speleers 2017; Giannelli, Jüttler, and Speleers 2012.

Extension to NURBS

Non-uniform B-splines (NURBS) are a well-known extension of B-splines Piegl and Tiller 2012. NURBS of degree d are obtained by weighting a B-spline basis with positive scalars $\{\omega_1, \dots, \omega_n\} \geq 0$:

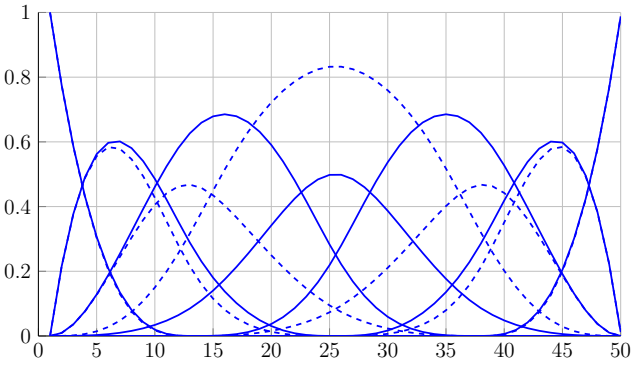
$$R_{\ell,d,\boldsymbol{\zeta}}(t) = \frac{B_{\ell,d,\boldsymbol{\zeta}}(t)\omega_\ell}{\sum_{i=1}^{n} B_{i,d,\boldsymbol{\zeta}}(t)\omega_i}, \ \forall \ell = 1, \dots, n. \tag{9.12}$$

Lastly, taking the control points $\{P_1, \dots, P_n\} \subset \mathbb{R}^p$ gives the NURBS curve

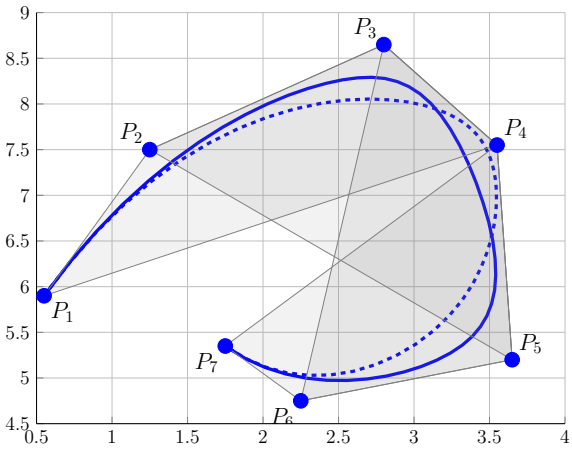
$$z(t) = \sum_{\ell=1}^{n} P_\ell R_{\ell,d,\boldsymbol{\zeta}}(t) = \frac{\sum_{\ell=1}^{n} B_{\ell,d,\boldsymbol{\zeta}}(t)\omega_\ell P_\ell}{\sum_{\ell=1}^{n} B_{\ell,d,\boldsymbol{\zeta}}(t)\omega_\ell}, \ \forall t \in [\tau_{d+1}, \tau_{n+1}]. \tag{9.13}$$

The one additional property wrt standard B-splines is their projective invariance property. The rational curve $z(t) \in \mathbb{R}^p$ may be represented in homogeneous coordinates as a polynomial curve $z^\omega(t) \in \mathbb{R}^{p+1}$. That is, to each pair of control point and weight $(P_i, \omega_i) \in \mathbb{R}^p \times \mathbb{R}_+$ corresponds a control point $P_i^\omega = \begin{bmatrix} \omega_i P_i^\top & \omega_i \end{bmatrix}^\top$ which characterizes the B-Spline curve $z^\omega(t) = \sum_{i=1}^{n} P_i^\omega B_{i,d,\boldsymbol{\zeta}}(t)$.

Hence, using NURBS is both beneficial due the additional degrees of freedom (the weights ω_i) and to the projective invariance property (any such transformation of a NURBS curve can be described through its projected control points) but it also has added complexities, e.g., in the the derivative description (the NURBS curve derivative is no longer itself a combination of NURBS). Figure 9.3a shows two basis functions defined by the same knot vector but having the different weights and Figure 9.3b shows the resulting curves which lie in the same regions (regardless of their weights).



(a) NURBS basis functions.



(b) NURBS curve and convex hulls.

Figure 9.3.: Illustration of NURBS functions and curve.

Different choices for the zero-order B-splines

In most definitions of B-splines, the recurrence which gives them starts with window function (differences of step functions). Strictly speaking, these starting functions may be chosen arbitrarily (as long as they are defined with a finite support). There are works which consider truncated trigonometric or truncated exponential functions (Christensen and Massopust 2012). It might be interesting to see whether some of these formulations have benefits for our profile generation problems.

10. Further benefits of flatness in control problems

Multi-physics systems are characterized by continuous-time dynamics, constraints and costs which more often than not operates at different time scales. Hence, flat-output representations where the system's states and inputs are written in terms of a flat output and its derivatives are increasingly popular (Fliess, Lévine, P. Martin, and Rouchon 1995), (Mounier and Rudolph 1998), (Levine 2009).

Having a flat representation is, in a sense, the equivalent of controllability for nonlinear dynamics. Furthermore, from the control viewpoint, expressing the input in terms of its flat output is a model inversion (the nonlinear input linearizes the nonlinear dynamics to arrive at the desired flat output).

i) While there are many formal results for flat-output representations, there is no (to my knowledge at least), a generic algorithm capable to obtain a flat representation for any of the nonlinear dynamics which admit one.

ii) Even when a flat representation is obtained, it may quickly become difficult to handle. I.e., usually the inputs have convoluted expressions in terms of the system's flat output which leads to difficult to handle constraint and cost formulations.

The aforementioned issues make clear that flat representations and the subsequent operations are challenging from both theoretical and practical viewpoints. Still, in my opinion, the positive aspects (such as having analytical descriptions of costs and constraints, avoiding the explicit differential equations linking the states and inputs) may greatly improve the state of the art in nonlinear control.

In what follows, I propose tackling some specific issues which, in my opinion, can have a significant impact. First, I propose to study whether the structure and properties of port-Hamiltonian systems can be used to provide a flat-based representation and, if so, whether there is a constructive algorithm which provides the flat output and the associated state/input representations. Further benefits of such properties in optimization based-control are worth to delve into. Next, as seen earlier in the manuscript, I am interested in power profile generation for electrical grids. I propose to use flatness to generate such profiles, with pre-defined complexity bounds, and which are optimal wrt power balance, demand satisfaction and resilient to unexpected behavior (e.g., line faults).

10.1. On flat representations for Port-Hamiltonian Systems

Port-Hamiltonian formulations (A. v. d. Schaft, Jeltsema, et al. 2014) provide a well-structured state-space model for electrical systems which still allows to extract physical quantities such as voltages, currents and powers. Furthermore, flatness is inherently suited for the hierarchical control of complex dynamics. Hence, building on the properties of these two classical notions and merging them is a worthwhile endeavor in the field of electrical grids (but not only).

In our recent works **J11**, **C42** we propose a flatness-based hierarchical control for power balancing in a meshed DC microgrid system. Firstly, through the PH formulation, the state-space representation of the microgrid system is described. Next, the differential flatness is employed to express the states and control inputs in function of the flat outputs, maintaining, at the same time, the properties of the PH model. We were able to generate optimal profiles for the current and voltage of the energy storage (ES) system while minimizing the electricity purchase from the external grid.

Through my work with, and initial guidance from Prof. Laurent Lefèvre, on the topic of port-Hamiltonian systems I consider that these preliminary results may be extended to answer:

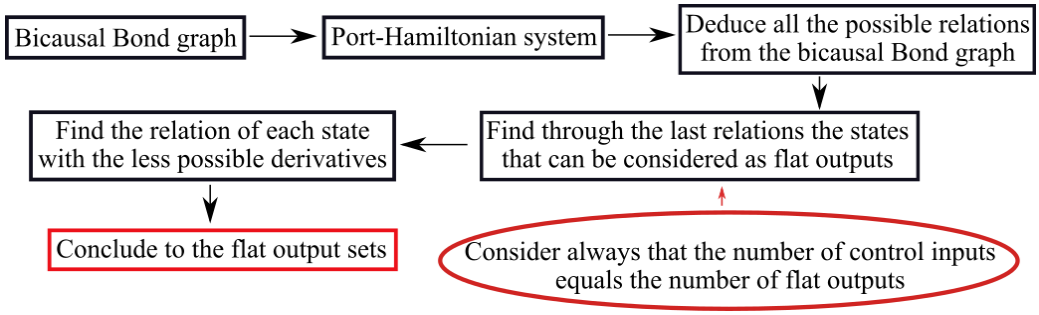


Figure 10.1.: From a pH model to a flat-ouput representation

i) Does any port-Hamiltonian system accept a flat-output representation? A preliminary result which employs the causality property of a Bond Graph encourages us to think that any PH system which is Bond Graph representable also has a flat representation **C55**.

ii) It is an open area whether the properties which make PH systems appealing (such as the modularity and physical interpretability) may be kept when translating the model through a flat representation. If these hold, I expect that effective optimization-based control strategies for multi-physical systems can be employed (e.g., in large-scale electrical grids).

I propose to build on the algorithm proposed in (Franke and Robenack 2013) which provides flat outputs for a series of examples of nonlinear physical systems (a satellite model, a rolling disc and a Brockett integrator). The algorithm provides an analytical computation of flat outputs for nonlinear control systems and it was implemented in **J11** to find a flat output set for the energy storage PH system.

Another method was introduced in (Gil, Pedraza, Delgado, and Sira-Ramirez 1997) and (Richard, Buisson, and Cormerais 2002), which proves the flatness of a system after inverting the dynamics of its associated Bond graph. This method is called bicausality, firstly presented by (P. Gawthrop 1994). Bicausality is a supplementary approach to the existing theory of the Bond graphs which inverses the system dynamics by decoupling the pairs of efforts and flows at each bond. Hence, bicausality could be an appropriate method to investigate flatness and, subsequently, develop a more straightforward way to find the possible flat outputs of the PH systems.

Let us sketch a possible method to find the appropriate sets of flat outputs for electrical circuit in port-Hamiltonian form in Figure 10.1:

1) exploit the bi-causality property of a Bond Graph;

2) obtain the PH formulation of the system;

3) deduce all the possible relations among the efforts and flows within the bicausal Bond graph;

4) find the states which might be considered as flat outputs;

5) find the relation of each state with the minimum number of derivatives;

6) deduce the flat output representation.

To illustrate the point, let us consider the model of an energy storage systems which includes a Split-Pi converter and a KiBaM battery. For the ES unit, we have four control inputs: $v_{DC}(t)$, $i_{R_{1b}}(t)$, $d_{1sc}(t)$, $d_{2sc}(t)$.

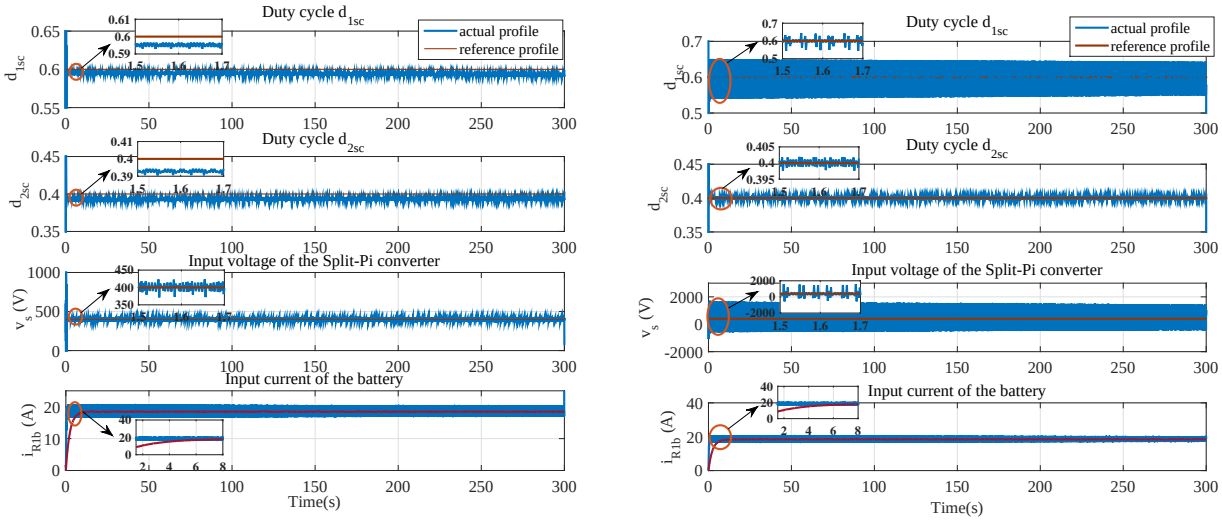
Using the algorithm provided in (Franke and Robenack 2013) we obtain the following set of flat outputs which subsequently leads to the re-formulated inputs:

$$\begin{cases} z_1(t) = & \frac{1}{I_{1sc}}\frac{p_{1sc}(t)^2}{2} + \frac{1}{I_{2sc}}\frac{p_{2sc}(t)^2}{2} + \frac{1}{C_{2sc}}\frac{q_{2sc}(t)^2}{2} \\ z_2(t) = & q_{3sc}(t) + q_{1b}(t) \\ z_3(t) = & q_{2b}(t) \\ z_4(t) = & q_{2sc}(t) \end{cases} \Rightarrow \begin{cases} d_{1sc} = & \Phi_8(z_1, z_2, \dot{z}_2, \ddot{z}_2, z_3, \dot{z}_3, \ddot{z}_3, z_4) \\ d_{2sc} = & \Phi_9(z_2, \ddot{z}_2, z_3, \dot{z}_3, \ddot{z}_3, z_4) \\ v_{DC} = & \Phi_{10}(z_1, \dot{z}_1, \ddot{z}_1, z_2, \dot{z}_2, \ddot{z}_2, \dddot{z}_2, \\ & \quad z_3, \dot{z}_3, \ddot{z}_3, \dddot{z}_3, z_4, \dot{z}_4, \ddot{z}_4) \\ i_{R_{1b}} = & \Phi_{11}(\dot{z}_3, \ddot{z}_3). \end{cases} \tag{10.1}$$

With the aforementioned algorithm, a simplified set of flat outputs is found:

$$\begin{cases} z_1(t) & = p_{1sc}(t) \\ z_2(t) & = q_{3sc}(t) \\ z_3(t) & = q_{2b}(t) \\ z_4(t) & = q_{2sc}(t) \end{cases} \Rightarrow \begin{cases} d_{1sc} & = \Phi_{19}(z_2, \dot{z}_2, \ddot{z}_2, \dot{z}_3, \ddot{z}_3, \dddot{z}_3, z_4, \dot{z}_4) \\ d_{2sc} & = \Phi_{20}(z_2, \ddot{z}_2, \ddot{z}_3, \dddot{z}_3, z_4) \\ v_{DC} & = \Phi_{21}(z_1, \dot{z}_1, \ddot{z}_1, z_2, \dot{z}_2, \ddot{z}_2, \dddot{z}_2, \dot{z}_3, \ddot{z}_3, \dddot{z}_3, \ddddot{z}_3, z_4, \dot{z}_4, \ddot{z}_4) \\ i_{R_{1b}} & = \Phi_{22}(\dot{z}_3, \ddot{z}_3) \end{cases} \tag{10.2}$$

with the inputs written in function of the flat outputs.



(a) Flat outputs set (10.1).

(b) Flat outputs set (10.2).

Figure 10.2.: Control inputs in function of two different sets of flat outputs. v_s, d_{1sc} and d_{1sc} are considered as constant in the reference PH system, while i_b varies.

In Fig. 10.2a and Fig. 10.2b, the four control inputs, $v_{DC}(t)$, $i_{R_{1b}}(t)$, $d_{1sc}(t)$, $d_{2sc}(t)$ of the ES are written in function of the two sets of flat outputs (10.1-10.2) and, after implementing the Split-Pi model in Matlab/Simulink both in PH form and in flat representation, the simulation results are obtained. Analyzing the two figures, Fig. 10.2a and Fig. 10.2b, we observe that with (10.1) we obtain stabler and less noise-affected results for v_s. Contrariwise, the i_b signal contains more noise-affected data. In the duty cycles, slight disturbances appear for both cases. In general, both sets of flat outputs, (10.1) and (10.2), can be considered for the flat representation of the Split-Pi converter. Recall that, for DC microgrids, the voltage regulation of the central transmission network is one of the principal factors to investigate. Here, the input voltage, v_s, to the Split-Pi converter is considered in the optimization problem to regulate the operation of the battery for cost minimization in the DC microgrid. In this problem, the flat representation of the ES system may be further combined with the B-spline parametrization.

It is important to note (and will be in fact subject of further study) that if a system admits flat outputs it has in fact an infinity of variations. Here we have shown two representations and we already can see the potential pitfalls. Most importantly, choosing a simple flat output may lead to convoluted state/input representations which then lead to difficult formulations of constraints and costs in an optimization control problem.

10.2. Generation of power profiles through flatness-based methods

DC microgrids (and other energy transport systems) are often controlled through a hierarchical scheme. One of the main reasons is the grid's temporal heterogeneity: a typical grid meshes together elements whose response time ranges from miliseconds (duty cycles in DC/DC converters), seconds (generator power outputs) and minutes and even hours (load dispatch decisions). Hence, it is common to have multiple layers which communicate top-down (the top layer provides a profile which has to be followed, within prescribed bounds, by the elements of the bottom layer).

The main issues stem from decisions about the discretization of continuous dynamics and the presence of nonlinearities (induced by the dynamics/control strategies proposed). Hence, I will consider flatness-based representation of power profiles in a DC microgrid, parametrized by a finite spline basis, which ensures reliable power balancing within a mixed-integer framework. This construction deals naturally with nonlinearities and avoids profile sampling altogether. In what follows I enumerate some directions of research, together with their expected difficulties:

i) Generation of power profiles through flatness-based methods. The nonlinear DC grid model is reformulated into a flat-output representation, thus expressing costs, constraints and profiles in a unitary manner, suitable for subsequent manipulation:
   - Use port-Hamiltonian model descriptions to automate the computation of the flat form, as described in the previous section and as tentatively discussed in **C55** or (Richard, Buisson, and Cormerais 2002).
   - Use a finite spline basis (B-splines or related spline families) to reformulate the continuous costs and constraints into a finite number of relations involving the splines' control points.

ii) Implementation of the power-balancing problem within a mixed-integer framework. To ensure user demand and power loss minimization in the face of intermittent generation and access to storage resources I consider a mixed-integer formulation:
   - Use the flat-output representation together with the spline parametrization to provide a mixed-integer description in terms of the splines' control points. A significant difficulty will be solving the resulting MINLPs (due to the nonlinearities of the dynamics and of the flat-output descriptions).
   - The knot vector defining a B-spline basis may be non-uniform. I plan to exploit this for modelling profiles with regions of quick variations (periods of peak functioning) such as to remain within a prescribed problem complexity (e.g., a limited number of binary variables controlling the unit commitment/load dispatch).

iii) Distributed optimization-based control design. Here I plan to integrate the aforementioned tools to provide new strategies for the robust and distributed control of the DC grid with stability guarantees.
   - Distributed control scheme for energy management. Owing to the different time scales, I will consider a multi-tier architecture. In the slow time scale, a coherent combination between feedback linearization and economic Model Predictive Control (eMPC), (Alamir and Pannochia 2020; Bordons, Garcia-Torres, and Ridao 2020; Cataldo, Bertolotti, and Scattolini 2019) handles power flow (capturing in the stage cost the price variations **J6**). In the fast time scale, passivity-based controllers will be used.

- Generation of flatness-based reference profiles. I will implement a flatness-based description Levine 2009 of the continuous fast dynamics in order to ensure bidirectional communication with the slow tier. From fast to slow: provide limitations (constraints, cost penalties) to be considered in the eMPC. From slow to fast: provide references to be tracked and which guarantee the desired power flow.

I plan to validate the above ideas mainly against the DC grid elevator architecture presented in Figure 10.3. The electro-mechanical elevator's variable load is covered by a combination of *fixed* (external grid, storage components, i.e., batteries and/or super capacitors) and *intermittent* (renewable energy sources – solar panels, plug-in vehicles - EVs) *sources*.

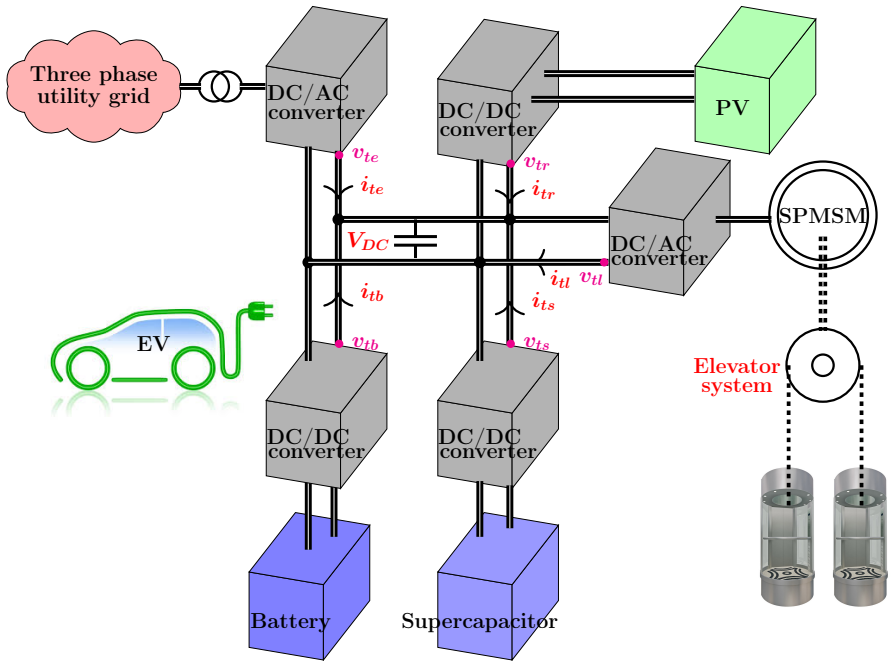


Figure 10.3.: DC grid benchmark.

The inter-play between the mechanical and electrical sub-systems of the elevator DC grid gives opportunities in power management. The elevator recovers energy through its PMSM (Permanent Magnet Synchronous Machine), thus, the energy produced when the elevator brakes may be stored instead of being evacuated through resistors. Solar panels can be added to the grid and plug-in vehicles may connect temporarily to the grid such that the energy stored ca be used to avoid buying from the utility grid during peak hours (i.e., peak shaving).

The goal is to ensure user satisfaction and grid reliability while simultaneously reducing initial (fixed storage capacity) and operating costs (power bought from the utility grid) with the aim of increasing self-reliance, up to and including full autonomy in the case of abnormal functioning (generator and/or utility grid failures). A distributed architecture with multiple elevators will be considered as an expansion.

The validation will be done in collaboration with SODIMAS which will provide real usage profiles (has already done so in the past in projects in which I was involved).

11. Data-driven alternatives to classical control optimization

Today's control problem are often coming from complex systems (e.g., large-scale systems with potentially heterogeneous components, strong couplings and/or poor observability). Keeping with my previous research activities, I already observed first-hand the difficulties and open topics appearing in electrical grid networks (which may have tens of thousands of nodes) and motion planning (where multiple drones have to interact and achieve consensus in a cluttered and changing environment). Since, at least in my work, the control problem reduces to the resolution of an optimization problem, I identify several issues which appear recurrently:

i) There are approaches which provide "nice" formulations but are hard to actually use. I.e., modeling the problem through a mixed-integer formulation (Gros and Zanon 2020) is elegant but scales poorly with problem dimension (computation time, convergence towards the solution, etc.). Similarly, set-based methods have many useful properties but are hard to apply for nonliner dynamics and/or large-scale systems.

ii) Constrained optimization problems (as those resulting from, e.g., MPC implementations (Rosolia and Borrelli 2017)) are difficult to solve with standard tools as the optimal solution is difficult to find (or to have guarantees of optimality). This happens when having non-linear terms, either in the constraints or in the cost (these appear naturally when doing a flat-output representation).

iii) Even when results work flawlessly in simulation, they do it so at the price of considering unrealistic models (simplifications which no longer reflect the process or assumptions about parameters which are not realistic). Take for example the ever popular control torque method. In theory, it permits a "clean" nonlinear feedback linearizable procedure (Westenbroek et al. 2020) but it does so by assuming a perfect knowledge of the system's states.

The common tread in the aforementioned issues is that classical implementations are often not sufficient or excessively complex. The idea is then to use heuristic approaches which provide a sub- or near-optimal solution in significantly less time and in more robust fashion than classical approaches. In what follows I propose to revisit some of my recurrent topics of interest from a data-driven perspective. That is to say, I propose to apply tools from potential field and machine learning theory to reduce the computation times and provide solutions which are close to the optimal ones (with or without guarantees for the optimality gap).

11.1. Potential field methods for navigation in a cluttered environment

Arguably the potential field control topic is a mature one with many results and directions of study. Still, I believe that there are further ways to improve upon the state of the art. In the near future I plan to concentrate on two topics:

i) exploit the structure of the obstacles (given as polyhedral sets) to allow tight maneuvering around them by an agent (or formation of agents);

ii) integrate the potential field into an MPC formulation in order to reduce the risk of local minima and improve the overall performance (faster movement, fewer or no collisions, etc.);

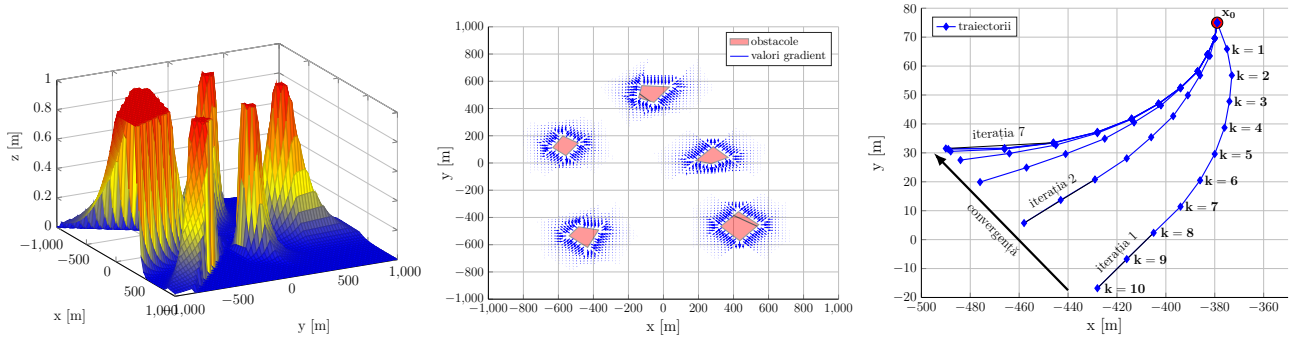
iii) explore learning-based methods which reduce the computational effort, ideally reduce the optimization problem from a nonlinear to a quadratic form.

The key, in my approach is the use of the *sum function* or similar construction which allows to define the potential field in terms of the support hyperplanes which define the obstacles. As noted elsewhere in the manuscript, a collection of polyhedral obstacles may be associated to a hyperplane arrangement which partitions the space into non-overlapping convex regions. The idea which I plan to follow is that inside such a region the values of the potential field and of its gradient are known. The result is then a piecewise formulation for potential field and gradient (similar in kind with the piecewise surfaces obtained by explicit MPC approaches).

Having a piecewise shape (ideally a quadratic or linear one) whose underlying support is given by a union of non-overlapping convex regions (those given by the hyperplane arrangement) means that we could represent it through a mixed-integer formulation. This in turn allows to integrate it in any subsequent optimization problem, e.g., in a MIQP-MPC or NLMIP-MPC construction.

An idea, building on results from (Tallamraju, Rajappa, Black, Karlapalem, and Ahmad 2018) is to run iteratively the MPC implementation for fixed values of the potential field. This method has the advantage of simplifying the optimization problem (maybe even from nonlinear to quadratic) but it requires successive runs until the trajectory converges towards a stable sequence. I find this idea interesting and plan to study it in more detail (e.g., to provide a proof of convergence for the sequence iteration).

To illustrate the aforementioned points I show in Figure 11.1a the resulting potential field for a collection of obstacles when applying the sum function and the associated gradient field in Figure 11.1b. In both cases, it can be seen that the obstacles' shape is still clearly visible. Lastly, Figure 11.1c shows a trajectory, as obtained by the MPC law, computed as in Tallamraju, Rajappa, Black, Karlapalem, and Ahmad 2018 with the addition that I iterate the MPC using the trajectory predicted at the previous run of the MPC (for the same simulation step) until the predicted trajectory becomes stable (it does not noticeably change from one iteration to the next).



(a) potential field (b) gradient (c) learning the trajectory

Figure 11.1.: Proof of concept illustration for potential field based on the sum function

11.2. Data collection through a team of autonomous aerial vehicles

I am greatly interested in providing a reliable framework in which information from a complex environment is gathered through a mesh of fixed ground sensors and a mobile team of Unmanned Aerial Vehicles (UAVs). Communication and power-limited sensors gather and store information about the environment. A motion planning mechanism provides trajectories which ensure data-collection from sensors, avoids forbidden regions while minimizing time and energy costs. Precision agriculture applications are typical benchmarks where a team of UAVs may be deployed to gather data about the

environment through both on-board payloads and clusters of ground sensors. Such a mission requires distributed control strategies and resilient task assignment while simultaneously solving motion planning, control and communication issues in the presence of minimum-energy and minimum-time restrictions.

While the the list of open issues is is very large, my interest lies in several specific directions:

i) How to maximize ground network lifetime through communication rules? Taking into account line-of-sight (LOS) restrictions, available energy and communication costs, partition the ground network into clusters (with possibly time-varying topology) to distribute equally the power consumption induced by communication costs (at ground level inside a cluster and between UAVs and cluster heads).

ii) How to ensure verification of feasibility, state/input bounds? Provide accurate models for the UAVs, which coupled with actuation bounds, sensor measurement limitations and operational constraints (e.g., ensure minimum time in the communication range) lead to clear characterizations of feasibility, input/state bounds and cost (e.g., minimize path length, time and/or energy consumed).

iii) How to generate trajectories ensuring constraint validation/cost minimization? Considering the dynamical models of the agents, propose a mechanism of trajectory generation which takes into account all known limitations (model-based and environmental). Ensure trajectory update when new information becomes available and guarantee tracking performance in the presence of disturbances.

iv) How to ensure fast task allocation and on-line motion planning? Provide a multi-agent reinforcement learning framework to ensure online trajectory reconfiguration, correcting UAVs trajectories with short-time update and safety constraints. Derive theoretical bounds that ensure fast learning and data-efficient algorithms, while minimising the actual sensor data to be sent to the UAVs.

v) How to ensure efficient task allocation with consensus guarantees? Combine optimal control techniques with distributed methods for position estimation and control of multi-agent systems to guarantee performance and satisfaction of constraints both at local and global levels. Offline versus online approaches are to be considered (the later, to account for runtime updates of the environment).

While many of my works deal with UAVs, I have always started from 'standard' dynamics (either fixed-wing or quadcopters). It is then interesting (through a collaboration with University of Porto, mediated by my former Master student Huu Thien Nguyen) for me to approach the motion planning and control mechanisms for reliable route selection, trajectory generation, and tracking for hybrid UAVs. Our preliminary investigations have identified several directions adapted to the particularities of hybrid UAV dynamics (i.e., varying modes of functioning, each with its own operational restrictions):

i) Provide a motion planning mechanism which, exploiting the model dynamics and taking into account operational limitations (available energy, payload weight, list of waypoints, etc.), generates:

   - optimal routes which ensure that the UAV passes near the waypoints, avoids forbidden regions and minimizes its time and energy costs;
   - deep stall maneuvers guaranteeing that the UAV can engage in a reliable and efficient post-stall maneuver, such as landing or take-off, e.g., in case of a missed landing on a predefined target zone.

ii) Apply robust hierarchical control strategies to ensure performance and stability criteria in the presence of disturbances (e.g., assuming Dryden wind turbulence models):

- disturbance rejection and constraint saturation through NMPC at translational and rotational levels;
- tube-based NMPC for trajectory tracking under model variations (normal parameter variation and switching between functioning modes);
- analysis of full control envelopes for all control regimes (e.g., recursive feasibility for NMPC).

iii) Extend the control approach for the coordination of multiple hybrid UAVs:

- trajectory generation for the whole ensemble of UAVs with motion coordination achieved by the highest level supervisor. Both leader-follower and leaderless arrangements are to be considered.
- coordination control synthesis, achieved through a balanced trade-off of the performance of all UAVs while satisfying all the constraints (state and control) and ensuring local stability and robustness.

We are interested in testing these algorithms in a cluttered environment. These benchmarks have particular (and interesting) restrictions: deadlines for package delivery; balancing between path length and order of delivery (a heavy package should be delivered earlier, even if further away, etc.).

11.3. Learning methods for fault handling in electrical grids

While there is an increasing push toward smart metering and improved data collection, large-scale grid networks often suffer from poor observability issues which in turn lead to conservative control architectures and decisions. In particular, fault events are hard to assess (timely detection and precise isolation are not easily achievable). The key in handling such events before they cascade out of control is to estimate the time and place of occurrence as quickly as possible with the limited information available. This means that I am interested in both the sensor placement problem and in the subsequent fault detection and isolation (FDI) problem.

These two issues (sensor placement and FDI) are usually treated separately (one is assumed to work flawlessly and the other is tackled on). In my future work I hope to attack both of them simultaneously in order to improve the performance of the overall scheme. In principle both directions would benefit from a data-driven approach: through an emulator (such as MatPower/MOST for electrical grids) output values for the nominal and under fault functioning are obtained; based on the available information residual signals (sensitive to fault occurrence and robust against 'normal' uncertainties and noises are constructed; these signals are used to decide upon a sensor placement and FDI is carried out. Noteworthy, while there are methods dealing with the concepts of network observability and FDI, they don't always handle well large-scale systems (and keep in mind that electrical grids can easily reach thousands of nodes). Hence, I propose to consider tools from evolutionary computing / machine learning. Some preliminary ideas are:

i) The sensor placement problem is a combinatorial one: select a subset of nodes which (after some criteria) provide the best FDI performance. Using explicitly binary variables to implement a "set covering problem" or a "test covering problem" as in Perelman, Abbas, Koutsoukos, and Amin 2016 works but only for relatively small-sized problems. Hence, I plan to apply evolutionary algorithms as for example a genetic algorithm as done in Casillas, Puig, Garza-Castanón, and Rosich 2013 for sensor selection in a water network (another example of large-scale distribution network).

ii) I propose to implement the fault detection and isolation mechanism with classifiers, commonly used in machine learning applications. Using the available data (the residual signals, labeled into nominal and affected by a a fault) I plan to train various linear classifiers such as k-NN, SVM, Naïve-Bayes or dictionary-based to permit subsequent detection and isolation of the

faults which may appear in the test residuals. This will permit to assess the performance of the FDI mechanism under realistic conditions (uncertainties, noises) and, depending on the chosen method, even quasi-instantaneous decision (the advantage of linear classifiers is that once trained they perform the classification in a negligible amount of time).

Bibliography

Alamo, T., J. M. Bravo, and E. F. Camacho. “Guaranteed state estimation by zonotopes”. In: *Automatica*, pp. 1035–1043 (9 pages), ISSN: 0005-1098.

Alamir, M. and G. Pannochia. “A new formulation of Economic Model Predictive Control without terminal constraint”. In: *arXiv preprint arXiv:2005.04381*, 2020.

Alraddadi, M., A. J. Conejo, and R. M. Lima. “Expansion Planning for Renewable Integration in Power System of Regions with Very High Solar Irradiation”. In: *Journal of Modern Power Systems and Clean Energy*, 2020.
PUBLISHED BY: SGEPRI.

Ananduta, W., J. M. Maestre, C. Ocampo-Martinez, and H. Ishii. “Resilient distributed model predictive control for energy management of interconnected microgrids”. In: *Optimal Control Applications and Methods*, pp. 146–169 (24 pages), 2020.
PUBLISHED BY: Wiley Online Library.

Berberich, J., J. Köhler, M. A. Muller, and F. Allgower. “Data-driven model predictive control with stability and robustness guarantees”. In: *IEEE Transactions on Automatic Control*, 2020.
PUBLISHED BY: IEEE.

Blindheim, S., S. Gros, and T. A. Johansen. “Risk-Based Model Predictive Control for Autonomous Ship Emergency Management”. 2020.

Bordons, C., F. Garcia-Torres, and M. A. Ridao. *Model Predictive Control of Microgrids*, 2020.
PUBLISHED BY: Springer.

Greeff, M. and A. P. Schoellig. “Exploiting differential flatness for robust learning-based tracking control using Gaussian processes”. In: *IEEE Control Systems Letters*, pp. 1121–1126 (6 pages), 2020.
PUBLISHED BY: IEEE.

Gros, S. and M. Zanon. “Reinforcement Learning for Mixed-Integer Problems Based on MPC”. In: *arXiv preprint arXiv:2004.01430*, 2020.

Gros, S., M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl. “From linear to nonlinear MPC: bridging the gap via the real-time iteration”. In: *International Journal of Control*, pp. 62–80 (19 pages), 2020.
PUBLISHED BY: Taylor & Francis.

Westenbroek, T. et al. “Feedback Linearization for Uncertain Systems via Reinforcement Learning”. Pp. 1–8 (8 pages), 2020.

Zafra-Cabeza, A., P. Velarde, and J. M. Maestre. “Multicriteria optimal operation of a microgrid considering risk analysis, renewable resources, and model predictive control”. In: *Optimal Control Applications and Methods*, pp. 94–106 (13 pages), 2020.
PUBLISHED BY: Wiley Online Library.

Zanon, M. and S. Gros. “Safe reinforcement learning using robust MPC”. In: *IEEE Transactions on Automatic Control*, 2020.
PUBLISHED BY: IEEE.

Baros, S. and A. M. Annaswamy. “Distributed optimal wind farm control for fatigue load minimization: A consensus approach”. In: *International Journal of Electrical Power & Energy Systems*, pp. 452–459 (8 pages), 2019.
PUBLISHED BY: Elsevier.

Bouzid, A. E. M. et al. “A novel Decoupled Trigonometric Saturated droop controller for power sharing in islanded low-voltage microgrids”. In: *Electric Power Systems Research*, pp. 146–161 (16 pages),

2019.
Published by: Elsevier.

Budaciu, C., N. Botezatu, M. Kloetzer, and A. Burlacu. "On the Evaluation of the Crazyflie Modular Quadcopter System", pp. 1189–1195 (7 pages). In: *Proc. 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2019. Zaragoza, Spain.

Cataldo, A., I. C. Bertolotti, and R. Scattolini. "Model predictive control tools for evolutionary plants", in: *Factories of the Future*, pp. 39–56 (18 pages), 2019.
Published by: Springer.

Eriksen, B. H., M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke. "The branching-course model predictive control algorithm for maritime collision avoidance". In: *Journal of Field Robotics*, pp. 1222–1249 (28 pages), 2019.
Published by: Wiley Online Library.

Gheneti, B. H. (2019). "Reconfigurable Autonomous Surface Vehicles: perception and trajectory optimization algorithms". PhD thesis. Massachusetts Institute of Technology.

Hernández, J., F. Sanchez-Sutil, and F. Muñoz-Rodríguez. "Design criteria for the optimal sizing of a hybrid energy storage system in PV household-prosumers to maximize self-consumption and self-sufficiency". In: *Energy*, pp. 115827, 2019.
Published by: Elsevier.

Li, S., J. Liu, and R. R. Negenborn. "Distributed coordination for collision avoidance of multiple ships considering ship maneuverability". In: *Ocean Engineering*, pp. 212–226 (15 pages), 2019.
Published by: Elsevier.

Li, S., J. Liu, R. R. Negenborn, and F. Ma. "Optimizing the joint collision avoidance operations of multiple ships from an overall perspective". In: *Ocean Engineering*, pp. 106511, 2019.
Published by: Elsevier.

Lorenzen, M., M. Cannon, and F. Allgöwer. "Robust MPC with recursive model update". In: *Automatica*, pp. 461–471 (11 pages), 2019.
Published by: Elsevier.

Mounier, H., S.-I. Niculescu, A. Cela, and S. Geamanu. "Flatness-based longitudinal vehicle control with embedded torque constraint". In: *IMA Journal of Mathematical Control and Information*, pp. 729–744 (16 pages), 2019.
Published by: Oxford University Press.

Schwenkel, L., J. Köhler, M. A. Müller, and F. Allgöwer. "Robust economic model predictive control without terminal conditions". In: *arXiv preprint arXiv:1911.12235*, 2019.

Siad, S. B., A. Malkawi, G. Damm, L. Lopes, and L. G. Dol. "Nonlinear control of a DC MicroGrid for the integration of distributed generation based on different time scales". In: *International Journal of Electrical Power & Energy Systems*, pp. 93–100 (8 pages), 2019.
Published by: Elsevier.

Soloperto, R., J. Köhler, M. A. Müller, and F. Allgöwer. "Collision avoidance for uncertain nonlinear systems with moving obstacles using robust Model Predictive Control", pp. 811–817 (7 pages). In: *2019 European Control Conference (ECC)*. 2019.

Wang, B. et al. "Hybrid energy storage system using bidirectional single-inductor multiple-port converter with model predictive control in DC microgrids". In: *Electric Power Systems Research*, pp. 38–47 (10 pages), 2019.
Published by: Elsevier.

Wang, S., M. Fu, Y. Wang, and L. Zhao. "A multi-layered potential field method for waterjet propelled unmanned surface vehicle local path planning with minimum energy consumption". In: *Polish Maritime Research*, pp. 351–357 (7 pages), 2019.

Wu, M., F. Gao, P. Yu, J. She, and W. Cao. "Improve Disturbance-Rejection Performance for an Equivalent-Input-Disturbance-Based Control System by Incorporating a Proportional-Integral Observer". In: *IEEE Transactions on Industrial Electronics*, pp. 1254–1260 (7 pages), 2019.
Published by: IEEE.

Alamir, M. "Stability proof for nonlinear MPC design using monotonically increasing weighting profiles without terminal constraints". In: *Automatica*, pp. 455–459 (5 pages), 2018.
PUBLISHED BY: Elsevier.

Berntorp, K., C. Danielson, A. Weiss, and S. Di Cairano. "Positive invariant sets for safe integrated vehicle motion planning and control", pp. 6957–6962 (6 pages). In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018.

Chen, L., H. Hopman, and R. R. Negenborn. "Distributed model predictive control for vessel train formations of cooperative multi-vessel systems". In: *Transportation Research Part C: Emerging Technologies*, pp. 101–118 (18 pages), 2018.
PUBLISHED BY: Elsevier.

Chiang, H.-T. L. and L. Tapia. "COLREG-RRT: An RRT-Based COLREGS-Compliant Motion Planner for Surface Vehicle Navigation". In: *IEEE Robotics and Automation Letters*, pp. 2024–2031 (8 pages), 2018.
PUBLISHED BY: IEEE.

Conte, G. et al. "Development and Experimental Tests of a ROS Multi-agent Structure for Autonomous Surface Vehicles". In: *Journal of Intelligent & Robotic Systems*, pp. 1–14 (14 pages), 2018.
PUBLISHED BY: Springer.

Craig, J. "Introduction to Robotics". 2018.
PUBLISHED BY: Pearson Education.

Drgoňa, J., D. Picard, M. Kvasnica, and L. Helsen. "Approximate model predictive building control via machine learning". In: *Applied Energy*, pp. 199–216 (18 pages), 2018.
PUBLISHED BY: Elsevier.

Filotheou, A., A. Nikou, and D. V. Dimarogonas. "Decentralized control of uncertain multi-agent systems with connectivity maintenance and collision avoidance", pp. 8–13 (6 pages). In: *2018 European Control Conference (ECC)*. 2018.

Iovine, A., T. Rigaut, G. Damm, E. De Santis, and M. D. Di Benedetto. "Power Management for a DC MicroGrid integrating Renewables and Storages". In: *Control Engineering Practice*, pp. 399–405 (7 pages), 2018.

Ke, F., Z. Li, and C. Yang. "Robust tube-based predictive control for visual servoing of constrained differential-drive mobile robots". In: *IEEE Transactions on Industrial Electronics*, pp. 3437–3446 (10 pages), 2018.
PUBLISHED BY: IEEE.

Kim, M., Y. Song, S. Wang, Y. Xia, and X. Jiang. "Secure logistic regression based on homomorphic encryption: Design and evaluation". In: *JMIR medical informatics*, pp. e19, 2018.
PUBLISHED BY: JMIR Publications Inc., Toronto, Canada.

Köhler, J., M. A. Müller, and F. Allgöwer. "Nonlinear reference tracking: An economic model predictive control perspective". In: *IEEE Transactions on Automatic Control*, pp. 254–269 (16 pages), 2018.
PUBLISHED BY: IEEE.

Kotyczka, P. and L. Lefèvre. "Discrete-time port-Hamiltonian systems: A definition based on symplectic integration". In: *arXiv preprint arXiv:1811.07852*, 2018.

Li, T., R. Zhao, C. P. Chen, L. Fang, and C. Liu. "Finite-time formation control of under-actuated ships using nonlinear sliding mode control". In: *IEEE Transactions on Cybernetics*, pp. 3243–3253 (11 pages), 2018.
PUBLISHED BY: IEEE.

Lyche, T., C. Manni, and H. Speleers. "Foundations of spline theory: B-splines, spline approximation, and hierarchical refinement", in: *Splines and PDEs: From Approximation Theory to Numerical Linear Algebra*, pp. 1–76 (76 pages), 2018.
PUBLISHED BY: Springer.

Manogaran, G. and D. Lopez. "Health data analytics using scalable logistic regression with stochastic gradient descent". In: *International Journal of Advanced Intelligence Paradigms*, pp. 118–132 (15

pages), 2018.
Published by: Inderscience Publishers (IEL).

Mohamed, A., F. Martin, and H. Axel. "Nonlinear Model Predictive Control for trajectory tracking and collision avoidance of underactuated vessels with disturbances". In: *Ocean Engineering*, pp. 168–180 (13 pages), 2018.
Published by: ELSEVIER.

Munir, S., M. Hovd, and S. Olaru. "Low complexity constrained control using higher degree Lyapunov functions". In: *Automatica*, pp. 215–222 (8 pages), 2018.
Published by: Elsevier.

Pang, S., J. Wang, J. Liu, and H. Yi. "Three-dimensional leader–follower formation control of multiple autonomous underwater vehicles based on line-of-sight measurements using the backstepping method". In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, pp. 819–829 (11 pages), 2018.
Published by: SAGE Publications Sage UK: London, England.

Shibata, K., N. Shibata, K. Nonaka, and K. Sekiguchi. "Model Predictive Obstacle Avoidance Control for Vehicles with Automatic Velocity Suppression using Artificial Potential Field". In: *IFAC-PapersOnLine*, pp. 313–318 (6 pages), 2018.
Published by: Elsevier.

Tallamraju, R., S. Rajappa, M. J. Black, K. Karlapalem, and A. Ahmad. "Decentralized mpc based obstacle avoidance for multi-robot target tracking scenarios", pp. 1–8 (8 pages). In: *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2018.

Verbandt, M. et al. "Linear Control Toolbox-supporting B-splines in LPV control". In: *Mechatronics*, pp. 78–89 (12 pages), 2018.
Published by: Elsevier.

Wang, N., Z. Sun, J. Yin, S.-F. Su, and S. Sharma. "Finite-time observer based guidance and control of underactuated surface vehicles with unknown sideslip angles and disturbances". In: *IEEE Access*, pp. 14059–14070 (12 pages), 2018.
Published by: IEEE.

Yu, P., M. Wu, J. She, K.-Z. Liu, and Y. Nakanishi. "Robust tracking and disturbance rejection for linear uncertain system with unknown state delay and disturbance". In: *IEEE/ASME Transactions on Mechatronics*, pp. 1445–1455 (11 pages), 2018.
Published by: IEEE.

Zanelli, A., G. Horn, G. Frison, and M. Diehl. "Nonlinear model predictive control of a human-sized quadrotor", pp. 1542–1547 (6 pages). In: *Proc. 2018 European Control Conference (ECC'18)*. 2018. Limassol, Cyprus.

Ballesteros-Tolosana, I., S. Olaru, P. Rodriguez-Ayerbe, G. Pita-Gil, and R. Deborne. "Collision-free trajectory planning for overtaking on highways", pp. 2551–2556 (6 pages). In: *2017 Conference on Decision and Control (CDC)*. 2017.

Bekcheva, M., H. Mounier, and L. Greco. "Control of differentially flat linear delay systems with constraints". In: *IFAC-PapersOnLine*, pp. 13348–13353 (6 pages), 2017.
Published by: Elsevier.

Berntorp, K., A. Weiss, C. Danielson, I. V. Kolmanovsky, and S. Di Cairano. "Automated driving: safe motion planning using positively invariant sets", pp. 1–6 (6 pages). In: *20th International Conference on Intelligent Transportation Systems (ITSC)*. 2017.

Cafieri, S. and R. Omheni. "Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes". In: *European Journal of Operational Research*, pp. 283–290 (8 pages), 2017.
Published by: Elsevier.

Chao, W., M. Feng, W. Qing, and W. Shuwu. "A situation awareness approach for USV based on Artificial Potential Fields", pp. 232–235 (4 pages). In: *Transportation Information and Safety (ICTIS), 2017 4th International Conference on*. 2017.

Cortes, C. A., S. F. Contreras, and M. Shahidehpour. "Microgrid topology planning for enhancing the reliability of active distribution networks". In: *IEEE Transactions on Smart Grid*, pp. 6369–6377 (9 pages), 2017.
PUBLISHED BY: IEEE.

Al-Daffaie, K. and S. Khan. "Logistic regression for circular data", pp. 030022. In: *AIP Conference Proceedings*. 2017.

Department of Energy Office of Energy Efficiency & Renewable Energy, E. E. R. E. (2017). *Commercial and Residential Hourly Load Profiles*.

Fang, H., Y. Wei, J. Chen, and B. Xin. "Flocking of second-order multiagent systems with connectivity preservation based on algebraic connectivity estimation". In: *IEEE transactions on cybernetics*, pp. 1067–1077 (11 pages), 2017.
PUBLISHED BY: IEEE.

Giernacki, W., M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski. "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering", pp. 37–42 (6 pages). In: *Proceedings of the 22nd IEEE International Conference on Methods and Models in Automation and Robotics (MMAR'22)*. 2017.

Grüne, L. and J. Pannek. "Nonlinear model predictive control", in: *Nonlinear Model Predictive Control*, pp. 45–69 (25 pages), 2017.
PUBLISHED BY: Springer.

Gulan, M. et al. "Embedded linear model predictive control for 8-bit microcontrollers via convex lifting". In: *IFAC-PapersOnLine*, pp. 10697–10704 (8 pages), 2017.
PUBLISHED BY: Elsevier.

Janecek, F., M. Klauco, M. Kaluz, and M. Kvasnica. "OPTIPLAN: A Matlab Toolbox for Model Predictive Control with Obstacle Avoidance". In: *IFAC-PapersOnLine*, pp. 531–536 (6 pages), 2017.
ISSN: 2405-8963.

Jawad, H., R. Nordin, S. Gharghan, A. Jawad, and M. Ismail. "Energy-efficient wireless sensor networks for precision agriculture: A review". In: *Sensors*, pp. 1781, 2017.

Liu, Z., C. Geng, and J. Zhang. "Model predictive controller design with disturbance observer for path following of unmanned surface vessel", pp. 1827–1832 (6 pages). In: *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*. 2017.

Lou, G., W. Gu, Y. Xu, M. Cheng, and W. Liu. "Distributed MPC-based secondary voltage control scheme for autonomous droop-controlled microgrids". In: *IEEE Transactions on Sustainable Energy*, pp. 792–804 (13 pages), 2017.
PUBLISHED BY: IEEE.

Lu, H., C. Liu, L. Guo, and W.-H. Chen. "Constrained anti-disturbance control for a quadrotor based on differential flatness". In: *International Journal of Systems Science*, pp. 1182–1193 (12 pages), 2017.
PUBLISHED BY: Taylor & Francis.

Lyu, H. and Y. Yin. "Ship's trajectory planning for collision avoidance at sea based on modified artificial potential field", pp. 351–357 (7 pages). In: *Robotics and Automation Engineering (ICRAE), 2017 2nd International Conference on*. 2017.

Mercy, T., R. Van Parys, and G. Pipeleers. "Spline-based motion planning for autonomous guided vehicles in a dynamic environment". In: *IEEE Tran. on Control Systems Technology*, pp. 1–8 (8 pages), 2017.
PUBLISHED BY: IEEE.

Mortaz, E. and J. Valenzuela. "Microgrid energy scheduling using storage from electric vehicles". In: *Electric Power Systems Research*, pp. 554–562 (9 pages), 2017.
PUBLISHED BY: Elsevier.

Rosolia, U. and F. Borrelli. "Learning model predictive control for iterative tasks. a data-driven control framework". In: *IEEE Transactions on Automatic Control*, pp. 1883–1896 (14 pages), 2017.
PUBLISHED BY: IEEE.

Samad, T. and A. M. Annaswamy. "Controls for smart grids: Architectures and applications". In: *Proceedings of the IEEE*, pp. 2244–2261 (18 pages), 2017.
PUBLISHED BY: IEEE.

Speleers, H. "Hierarchical spline spaces: quasi-interpolants and local approximation estimates". In: *Advances in Computational Mathematics*, pp. 235–255 (21 pages), 2017.
PUBLISHED BY: Springer.

Szmuk, M., C. A. Pascucci, D. Dueri, and B. Acikmese. "Convexification and real-time on-board optimization for agile quad-rotor maneuvering and obstacle avoidance", in: *2017 International Conference on Intelligent Robots and Systems (IROS)*. 2017.

Van Parys, R. and G. Pipeleers. "Spline-based motion planning in an obstructed 3D environment". In: *IFAC-PapersOnLine*, pp. 8668–8673 (6 pages), 2017.
PUBLISHED BY: Elsevier.

Vittek, J., P. Butko, B. Ftorek, P. Makys, and L. Gorel. "Energy Near-Optimal Control Strategies for Industrial and Traction Drives with a.c. Motors". In: *Mathematical Problems in Engineering*, 2017.

Vu, T. V. et al. "Robust adaptive droop control for DC microgrids". In: *Electric Power Systems Research*, pp. 95–106 (12 pages), 2017.
PUBLISHED BY: Elsevier.

Weiss, A., C. Danielson, K. Berntorp, I. Kolmanovsky, and S. D. Cairano. "Motion planning with invariant set trees", pp. 1625–1630 (6 pages). In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. 2017.

Wu, X. and A. J. Conejo. "An efficient tri-level optimization model for electric grid defense planning". In: *IEEE Transactions on Power Systems*, pp. 2984–2994 (11 pages), 2017.
PUBLISHED BY: IEEE.

Wytock, M., N. Moehle, and S. Boyd. "Dynamic energy management with scenario-based robust MPC", pp. 2042–2047 (6 pages). In: *2017 American Control Conference (ACC)*. 2017.

Xie, X., D. Yue, H. Zhang, and C. Peng. "Control synthesis of discrete-time T–S fuzzy systems: Reducing the conservatism whilst alleviating the computational burden". In: *IEEE transactions on cybernetics*, pp. 2480–2491, 2017.
PUBLISHED BY: IEEE.

Zhu, Q., J. Ma, Z. Liu, and K. Liu. "Containment control of autonomous surface vehicles: A nonlinear disturbance observer–based dynamic surface control design". In: *Advances in Mechanical Engineering*, pp. 1687814017727384, 2017.
PUBLISHED BY: SAGE Publications Sage UK: London, England.

Abdelaal, M., M. Fränzle, and A. Hahn. "NMPC-based trajectory tracking and collison avoidance of underactuated vessels with elliptical ship domain". In: *IFAC-PapersOnLine*, pp. 22–27 (6 pages), 2016.
PUBLISHED BY: Elsevier.

Baghaee, H. R., M. Mirsalim, and G. B. Gharehpetian. "Performance improvement of multi-DER microgrid for small-and large-signal disturbances and nonlinear loads: novel complementary control loop and fuzzy controller in a hierarchical droop-based control scheme". In: *IEEE Systems Journal*, pp. 444–451 (8 pages), 2016.
PUBLISHED BY: IEEE.

Bakari, H., T. Adegoke, and A. Yahya. "Application of Newton Rhapson Method to Non-linear Models". In: *International Journal of Mathematics and Statistics Studies*, pp. 21–31 (11 pages), 2016.

Chen, C., J. Wang, F. Qiu, and D. Zhao. "Resilient distribution system by microgrids formation after natural disasters". In: *IEEE Transactions on Smart Grid*, pp. 958–966 (9 pages), 2016.
PUBLISHED BY: IEEE.

Chen, Y.-b., G.-c. Luo, Y.-s. Mei, J.-q. Yu, and X.-l. Su. "UAV path planning using artificial potential field method updated by optimal control theory". In: *International Journal of Systems Science*, pp. 1407–1420 (14 pages), 2016. ISSN: 0020-7721.

Engelhardt, T., T. Konrad, B. Schäfer, and D. Abel. "Flatness-based control for a quadrotor camera helicopter using model predictive control trajectory generation", pp. 852–859 (8 pages). In: *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED'16)*. 2016.
PUBLISHED BY: IEEE.

Gavagsaz-Ghoachani, R. et al. "Stability issue of DC-DC converters with input LC filter via flatness-based control", pp. 1–6 (6 pages). In: *2016 IEEE Transportation Electrification Conference and Expo (ITEC)*. 2016.

Johansen, T. A., T. Perez, and A. Cristofaro. "Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment". In: *IEEE transactions on intelligent transportation systems*, pp. 3407–3422 (16 pages), 2016.
PUBLISHED BY: IEEE.

Khodabakhsh, R. and S. Sirouspour. "Optimal control of energy storage in a microgrid by minimizing conditional value-at-risk". In: *IEEE Transactions on Sustainable Energy*, pp. 1264–1273 (10 pages), 2016.

Kyurkchiev, N. and S. Markov. "Approximation of the cut function by some generic logistic functions and applications". In: *Advances in Applied Sciences*, pp. 24–29 (6 pages), 2016.

Liu, L., X. Meng, and C. Liu. "A review of maximum power point tracking methods of PV power system at uniform and partial shading". In: *Renewable and Sustainable Energy Reviews*, pp. 1500–1507 (8 pages), 2016.
PUBLISHED BY: Elsevier.

Manyam, S., D. Casbeer, and K. Sundar. "Path planning for cooperative routing of air-ground vehicles". In: *arXiv preprint:1605.09739*, 2016.

National Renewable Energy Laboratory, N. R. E. L. (2016). *Rural African Commercial Load Profile*.

Parisio, A., E. Rikos, and L. Glielmo. "Stochastic model predictive control for economic/environmental operation management of microgrids: An experimental case study". In: *Journal of Process Control*, pp. 24–37 (14 pages), 2016.
PUBLISHED BY: Elsevier.

Perelman, L. S., W. Abbas, X. Koutsoukos, and S. Amin. "Sensor placement for fault location identification in water networks: A minimum test cover approach". In: *Automatica*, pp. 166–176 (11 pages), 2016.
PUBLISHED BY: Elsevier.

Schiffer, J., E. Fridman, R. Ortega, and J. Raisch. "Stability of a class of delayed port-Hamiltonian systems with application to microgrids with distributed rotational and electronic generation". In: *Automatica*, pp. 71–79 (9 pages), 2016.
PUBLISHED BY: Elsevier.

Ahumada, C., R. Cárdenas, D. Saez, and J. M. Guerrero. "Secondary control strategies for frequency restoration in islanded microgrids with consideration of communication delays". In: *IEEE Transactions on Smart Grid*, pp. 1430–1441 (12 pages), 2015.
PUBLISHED BY: IEEE.

Althoff, M. "CORA 2015 Manual". In: *TU Munich*, 2015.

Badgwell, T. A. and S. J. Qin. "Model-predictive control in practice". In: *Encyclopedia of Systems and Control*, pp. 756–760 (5 pages), 2015.
PUBLISHED BY: Springer.

Bays, M. J., R. D. Tatum, L. Cofer, and J. R. Perkins. "Automated scheduling and mission visualization for mine countermeasure operations", pp. 1–7 (7 pages). In: *OCEANS 2015-MTS/IEEE Washington*. 2015.

Escobar, G., J. van der Schaft, and R. Ortega. "A Hamiltonian viewpoint in the modeling of switching power converters". In: *Automatica*, pp. 445–45, 2015.

Franzè, G. and W. Lucia. "The obstacle avoidance motion planning problem for autonomous vehicles: A low-demanding receding horizon control scheme". In: *Systems & Control Letters*, pp. 1–10 (10

pages), 2015.
Published by: Elsevier.

Hehn, M. and R. D'Andrea. "Real-time trajectory generation for quadrocopters". In: *IEEE Transactions on Robotics*, pp. 877–892 (16 pages), 2015.
Published by: IEEE.

Lifshitz, D. and G. Weiss. "Optimal energy management for grid-connected storage systems". In: *Optimal Control: Application and Methods*, pp. 447–462 (16 pages), 2015.

Simpson-Porco, J. W. et al. "Secondary frequency and voltage control of islanded microgrids via distributed averaging". In: *IEEE Transactions on Industrial Electronics*, pp. 7025–7038 (14 pages), 2015.
Published by: IEEE.

Van Loock, W., G. Pipeleers, and J. Swevers. "Optimal motion planning for differentially flat systems with guaranteed constraint satisfaction", pp. 4245–4250 (6 pages). In: *2015 American Control Conference*. 2015.

Wang, P., X. Lu, X. Yang, W. Wang, and D. Xu. "An improved distributed secondary control method for DC microgrids with enhanced dynamic current sharing performance". In: *IEEE Transactions on Power Electronics*, pp. 6658–6673, 2015.
Published by: IEEE.

Bipin, K., V. Duggal, and K. M. Krishna. "Autonomous navigation of generic quadrocopter with minimum time trajectory planning and control", pp. 66–71 (6 pages). In: *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. 2014.

Bracco, S., F. Delfino, F. Pampararo, M. Robba, and M. Rossi. "A mathematical model for the optimal operation of the University of Genoa Smart Polygeneration Microgrid: Evaluation of technical, economic and environmental performance indicators". In: *Energy*, pp. 912–922 (11 pages), 2014.
Published by: Elsevier.

CIAT entreprise, U. T. C. (2014). *Temperature and irradiation data*.

Khodaei, A. "Resiliency-oriented microgrid optimal scheduling". In: *IEEE Transactions on Smart Grid*, pp. 1584–1591 (8 pages), 2014.
Published by: IEEE.

Mahmoud, M. S., S. A. Hussain, and M. A. Abido. "Modeling and control of microgrid: An overview". In: *Journal of the Franklin Institute*, pp. 2822–2859 (38 pages), 2014.
Published by: Elsevier.

Parisio, A., E. Rikos, and L. Glielmo. "A model predictive control approach to microgrid operation optimization". In: *IEEE Transactions on Control Systems Technology*, pp. 1813–1827 (15 pages), 2014.
Published by: IEEE.

Rahimiyan, M., L. Baringo, and A. J. Conejo. "Energy management of a cluster of interconnected price-responsive demands". In: *IEEE Transactions on Power Systems*, pp. 645–655 (11 pages), 2014.

Schaft, A. van der, D. Jeltsema, et al. "Port-Hamiltonian systems theory: An introductory overview". In: *Foundations and Trends® in Systems and Control*, pp. 173–378 (206 pages), 2014.
Published by: Now Publishers, Inc.

Shafiee, Q., J. M. Guerrero, and J. C. Vasquez. "Distributed secondary control for islanded microgrids—A novel approach". In: *IEEE Transactions on power electronics*, pp. 1018–1031 (14 pages), 2014.
Published by: IEEE.

Yang, Y., J. Du, H. Liu, C. Guo, and A. Abraham. "A trajectory tracking robust controller of surface vessels with disturbance uncertainties". In: *IEEE Transactions on Control Systems Technology*, pp. 1511–1518 (8 pages), 2014.
Published by: IEEE.

Casillas, M. V., V. Puig, L. E. Garza-Castanón, and A. Rosich. "Optimal sensor placement for leak location in water distribution networks using genetic algorithms". In: *Sensors*, pp. 14984–15005 (22

pages), 2013.
PUBLISHED BY: Multidisciplinary Digital Publishing Institute.

Ferrara, A., G. P. Incremona, and L. Magni. "A robust MPC/ISM hierarchical multi-loop control scheme for robot manipulators", pp. 3560–3565 (6 pages). In: *IEEE 52nd Annual Conference on Decision and Control (CDC)*. 2013.

Franke, M. and K. Robenack. "On the computation of flat outputs for nonlinear control systems", pp. 167–172. In: *Control Conference (ECC), 2013 European*. 2013.

Hovd, M. and F. Stoican. "On the design of exact penalty functions for MPC using mixed integer programming". In: *International Journal of Computers & Chemical Engineering*, 2013. ISSN: 0098-1354.
DOI: 10.1016/j.compchemeng.2013.07.001.

Mueller, M. W. and R. D'Andrea. "A model predictive controller for quadrocopter state interception", pp. 1383–1389 (7 pages). In: *Proc. 2013 IEEE European Control Conference (ECC'13)*. 2013. Zürich, Switzerland.

Pérez, E., H. Beltran, N. Aparicio, and P. Rodriguez. "Predictive power control for PV plants with energy storage". In: *IEEE Transactions on Sustainable Energy*, pp. 482–490 (9 pages), 2013.

Christensen, O. and P. Massopust. "Exponential B-splines and the partition of unity property". In: *Advances in Computational Mathematics*, pp. 301–318 (18 pages), 2012.
PUBLISHED BY: Springer.

Giannelli, C., B. Jüttler, and H. Speleers. "THB-splines: The truncated basis for hierarchical splines". In: *Computer Aided Geometric Design*, pp. 485–498 (14 pages), 2012.
PUBLISHED BY: Elsevier.

Grüne, L. "NMPC without terminal constraints". In: *IFAC Proceedings Volumes*, pp. 1–13 (13 pages), 2012.
PUBLISHED BY: Elsevier.

Herissé, B., T. Hamel, R. Mahony, and F.-X. Russotto. "Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow". In: *IEEE Transactions on robotics*, pp. 77–89 (13 pages), 2012.
PUBLISHED BY: IEEE.

Hooshmand, A., M. Poursaeidi, J. Mohammadpour, H. Malki, and K. Grigoriads. "Stochastic model predictive control method for microgrid management", pp. 1–7 (7 pages). In: *IEEE PESInnovative Smart Grid Technologies (ISGT)*. 2012.

Latombe, J.-C. *Robot motion planning*, 2012.
PUBLISHED BY: Springer.

Lo Prete, C. et al. "Sustainability and reliability assessment of microgrids in a regional electricity market". In: *Energy*, pp. 192–202 (11 pages), 2012.
PUBLISHED BY: Elsevier.

Löfberg, J. "Oops! I cannot do it again: Testing for recursive feasibility in MPC". In: *Automatica*, pp. 550–555 (6 pages), 2012.
PUBLISHED BY: Elsevier.

Müller, M. A., M. Reble, and F. Allgöwer. "Cooperative control of dynamically decoupled systems via distributed model predictive control". In: *International Journal of Robust and Nonlinear Control*, pp. 1376–1397 (22 pages), 2012.
PUBLISHED BY: Wiley Online Library.

Piegl, L. and W. Tiller. *The NURBS book*, 2012.
PUBLISHED BY: Springer Science & Business Media.

Qi, W., J. Liu, and P. Christofides. "Supervisory predictive control for long-term scheduling of an integrated wind/solar energy generation and water desalination system". In: *IEEE Transactions on Control Systems Technology*, pp. 504–512 (9 pages), 2012.

Reble, M. and F. Allgöwer. "Unconstrained model predictive control and suboptimality estimates for nonlinear continuous-time systems". In: *Automatica*, pp. 1812–1817 (6 pages), 2012.
PUBLISHED BY: Elsevier.

Schoellig, A. P., F. L. Mueller, and R. D'andrea. "Optimization-based iterative learning for precise quadrocopter trajectory tracking". In: *Autonomous Robots*, pp. 103–127 (25 pages), 2012.
PUBLISHED BY: Springer.

Suryawan, F., J. De Doná, and M. Seron. "Splines and polynomial tools for flatness-based constrained motion planning". In: *Int. Journal of Systems Science*, pp. 1396–1411 (16 pages), 2012.
PUBLISHED BY: Taylor & Francis.

Zapryanova, T. and G. Tachev. "Generalized inverse theorem for Schoenberg operator". In: *Journal of Modern Mathematics Frontier*, pp. 11–16 (6 pages), 2012.
PUBLISHED BY: Science
PUBLISHED BY: Engineering Publishing Company.

Ziegler, G. M. *Lectures on polytopes*, 2012.
PUBLISHED BY: Springer.

Grüne, L. and J. Pannek. *Nonlinear Model Predictive Control: Theory and Algorithms*, 2011.
PUBLISHED BY: Springer-Verlag, London.

Hehn, M. and R. D'Andrea. "Quadrocopter trajectory generation and control". In: *IFAC Proceedings Volumes*, pp. 1485–1491 (7 pages), 2011.
PUBLISHED BY: Elsevier.

Karaman, S. and E. Frazzoli. "Sampling-based algorithms for optimal motion planning". In: *The International Journal of Robotics Research*, pp. 846–894 (49 pages), 2011. ISSN: 0278-3649.

Kuwata, Y., M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger. "Safe maritime navigation with COLREGS using velocity obstacles", pp. 4728–4734 (7 pages). In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. 2011.

Louembet, C. and G. Deaconu. "Collision avoidance in low thrust rendezvous guidance using flatness and positive B-splines", pp. 456–461 (6 pages). In: *Proceedings of the 2011 American Control Conference*. 2011.

Parisio, A. and L. Glielmo. "Energy efficient microgrid management using Model Predictive Control", pp. 5449–5454 (6 pages). In: *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*. 2011.

Sanseverino, E., M. Di Silvestre, M. Ippolito, A. De Paola, and G. Lo Re. "An execution, monitoring and replanning approach for optimal energy management in microgrids". In: *Energy*, pp. 3429–3436 (8 pages), 2011.
PUBLISHED BY: Elsevier.

Tsikalakis, A. and N. Hatziargyriou. "Centralized control for optimizing microgrids operation", pp. 1–8 (8 pages). In: *IEEE Power and Energy Society General Meeting*. 2011.

Vielma, J. P. and G. L. Nemhauser. "Modeling disjunctive constraints with a logarithmic number of binary variables and constraints". In: *Mathematical Programming*, pp. 49–72 (24 pages), 2011.
PUBLISHED BY: Springer.

Althoff, M., O. Stursberg, and M. Buss. "Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes". In: *Nonlinear analysis: hybrid systems*, pp. 233–249 (17 pages), 2010.

Do, K. D. "Practical control of underactuated ships". In: *Ocean Engineering*, pp. 1111–1119 (9 pages), 2010.
PUBLISHED BY: Elsevier.

Gover, E. and N. Krikorian. "Determinants and the volumes of parallelotopes and zonotopes". In: *Linear Algebra and its Applications*, pp. 28–40 (13 pages), 2010.

Khalid, M. and A. Savkin. "A model predictive control approach to the problem of wind power smoothing with controlled battery storage". In: *Renewable Energy*, pp. 1520–1526 (7 pages), 2010.

Liu, J., X. Chen, D. Muñoz de la Peña, and P. D. Christofides. "Sequential and iterative architectures for distributed model predictive control of nonlinear process systems". In: *AIChE Journal*, pp. 2137–2149 (13 pages), 2010.
PUBLISHED BY: Wiley Online Library.

Louembet, C., F. Cazaurang, and A. Zolghadri. "Motion planning for flat systems using positive B-splines: An LMI approach". In: *Automatica*, pp. 1305–1309 (5 pages), 2010.
PUBLISHED BY: Elsevier.

Milovanović, G. V. and Z. Udovičić. "Calculation of coefficients of a cardinal B-spline". In: *Applied mathematics letters*, pp. 1346–1350 (5 pages), 2010.
PUBLISHED BY: Elsevier.

Olaru, S., J. A. De Doná, M. M. Seron, and F. Stoican. "Positive invariant sets for fault tolerant multisensor control schemes". In: *International Journal of Control*, pp. 2622–2640 (19 pages), 2010.
PUBLISHED BY: Taylor & Francis.

Pfingsthorn, M., A. Birk, and H. Bülow. "An efficient strategy for data exchange in multi-robot mapping under underwater communication constraints", pp. 4886–4893 (8 pages). In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010.
DOI: 10.1109/IROS.2010.5650270.

Polyuga, R. V. and A. Van der Schaft. "Structure preserving model reduction of port-Hamiltonian systems by moment matching at infinity". In: *Automatica*, pp. 665–672 (8 pages), 2010.
PUBLISHED BY: Elsevier.

Puig, V. "Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies". In: *International Journal of Applied Mathematics and Computer Science*, pp. 619–635 (17 pages), 2010.
PUBLISHED BY: Versita.

Divya, K. and J. Østergaard. "Battery energy storage technology for power systems—An overview". In: *Electric Power Systems Research*, pp. 511–520 (10 pages), 2009.
PUBLISHED BY: Elsevier.

Duindam, V., A. Macchelli, S. Stramigioli, and H. Bruyninckx. *Modeling and control of complex physical systems: the port-Hamiltonian approach*, 2009.
PUBLISHED BY: Springer Science & Business Media.

Hawkes, A., D. Brett, and N. Brandon. "Fuel cell micro-CHP techno-economics: Part 1–model concept and formulation". In: *International Journal of Hydrogen Energy*, pp. 9545–9557 (13 pages), 2009.
PUBLISHED BY: Elsevier.

Levine, J. *Analysis and control of nonlinear systems: A flatness-based approach*, 2009.
PUBLISHED BY: Springer Science & Business Media.

Negenborn, R., M. Houwing, B. De Schutter, and J. Hellendoorn. "Model predictive control for residential energy resources using a mixed-logical dynamic model", pp. 702–707 (6 pages). In: *International Conference on Networking, Sensing and Control*. 2009.

Østergaard, K. Z., J. Stoustrup, and P. Brath. "Linear parameter varying control of wind turbines covering both partial load and full load conditions". In: *International Journal of Robust and Nonlinear Control*, pp. 92–116 (25 pages), 2009.

Srinivasan, B., P. Huguenin, and D. Bonvin. "Global stabilization of an inverted pendulum–Control strategy and experimental verification". In: *Automatica*, pp. 265–269 (5 pages), 2009.
PUBLISHED BY: Elsevier.

Mazumder, S. K., M. Tahir, and K. Acharya. "Master-slave current-sharing control of a parallel dc-dc converter system over an RF communication interface". In: *IEEE transactions on industrial electronics (1982)*, pp. 59, 2008.

Vitus, M. P., S. L. Waslander, and C. J. Tomlin. "Locally optimal decomposition for autonomous obstacle avoidance with the tunnel-MILP algorithm", pp. 540–545 (6 pages). In: *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. 2008.

Zervas, P., H. Sarimveis, J. Palyvos, and N. Markatos. "Model-based optimal control of a hybrid power generation system consisting of photovoltaic arrays and fuel cells". In: *Journal of Power Sources*, pp. 327–338 (12 pages), 2008.
PUBLISHED BY: Elsevier.

Cowling, I. D., O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke. "A prototype of an autonomous controller for a quadrotor UAV", pp. 4001–4008 (8 pages). In: *European Control Conference (ECC)*. 2007.

Diaf, S., D. Diaf, M. Belhamel, M. Haddadi, and A. Louche. "A methodology for optimal sizing of autonomous hybrid PV/wind system". In: *Energy Policy*, pp. 5708–5718 (11 pages), 2007.

Hsu, D., J.-C. Latombe, and H. Kurniawati. "On the probabilistic foundations of probabilistic roadmap planning", in: *Robotics Research*, pp. 83–97 (15 pages), 2007.
PUBLISHED BY: Springer.

Kofman, E., H. Haimovich, and M. M. Seron. "A systematic method to obtain ultimate bounds for perturbed systems". In: *International Journal of Control*, pp. 167–178 (12 pages), 2007.
PUBLISHED BY: Taylor & Francis.

Real, A. del, A. Arce, and C. Bordons. "Hybrid model predictive control of a two-generator power plant integrating photovoltaic panels and a fuel cell", pp. 5447–5452 (6 pages). In: *Proceedings of the 46th IEEE Conference on Decision and Control*. 2007.

Alamir, M. *Stabilization of nonlinear systems using receding-horizon control schemes: a parametrized approach for fast systems*, 2006.
PUBLISHED BY: Springer.

Fredriksen, E. and K. Y. Pettersen. "Global κ-exponential way-point maneuvering of ships: Theory and experiments". In: *Automatica*, pp. 677–687 (11 pages), 2006.
PUBLISHED BY: Elsevier.

LaValle, S. M. *Planning algorithms*, 2006.
PUBLISHED BY: Cambridge university press.

Limón, D., T. Alamo, F. Salas, and E. F. Camacho. "On the stability of constrained MPC without terminal constraint". In: *IEEE Transactions on Automatic Control*, pp. 832–836 (5 pages), 2006.
PUBLISHED BY: IEEE.

Craig, J. J. *Introduction to robotics: mechanics and control*, 2005.
PUBLISHED BY: Pearson/Prentice Hall Upper Saddle River, NJ, USA:

Crocker, T. R. (2005). "Power converter and method for power conversion". US Patent 6914420.

Mayne, D. Q., M. M. Seron, and S. Raković. "Robust model predictive control of constrained linear systems with bounded disturbances". In: *Automatica*, pp. 219–224 (6 pages), 2005.
PUBLISHED BY: Elsevier.

Puri, A. "A survey of unmanned aerial vehicles UAV for traffic surveillance". In: *Department of computer science and engineering, Univ. of South Florida*, pp. 1–29 (29 pages), 2005.

Camacho, E. F. and C. Bordons. "Model predictive control. Advanced textbooks in control and signal processing". In: *Springer-Verlag, London*, 2004.

Fukuda, K. "From the zonotope construction to the Minkowski addition of convex polytopes". In: *Journal of Symbolic Computation*, pp. 1261–1272 (12 pages), 2004.

Langson, W., I. Chryssochoos, S. Raković, and D. Q. Mayne. "Robust model predictive control using tubes". In: *Automatica*, pp. 125–133 (9 pages), 2004.
PUBLISHED BY: Elsevier.

Löfberg, J. "YALMIP: A toolbox for modeling and optimization in MATLAB", in: *Proc. of the CACSD Conference*. 2004.

Stanley, R. P. "An Introduction to Hyperplane Arrangements", in: *Lecture notes, IAS/Park City Mathematics Institute*. 2004.

Stanley, R. P. et al. "An introduction to hyperplane arrangements". In: *Geometric combinatorics*, pp. 389–496 (108 pages), 2004.

Cannon, M., V. Deshmukh, and B. Kouvaritakis. "Nonlinear model predictive control with polytopic invariant sets". In: *Automatica*, pp. 1487–1494 (8 pages), 2003.
PUBLISHED BY: Elsevier.

Guibas, L. J., A. Nguyen, and L. Zhang. "Zonotopes As Bounding Volumes", pp. 803–812 (10 pages). In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. 2003.
PUBLISHED BY: SIAM. Philadelphia, PA, USA.

Lewis, F. L., D. M. Dawson, and C. T. Abdallah. *Robot manipulator control: theory and practice*, 2003.
PUBLISHED BY: CRC Press.

Löfberg, J. *Minimax approaches to robust model predictive control*, 2003.
PUBLISHED BY: Linköping Univ. Electronic Press.

Bemporad, A., M. Morari, V. Dua, and E. N. Pistikopoulos. "The explicit linear quadratic regulator for constrained systems". In: *Automatica*, pp. 3–20 (18 pages), 2002.
PUBLISHED BY: Elsevier.

Beutel, L., H. Gonska, D. Kacsó, and G. Tachev. "On variation-diminishing Schoenberg operators: new quantitative statements". In: *Multivariate Approximation and Interpoltaion with Applications (ed. by M. Gasca), Monogr. Academia Ciencas de Zaragoza*, pp. 9–58 (50 pages), 2002.

Fossen, T. I. *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*, 2002.
PUBLISHED BY: Marine Cybernetics.

Olfati-Saber, R. and R. M. Murray. "Distributed cooperative control of multiple vehicle formations using structural potential functions", pp. 242–248 (7 pages). In: *IFAC world congress*. 2002.

Ortega, R., A. Van Der Schaft, B. Maschke, and G. Escobar. "Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems". In: *Automatica*, pp. 585–596 (12 pages), 2002.
PUBLISHED BY: Elsevier.

Richard, P.-Y., J. Buisson, and H. Cormerais. "Analysis of flatness using bond graphs and bicausality". In: *IFAC Proceedings Volumes*, pp. 25–30 (6 pages), 2002.
PUBLISHED BY: Elsevier.

Richards, A. and J. P. How. "Aircraft trajectory planning with collision avoidance using mixed integer linear programming", pp. 1936–1941 (6 pages). In: *Proceedings of the 2002 American Control Conference.* 2002.

Wächter, A. (2002). "An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering". PhD thesis. Carnegie Mellon University.

Hazewinkel, M. *Multinomial coefficient, Encyclopedia of Mathematics*, 2001.
PUBLISHED BY: Springer.

Hovd, M. and R. D. Braatz. "Handling state and output constraints in MPC using time-dependent weights", pp. 2418–2423 (6 pages). In: *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*. 2001.

Chen, W.-H., D. J. Ballance, P. J. Gawthrop, and J. O'Reilly. "A nonlinear disturbance observer for robotic manipulators". In: *IEEE Transactions on industrial Electronics*, pp. 932–938 (7 pages), 2000.
PUBLISHED BY: IEEE.

Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. Scokaert. "Constrained model predictive control: Stability and optimality". In: *Automatica*, pp. 789–814 (26 pages), 2000.
PUBLISHED BY: Elsevier.

Poignet, P. and M. Gautier. "Nonlinear model predictive control of a robot manipulator", pp. 401–406 (6 pages). In: *Proceedings. 6th International Workshop on Advanced Motion Control*. 2000.

Commandant, U. "International regulations for prevention of collisions at sea, 1972 (72 COLREGS)". In: *US Department of Transportation, US Coast Guard*, 1999.

Fliess, M., J. Levine, P. Martin, and P. Rouchon. "A Lie-Backlund approach to equivalence and flatness of nonlinear systems". In: *IEEE Tran. on Automatic Control*, pp. 922–937 (16 pages), 1999. ISSN:

0018-9286.
DOI: 10.1109/9.763209.

Grigg, C. et al. "The IEEE reliability test system-1996. A report prepared by the reliability test system task force of the application of probability methods subcommittee". In: *IEEE Transactions on Power Systems*, pp. 1010–1020 (11 pages), 1999.

Lutterkort, D. and J. Peters. "Tight linear bounds on the distance between a spline and its b-spline control polygon". 1999.

Chen, H. and F. Allgöwer. "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability". In: *Automatica*, pp. 1205–1217 (13 pages), 1998.
PUBLISHED BY: Elsevier.

Kolmanovsky, I. and E. G. Gilbert. "Theory and computation of disturbance invariant sets for discrete-time linear systems". In: *Mathematical problems in engineering*, pp. 317–367 (51 pages), 1998.
PUBLISHED BY: Hindawi.

Kuhn, W. "Rigorously computed orbits of dynamical systems without the wrapping effect". In: *Computing*, pp. 47–67 (21 pages), 1998. ISSN: 0010-485X, 1436-5057.

LaValle, S. M. (1998). *Rapidly-exploring random trees: A new tool for path planning*. Tech. rep.

Mounier, H. and J. Rudolph. "Flatness-based control of nonlinear delay systems: A chemical reactor example". In: *International Journal of Control*, pp. 871–890 (20 pages), 1998.
PUBLISHED BY: Taylor & Francis.

Barraquand, J. et al. "A random sampling scheme for path planning". In: *The International Journal of Robotics Research*, pp. 759–774 (16 pages), 1997.
PUBLISHED BY: Sage Publications Sage CA: Thousand Oaks, CA.

Gil, J. C., A. Pedraza, M. Delgado, and H. Sira-Ramirez. "Flatness and passivity from a bond graph", pp. 1516–1521 (6 pages). In: *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*. 1997.

Butcher, J. C. and G. Wanner. "Runge-Kutta methods: some historical notes". In: *Applied Numerical Mathematics*, pp. 113–151 (39 pages), 1996.
PUBLISHED BY: Elsevier.

Fliess, M., J. Lévine, P. Martin, and P. Rouchon. "Flatness and defect of non-linear systems: introductory theory and examples". In: *International journal of control*, pp. 1327–1361 (35 pages), 1995.
PUBLISHED BY: Taylor & Francis.

Gawthrop, P. "Bicausal bond graphs". In: *SIMULATION SERIES*, pp. 83–83, 1994.
PUBLISHED BY: SCS SOCIETY FOR COMPUTER SIMULATION.

Manwell, J. and J. McGowan. "Lead acid battery storage model for hybrid energy systems". In: *Solar Energy*, pp. 399–405 (7 pages), 1993.

Manwell, J. F. and J. G. McGowan. "Lead acid battery storage model for hybrid energy systems". In: *Solar Energy*, pp. 399–405 (7 pages), 1993.
PUBLISHED BY: Elsevier.

Uebel, M., I. Minis, and K. Cleary. "Improved computed torque control for industrial robots", pp. 528–533 (6 pages). In: *Proceedings of the IEEE International Conference on Robotics and Automation*. 1992.

Boor, C. de. "Cutting corners always works". In: *Computer Aided Geometric Design*, pp. 125–131 (7 pages), 1987.
PUBLISHED BY: Elsevier.

Fletcher, R. *Practical Methods of Optimization*, 1987.
PUBLISHED BY: John Wiley\ & Sons, Inc.

Takagi, T. and M. Sugeno. "Fuzzy identification of systems and its applications to modeling and control". In: *IEEE transactions on systems, man, and cybernetics*, pp. 116–132 (17 pages), 1985.
PUBLISHED BY: IEEE.

Andersson, J. A. E., J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. "CasADi – A software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation*, In Press, 2018.