Flatness and Optimization in Robotic Motion Planning and Control

Ionela Prodan

Univ. Grenoble Alpes, Grenoble INP, LCIS, Valence, F-26009, France,
ionela.prodan@lcis.grenoble-inp.fr,
https://sites.google.com/site/iprodanionela/

April 1, 2026

Robotics Principia, GDR Robotic Spring School, March 30th - April 3rd 2026, Clermont Ferrand

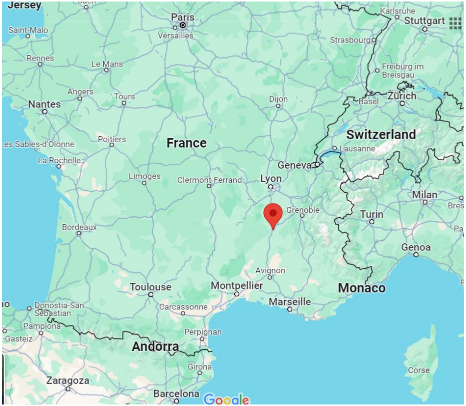
Grenoble INP - Univ. Grenoble Alpes

Graduate schools of Engineering and Management

- 8350 students, 38 laboratories, 465 partner companies
- in the Engineering and Technology area, Grenoble INP - UGA is ranked the 93rd position worldwide, the 5th position in France, i.e., **the 1st institution outside the Paris region**.

Grenoble INP - Esisar school and LCIS lab

- Engineering school in cybersecure intelligent systems
- 500 students, 20% international students
- Laboratory of Conception and Integration of Systems
  - $>$ 60 researchers in Computer Science, Electronics and Automation
  - research areas related to embedded and communicating systems.



Valence, Auvergne-Rhône-Alpes region, France



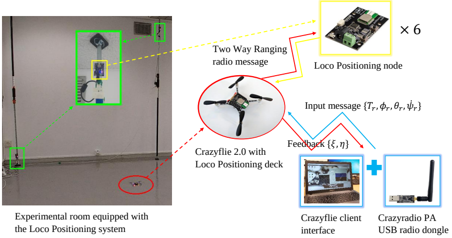


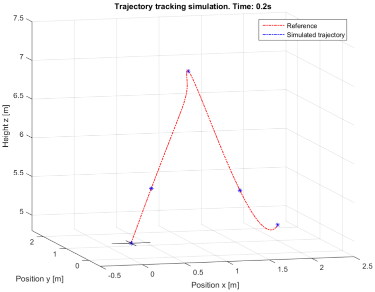

Valence, Auvergne-Rhône-Alpes region, France

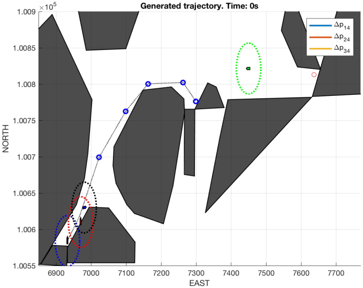
Research group:
PhD students: Vincent Marguet, Huu Thinh Do, Minh Tuan Dinh, Bogdan Gheorghe, Cong Khnah Dinh, Theodor Nicu, Duc Tri Vo, Huu Thien Nguyen
Master and intern students: Killian, Hugo, Trang, Hung

Quick overview of my research applications

Unmanned autonomous vehicles



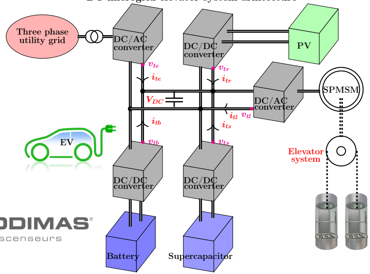




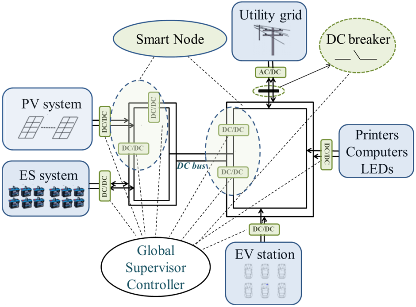
Quick overview of my research applications

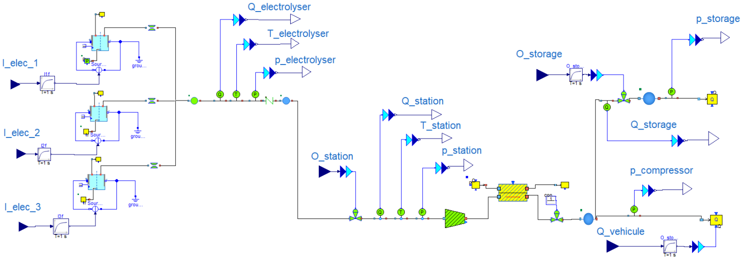
DC microgrids

DC microgrid elevator system arhitecture



SODIMAS ascenseurs

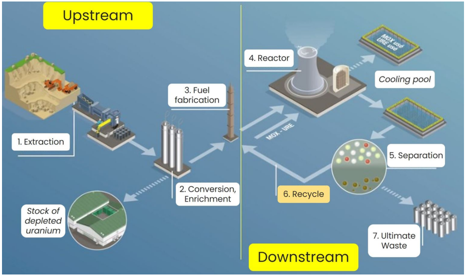




eDF

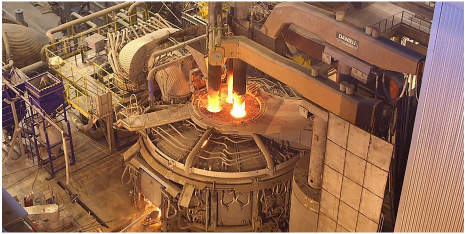
Quick overview of my research applications

Uranium Extraction-Scrubbing Operation in Spent Nuclear Fuel Treatment Process



U (95%) **Pu (1%)**

Control of a three-phase arc furnace producing Silico-manganese



Constrained control, Optimization, Flat mappings

Why optimization-based control?

Today's systems have complex architectures, are bounded by heterogeneous constraints and have to respect challenging operational costs. Optimization based approaches, which fully exploit changes in mission, initial condition and environment, are becoming ubiquitous.

How flat mappings can simplify the control of nonlinear systems?

Transforming them into an equivalent linear form simplifies the control design process.

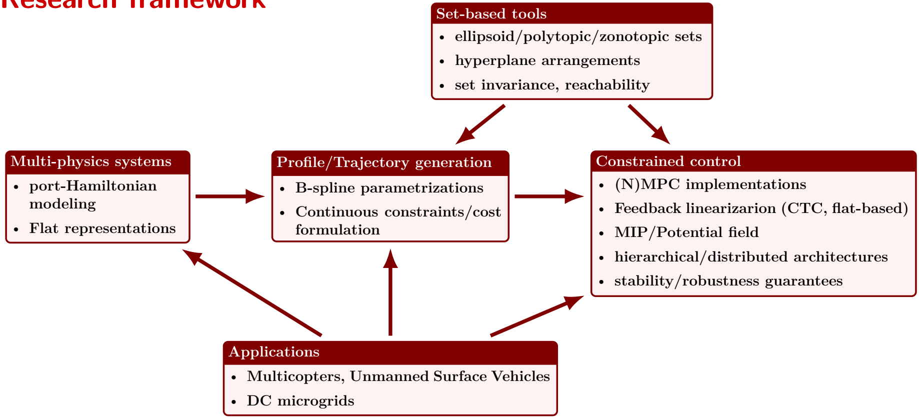
In the pursuit of certificates

Formal guarantees that constrained systems meet specified performance and safety criteria.

Outline

1. Research framework
   - Constrained profile generation via flatness
   - Challenges of convoluted constraints arising from flat mappings

Research framework



- Port Hamiltonian modeling (Maschke and Schaft, 1992), Differential flatness (Fliess et al., 1995), B-splines (Piegl and Tiller, Suryawan, De Doná, and Seron, 1995, 2011)
- Feedback linearization (Charlet, Lévine, and Marino, 1989, Banaszuk and Hauser, 1995, Devasia, Chen, and Paden, 1996, Formentin and Lovera, 2011, Mellinger and Kumar, 2011)
- Nonlinear Model Predictive Control (Mayne et al., 2000, Grüne and Pannek, 2011, Alamir, 2018, Pereira, Kolmanovsky, and Cesnik, 2022, Convens et al., 2024)

Nonlinear dynamics and its flatness property

Applying physics rules we obtain a first order differential equation representation for the system:

$$\dot{x}(t) = f(x(t), u(t))$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$.

Nonlinear dynamics and its flatness property

Applying physics rules we obtain a first order differential equation representation for the system:

$$\dot{x}(t) = f(x(t), u(t))$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$.

The system is differentially flat if there exist $z(t) \in \mathbb{R}^m$ s.t. the states and inputs can be algebraically expressed in terms of $z(t)$ and a finite number of its derivatives (Fliess et al., 1995):

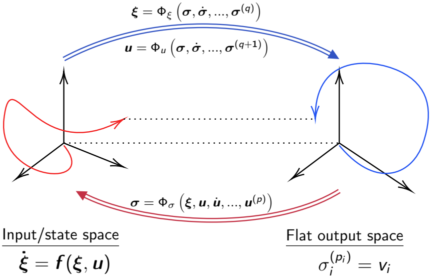
$$x(t) = \Phi_0(z(t), \dot{z}(t), \cdots, z^{(q)}(t)),$$
$$u(t) = \Phi_1(z(t), \dot{z}(t), \cdots, z^{(q+1)}(t)),$$

where

$$z(t) = \gamma(x(t), u(t), \dot{u}(t), \cdots, u^{(q)}(t)).$$

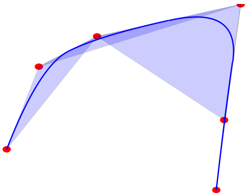
Properties:

- elimination of the differential equations describing the system dynamics;
- reduction of the number of considered variables.



From the control viewpoint, expressing the input in terms of the flat output is a model inversion.

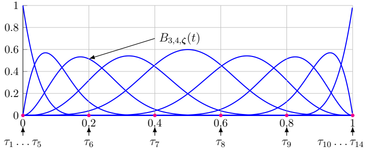
Bspline functions and their properties



Flat output parametrization:

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

with $\quad B_{\ell,d,\zeta}(t) = \frac{t-\tau_\ell}{\tau_{\ell+d}-\tau_\ell} B_{\ell,d-1,\zeta}(t) + \frac{\tau_{\ell+d+1}-t}{\tau_{\ell+d+1}-\tau_{\ell+1}} B_{\ell+1,d-1,\zeta}(t)$



Properties [Lyche et al. 2018]:

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

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

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

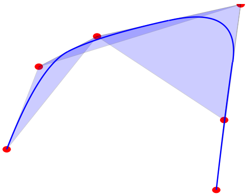
$$\sum_{i=\ell-d}^{\ell} B_{i,d,\zeta}(t) = 1, \; \forall t \in [\tau_\ell, \tau_{\ell+1})$$

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

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

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

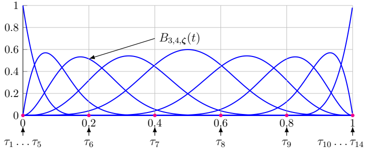
Bspline functions and their properties



Flat output parametrization:

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

with $\quad B_{\ell,d,\zeta}(t) = \frac{t-\tau_\ell}{\tau_{\ell+d}-\tau_\ell} B_{\ell,d-1,\zeta}(t) + \frac{\tau_{\ell+d+1}-t}{\tau_{\ell+d+1}-\tau_{\ell+1}} B_{\ell+1,d-1,\zeta}(t)$



Properties [Lyche et al. 2018]:

P1) Local support:

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

P2) Global partition of unity:

$$\sum_{i=1}^{n} B_{i,d,\zeta}(t) = 1,\ \forall t \in [\tau_{d+1}, \tau_{n+1})$$

P3) Global convexity:

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

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

B-spline approximations and knot refinements

How can the complexity of the B-spline curve be handled[1]?

[1] Vincent Marguet, Cong Khanh Dinh, Florin Stoican and Ionela Prodan: Indoor formation motion planning using B-splines parametrization and evolutionary optimization. *Control Engineering Practice*, 152 (2024): 106066.

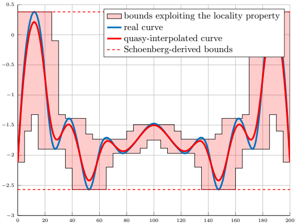
B-spline approximations and knot refinements

How can the complexity of the B-spline curve be handled[1]?

Idea

Quasy-interpolants such as the Schoenberg operator provide simple approximations of the curve Lyche, Manni, and Speleers, 2018:

$$\tilde{f}(t) = \sum_{i=1}^{n} f(\xi^{\star}_{i,d,\xi})\tilde{B}_{i,d}(t)$$

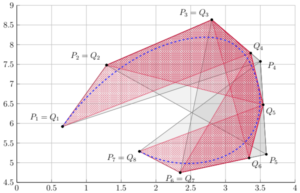


- still nonlinear in the control points but fewer constraints
- the approximation is close to the curve, but the interpolation error bounds are conservative

Idea

Knot refinement ensures tighter approximation of the curve:

$$Q_i = \alpha_i P_i + (1 - \alpha_i)P_{i-1}$$



- the number of variables within the optimization problem remains unchanged (Q_i depend linearly on P_i)
- the convex hull approximation becomes tighter (more constraints, but this is acceptable)

[1] Vincent Marguet, Cong Khanh Dinh, Florin Stoican and Ionela Prodan: Indoor formation motion planning using B-splines parametrization and evolutionary optimization. *Control Engineering Practice*, 152 (2024): 106066.

Flat systems - Applications & key idea

Aerial vehicles

Monocopter

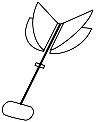
Multicopter

Fixed-wing UAV

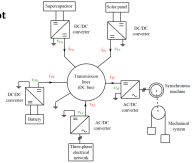
VTOL Tailsitter

Car-like robot/manipulator

Flapping-wing robot



Microgrid elevator

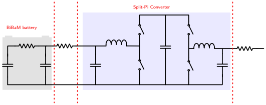


Spacecraft model

Quadruped robot

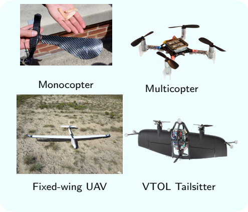
Offshore Crane

Battery & Split-pi converter



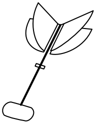
Flat systems - Applications & key idea

Aerial vehicles

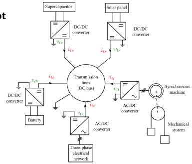


Car-like robot/manipulator

Flapping-wing robot



Microgrid elevator

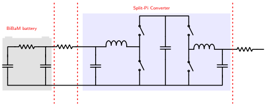


Spacecraft model

Quadruped robot

Offshore Crane

Battery & Split-pi converter



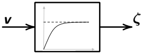
$$\dot{\boldsymbol{x}} = \mathrm{f}(\boldsymbol{x}, \boldsymbol{u})$$

Nonlinear system

Dynamic feedback linearization

Flat representation

Flat mappings

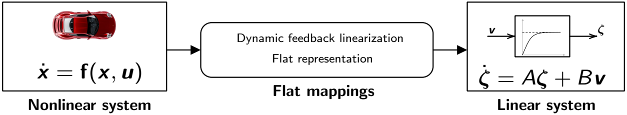


$$\dot{\zeta} = A\zeta + B\boldsymbol{v}$$

Linear system

The price for linearization: convoluted constraint

The system to control now is **linear**...



The price for linearization: convoluted constraint

The system to control now is **linear**...

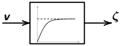
$\dot{\boldsymbol{x}} = \mathbf{f}(\boldsymbol{x}, \boldsymbol{u})$

Nonlinear system

Dynamic feedback linearization

Flat representation

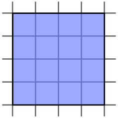
Flat mappings



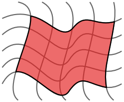
$\dot{\zeta} = A\zeta + B\boldsymbol{v}$

Linear system

$(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{B}$



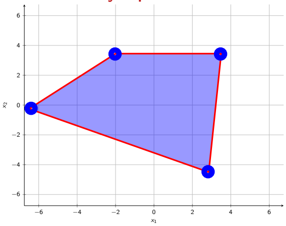
$(\zeta, \boldsymbol{v}) \in \mathcal{V}$



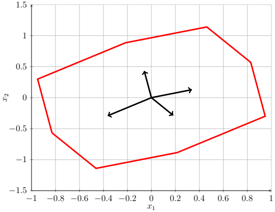
but the constraints become **convoluted**!

Brief overview on the geometric tools

Polytopic sets



Zonotopic sets



- have a dual representation:
  - half-space:

    $$\{x \in \mathbb{R}^d : h_i x \leq k_i,\ i = 1 \dots N_h\}$$

  - vertex:

    $$x = \{\sum_i \alpha_i v_i, \alpha_i \geq 0, \sum_i \alpha_i = 1, i = 1 \dots N_v\}$$

- efficient algorithms for set containment problems Gritzmann and Klee, 1994
- can approximate any convex shape Bronstein, 2008

- are a Minkowski sum of generators:

$$\mathcal{Z}(G, c) = \{x \in \mathbb{R}^d : x = c + \sum_{k=1}^{m} \xi_k g_k, |\xi_k| \leq 1\}$$

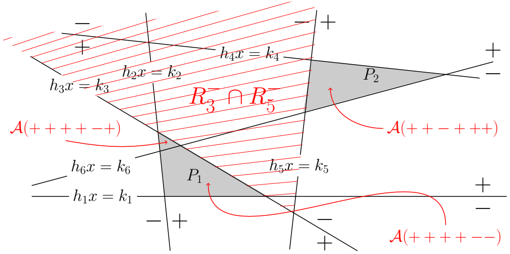
$$= \{c\} \oplus G\mathbb{B}^m_\infty.$$

- their volume has an explicit formulation Gover and Krikorian, 2010:

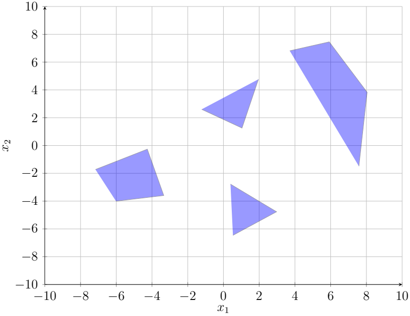
$$\mathrm{Vol}(\mathcal{Z}(G, c)) = \sum_{1 \leq j_1 < j_2 \dots j_d \leq m} \left|\det(G^{j_1 \dots j_d})\right|$$

Brief overview on the geometric tools

Hyperplane arrangements



How to safely and efficiently navigate in a cluttered environment?



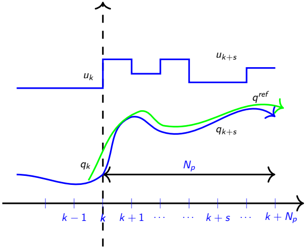
- a collection of hyperplanes $\mathcal{H}_i$ cuts the space into a collection of disjoint, convex cells Geyer, Torrisi, and Morari, 2008, Stanley et al., 2004:

$$\mathcal{A}(\mathbb{H}) = \bigcup_{l=1\dots\gamma(N)} \mathcal{A}(\sigma_l), \qquad \mathcal{A}(\sigma) = \bigcap_{i\in\mathcal{I}} \mathcal{R}_i^{\sigma(i)}$$

- each cell is uniquely described by a sign combination $\sigma \in \{-,+\}^N$ (to select the half-space in which it lies)
- the cells are classified into admissible and forbidden Prodan et al., 2015.

Constrained optimization-based control

- Prediction model and constraints handling
- Implicit (on-line) vs. explicit (off-line) implementation
- Distributed/ hierarchical implementations



model of the system

reference trajectory

$$\arg\min_{\tilde{u}} \; V_f(x(k+N_p), q(k+N_p)) + \sum_{s=1}^{N_p-1} V_n(x(k+s), u(k+s), q(k+s))$$

$$\text{subject to:} \begin{cases} x(k+s+1) = f(x(k+s), u(k+s)), & s = 0 : N_p - 1, \\ h(x(k+s), u(k+s), q(k+s)) \leq 0, & s = 1 : N_p - 1. \end{cases}$$

constraints

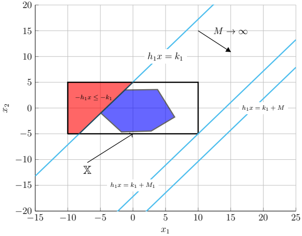
optimization solver

Propoi, 1963, Cutler et al., 2007, Richalet and O'Donovan, 2009, Rawlings and Mayne, 2009

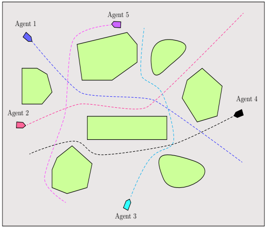
Constrained optimization-based control

- Non-convex constraints handling through direct and indirect methods.

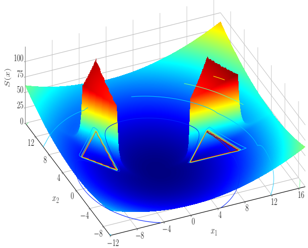
Mixed-Integer Programming (MIP)
Jünger et al., 2009



Collision avoidance applications



Artificial Potential Field (AFP)
Khatib, 1986



model of the system · **reference trajectory** · **constraints** · **optimization solver**

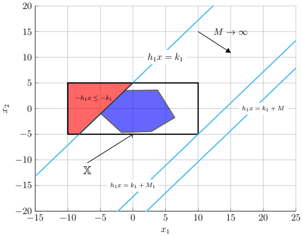
$$\arg\min_{\tilde{u}} \; V_f(x(k+N_p), q(k+N_p)) + \sum_{s=1}^{N_p-1} V_n(x(k+s), u(k+s), q(k+s))$$

$$\text{subject to: } \begin{cases} x(k+s+1) = f(x(k+s), u(k+s)), & s = 0 : N_p - 1, \\ h(x(k+s), u(k+s), q(k+s)) \leq 0, & s = 1 : N_p - 1. \end{cases}$$

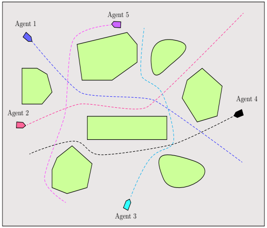
Constrained optimization-based control

- Non-convex constraints handling through direct and indirect methods.

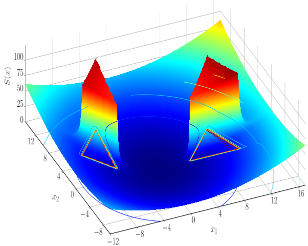
Mixed-Integer Programming (MIP)
Jünger et al., 2009



Collision avoidance applications



Artificial Potential Field (AFP)
Khatib, 1986



model of the system — **reference trajectory** — **constraints** — **optimization solver**

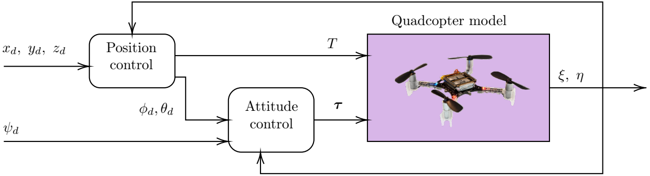
$$\arg\min_{\tilde{u}} \; V_f(x(k+N_p), q(k+N_p)) + \sum_{s=1}^{N_p-1} V_n(x(k+s), u(k+s), q(k+s))$$

$$\text{subject to:} \begin{cases} x(k+s+1) = f(x(k+s), u(k+s)), & s = 0 : N_p - 1, \\ h(x(k+s), u(k+s), q(k+s)) \le 0, & s = 1 : N_p - 1. \end{cases}$$

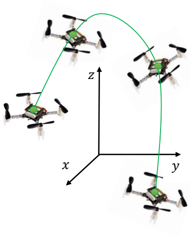
Constrained profile generation via flatness and B-splines

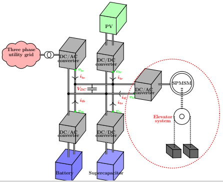
Connectivity maintenance and formation control

Communication-induced trajectories for multiple drones



Applications:





minimize Cost function (energy/time/length):

such that
- model (dynamics and system limitations)
- stay in the defined area
- waypoint constraints
- communication constraints
- other state/input constraints

Idea

Flat representation ⇒ Reformulation of constraints and cost ⇒
⇒ B-spline parametrization and know refinement ⇒
⇒ Differential evolution ⇒ Profile generation

Optimization problem for constrained trajectories generation

Multicopters trajectories generation:

$$\min_{\mathbf{z}(t)} \quad \int_0^{T_f} \mathcal{E}(\mathbf{z}(t))\mathrm{d}t \tag{1a}$$

$$\text{s.t.} \quad \mathbf{z}_i(t) \subset \mathbf{K}_p,\ 0 \le t \le T_f,\ \forall i, \tag{1b}$$

$$v_i(t) \le v_{max},\ 0 \le t \le T_f,\ \forall i, \tag{1c}$$

$$|\phi_i(t)| \le \epsilon,\ |\theta_i(t)| \le \epsilon,\ 0 \le t \le T_f,\ \forall i, \tag{1d}$$

$$T_{min} \le T_i(t) \le T_{max},\ 0 \le t \le T_f,\ \forall i, \tag{1e}$$

$$|p_i(t)| \le \omega_{max},\ |q_i(t)| \le \omega_{max},\ 0 \le t \le T_f, \forall i, \tag{1f}$$

$$\mathbf{p}_i(0) = \mathbf{p}_0^i, \mathbf{v}_i(0) = \mathbf{0}_{3\times 1},\ \mathbf{a}_i(0) = \mathbf{0}_{3\times 1},\ \forall i, \tag{1g}$$

$$\mathbf{p}_i(T_f) = \mathbf{p}_f^i, \mathbf{v}_i(T_f) = \mathbf{0}_{3\times 1},\ \mathbf{a}_i(T_f) = \mathbf{0}_{3\times 1},\ \forall i, \tag{1h}$$

$$\|\mathbf{z}_1(t_\ell) - \mathbf{p}_\ell^{wp}\|_2 \le d_{wp},\ \ell = 1, \ldots, N_{WP}, \tag{1i}$$

$$\mathbf{z}_j(t) - \mathbf{z}_i(t) = \Delta^{ij,h},\ t_0^h \le t \le t_f^h,\ \forall i, j, h \tag{1j}$$

$$d_{ij}(t) \le \rho,\ \forall (i,j) \in \mathcal{L},\ 0 \le t \le T_f, \tag{1k}$$

$$\mathbf{z}_i(t) \in C(\cup_{l=1}^{N_o} \mathcal{O}_l),\ 0 \le t \le T_f,\ \forall i, \tag{1l}$$

Cost (energy, time, length...)
Position constraint
Velocity constraint
Angular constraint
Thrust constraint
Angular velocity constraint
Initial conditions
Final conditions
Way-point constraint
Formation constraint
Communication constraint
Obstacle avoidance

Differential evolution is employed for solving the nonconvex optimization problem.

[1] Marguet, Vincent, Cong Khanh Dinh, Florin Stoican, and Ionela Prodan. "Indoor formation motion planning using B-splines parametrization and evolutionary optimization." Control Engineering Practice 152 (2024): 106066.

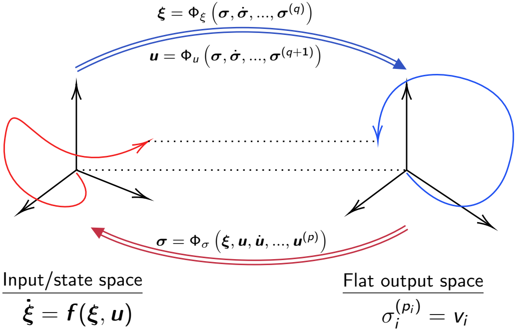
How to handle the intricate challenges of convoluted constraints arising from flat mappings?

A multicopter application

Model Predictive Control for multicopter navigation

	Reference	Remarks
Position control	Nguyen, Prodan, and Lefevre, 2019	NMPC with FL local controller, computationally complex
	Gomaa et al., 2022	fast NMPC with stability, feasibility guaranteed, strict tuning
	Mueller and D'Andrea, Do and Prodan, Lu et al., 2013, 2023, 2017	QP-MPC in the flat output space, with convoluted constraint approximation.
	Liu, Lu, and Chen, 2015	*Explicit* solution (PWA) of Bézier curves, no stability analysis and constraint characterization
Attitude control	Jiajin et al., 2017	*Explicit* MPC; Roll, pitch angles ≈ 0
	Nguyen, Prodan, and Lefèvre, 2020	NMPC, stability and feasibility guaranteed, conservative constraints characterization with CTC local controller
	Zanelli et al., 2018	Low-cost NMPC, suboptimal formulation, hovering point approximation
Full dynamics	Wang et al., 2021	Roll, pitch angles ≈ 0; Soft constraint
	Torrente et al., 2021	Learning MPC, robust disturbance rejection, lack of theoretical guarantees
	Cohen, Abdulrahim, and Forbes, 2020	Finite horizon LQR for the approximated dynamics, constraints neglected

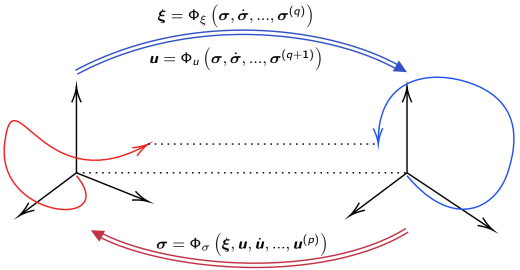
Differential flatness[2] - a linearization tool



- Elimination of the differential equations.
- Reduction of the number of considered variables.
- Inverse dynamic: expression of the input in terms of the flat output.

[2] Fliess, M., Lévine, J., Martin, P., & Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. International journal of control, 61(6), 1327-1361.

Differential flatness[2] - a linearization tool



Input/state space

$$\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u})$$

Flat output space

$$\sigma_i^{(p_i)} = v_i$$

- Elimination of the differential equations.
- Reduction of the number of considered variables.
- Inverse dynamic: expression of the input in terms of the flat output.

$\dot{\boldsymbol{\xi}} = f(\boldsymbol{\xi}, \boldsymbol{u})$

Constraints : $\boldsymbol{u} \in \mathcal{U}$

Approximated via Taylor expansion →

$\dot{\boldsymbol{\xi}} \approx A_{approx}\boldsymbol{\xi} + B_{approx}\boldsymbol{u}$

Constraint : $\boldsymbol{u} \in \mathcal{U}$

Simple constraints
but sensitive to uncertainties
due to model approximation

Coordinate change $(\boldsymbol{\xi}, \boldsymbol{u}) = \Phi(\boldsymbol{\sigma}, \boldsymbol{v})$ →

$\dot{\boldsymbol{\sigma}} = A\boldsymbol{\sigma} + B\boldsymbol{v}$

Constraint : $(\boldsymbol{\sigma}, \boldsymbol{v}) \in \mathcal{V}$

Precisely handle the model
but the constraints will be complicated

[2] Fliess, M., Lévine, J., Martin, P., & Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. International journal of control, 61(6), 1327-1361.

Constraint deformation: an eternal rival of feedback linearization

- In general, feedback linearization will alter the constraints' description in a nonlinear and convoluted manner.
- The main challenge resides on the constraint satisfaction not only for the new input but also for the new state variables.

Input/state space

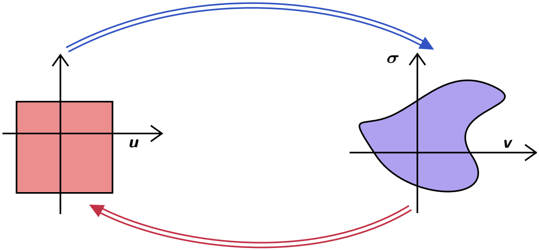
$$\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u})$$

$\boldsymbol{u} \in \mathcal{U}$

Flat output space

$$\dot{\boldsymbol{\sigma}} = A\boldsymbol{\sigma} + B\boldsymbol{v}$$

$(\boldsymbol{\sigma}, \boldsymbol{v}) \in \mathcal{V}$



Constraint deformation: Quadcopter position control case

Particularly for the quadcopter position control problem, the new nonlinear constraints involve only the new input in the flat output space.

Input/state space

$$\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}, \boldsymbol{u})$$

$$\boldsymbol{u} \in \mathcal{U}$$

Flat output space

$$\ddot{\boldsymbol{\sigma}} = \boldsymbol{v}$$

$$\boldsymbol{v} \in \mathcal{V}_c$$

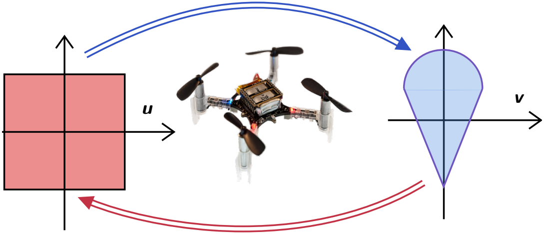


Figure 1: Constraint deformation due to the flatness-based coordinate change

How to exploit the linearized dynamics and tackle the distorted constraint at the same time?

Constraint deformation: Quadcopter position control case

Particularly for the quadcopter position control problem, the new nonlinear constraints involve only the new input in the flat output space.

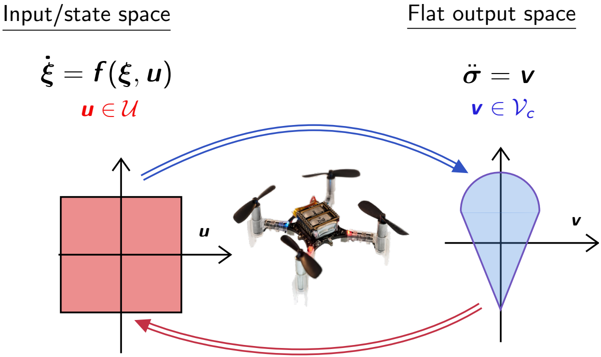


Figure 1: Constraint deformation due to the flatness-based coordinate change

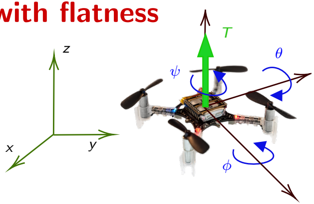
How to exploit the linearized dynamics and tackle the distorted constraint at the same time?

⇒ Model Predictive Control: a constraint-handling framework.

Quadcopter's position control problem with flatness

Consider the translational dynamics of a quadcopter:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} T(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\ T(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\ T\cos\phi\cos\theta - g \end{bmatrix},$$



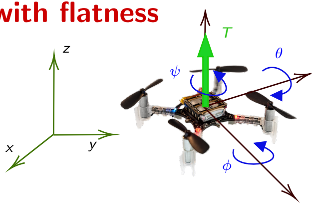
$$\boldsymbol{u} = [T \;\; \phi \;\; \theta]^\top \in \mathcal{U} \triangleq \{0 \leq T \leq T_{\max}, |\phi| \leq \phi_{max}, |\theta| \leq \theta_{max}\}$$

where x, y, z: position components of the drone; T: the thrust provided by the propellers; ϕ, θ, ψ: the roll, pitch and yaw angles of the drone, respectively.

Quadcopter's position control problem with flatness

Consider the translational dynamics of a quadcopter:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}T(\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi)\\ T(\cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi)\\ T\cos\phi\cos\theta-g\end{bmatrix},$$



$$\boldsymbol{u}=[T\ \ \phi\ \ \theta]^\top \in \mathcal{U} \triangleq \{0\leq T\leq T_{\max}\ , |\phi|\leq \phi_{max}, |\theta|\leq \theta_{max}\}$$

where x, y, z: position components of the drone; T: the thrust provided by the propellers; ϕ, θ, ψ: the roll, pitch and yaw angles of the drone, respectively.
With the flat output $\boldsymbol{\sigma}=[\sigma_1,\sigma_2,\sigma_3]^\top \triangleq [x, y, z]^\top$, we have

The flat representation:

$$x=\sigma_1, y=\sigma_2, z=\sigma_3,$$
$$T=\sqrt{\ddot{\sigma}_1^2+\ddot{\sigma}_2^2+(\ddot{\sigma}_3+g)^2},$$
$$\phi=\arcsin\left((\ddot{\sigma}_1\sin\psi-\ddot{\sigma}_2\cos\psi)/T\right),$$
$$\theta=\arctan\left((\ddot{\sigma}_1\cos\psi+\ddot{\sigma}_2\sin\psi)/(\ddot{\sigma}_3+g)\right),$$

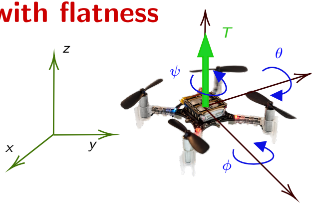
The linearizing law:

$$\boldsymbol{u}=\varphi_\psi(\boldsymbol{v})$$
$$\triangleq \begin{cases} T &= \sqrt{v_1^2+v_2^2+(v_3+g)^2},\\ \phi &= \arcsin\left((v_1\sin\psi-v_2\cos\psi)/T\right),\\ \theta &= \arctan\left((v_1\cos\psi+v_2\sin\psi)/(v_3+g)\right),\end{cases}$$

Quadcopter's position control problem with flatness

Consider the translational dynamics of a quadcopter:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix} T(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\ T(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\ T\cos\phi\cos\theta - g\end{bmatrix},$$



$$\boldsymbol{u} = [T \;\; \phi \;\; \theta]^\top \in \mathcal{U} \triangleq \{0 \le T \le T_{\max}, |\phi| \le \phi_{max}, |\theta| \le \theta_{max}\}$$

where x, y, z: position components of the drone; T: the thrust provided by the propellers; ϕ, θ, ψ: the roll, pitch and yaw angles of the drone, respectively.
With the flat output $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \sigma_3]^\top \triangleq [x, y, z]^\top$, we have

The flat representation:

$$x = \sigma_1, y = \sigma_2, z = \sigma_3,$$

$$T = \sqrt{\ddot{\sigma}_1^2 + \ddot{\sigma}_2^2 + (\ddot{\sigma}_3 + g)^2},$$

$$\phi = \arcsin\left((\ddot{\sigma}_1 \sin\psi - \ddot{\sigma}_2\cos\psi)/T\right),$$

$$\theta = \arctan\left((\ddot{\sigma}_1\cos\psi + \ddot{\sigma}_2\sin\psi)/(\ddot{\sigma}_3 + g)\right),$$

The linearizing law:

$$\boldsymbol{u} = \varphi_\psi(\boldsymbol{v}) \triangleq \begin{cases} T &= \sqrt{v_1^2 + v_2^2 + (v_3 + g)^2}, \\ \phi &= \arcsin\left((v_1\sin\psi - v_2\cos\psi)/T\right), \\ \theta &= \arctan\left((v_1\cos\psi + v_2\sin\psi)/(v_3 + g)\right), \end{cases}$$

The linearized dynamics with the new input set $\mathcal{V}$:

$$\ddot{\sigma}_1 = v_1, \ddot{\sigma}_2 = v_2, \ddot{\sigma}_3 = v_3,$$

$$\boldsymbol{v} = [v_1, v_2, v_3]^\top \in \mathcal{V} \triangleq \{\boldsymbol{v} \in \mathbb{R}^3 : \varphi_\psi(\boldsymbol{v}) \in \mathcal{U}\}$$

Input constraint sets in the flat output space

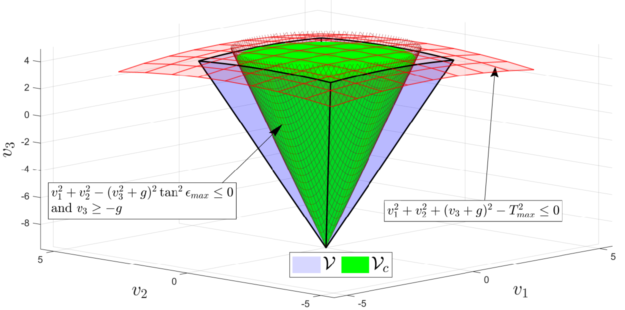
$$\mathcal{V} \triangleq \left\{ \boldsymbol{v} : \begin{bmatrix} 0 \\ -\phi_{max} \\ -\theta_{max} \end{bmatrix} \leq \begin{bmatrix} \sqrt{v_1^2 + v_2^2 + (v_3 + g)^2} \\ \arcsin\left((v_1 \sin\psi - v_2 \cos\psi)/T\right) \\ \arctan\left((v_1 \cos\psi + v_2 \sin\psi)/(v_3 + g)\right) \end{bmatrix} \leq \begin{bmatrix} T_{max} \\ \phi_{max} \\ \theta_{max} \end{bmatrix} \right\}$$

The constraint set $\mathcal{V}$ has the following disadvantages[3]:

- Non-convexity;
- ψ-dependence, which means $\mathcal{V}$ is time varying during the drone's navigation.

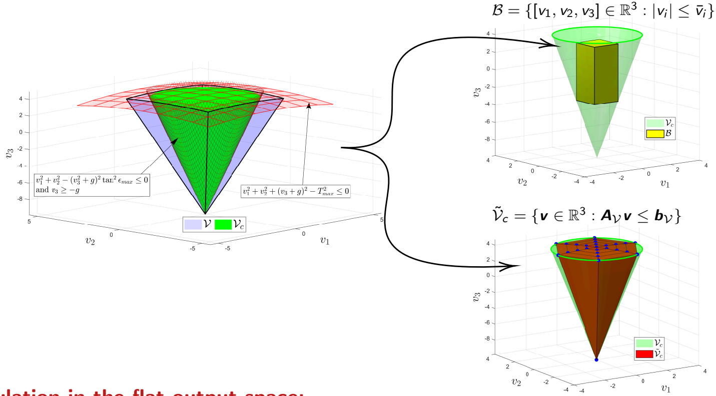
Proposition: Consider a convex subset of $\mathcal{V}$, denoted as $\mathcal{V}_c$:

$$\mathcal{V}_c = \left\{ \boldsymbol{v} \in \mathbb{R}^3 : \begin{bmatrix} v_1^2 + v_2^2 + (v_3 + g)^2 - T_{max}^2 \\ v_1^2 + v_2^2 - (v_3^2 + g)^2 \tan^2 \epsilon_{max} \end{bmatrix} \leq 0, \epsilon_{max} \triangleq \min(\theta_{max}, \phi_{max}) \text{ and } v_3 \geq -g \right\}$$



[3] Nguyen, N. T., Prodan, I., & Lefèvre, L. (2018, June). Effective angular constrained trajectory generation for thrust-propelled vehicles. In 2018 European Control Conference (pp. 1833-1838). IEEE.

Quadratic Program MPC with the input polytopic approximations



MPC formulation in the flat output space:

$$\underset{\boldsymbol{v}_k, \dots, \boldsymbol{v}_{k+N_p-1}}{\arg\min} \sum_{i=0}^{N_p-1} \left\|\boldsymbol{\xi}_{i+k}\right\|_Q^2 + \|\boldsymbol{v}_{i+k}\|_R^2 + \left\|\boldsymbol{\xi}_{k+N_p}\right\|_P^2$$

$$\text{s.t}: \boldsymbol{\xi}_{i+k+1} = A\boldsymbol{\xi}_{i+k} + B\boldsymbol{v}_{i+k}; \boldsymbol{\xi}_{k+N_p} \in \mathcal{X}_f; \boldsymbol{v}_{i+k} \in \tilde{\mathcal{V}}_c/\mathcal{B},\ i \in \{0, 1, \dots, N_p - 1\}$$

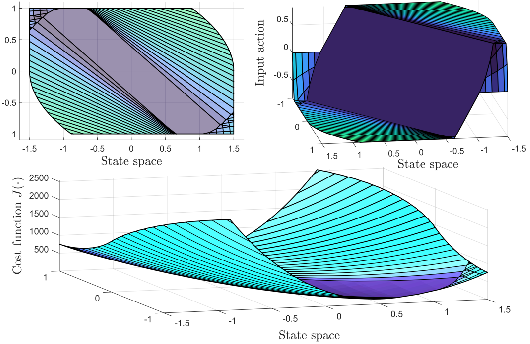
where $\boldsymbol{\xi}_k = [\boldsymbol{\sigma}_k,\ \dot{\boldsymbol{\sigma}}_k]^\top \in \mathbb{R}^6$, $\boldsymbol{v}_k = [v_{1,k},\ v_{2,k},\ v_{3,k}]^\top \in \mathbb{R}^3$ denotes the state vector, and the new input at time step k.

Explicit MPC - problem statement

Denote $\boldsymbol{v}^*(\cdot|\boldsymbol{\xi}_k)$ as the optimal sequence of $\boldsymbol{v}_{i+k}$ in the MPC problem, then its solution can be represented as a piece-wise function of $\boldsymbol{\xi}_k$ as follows Bemporad et al., 2002:

$$\underset{j\in\{1,\dots,M\}}{\left(\boldsymbol{v}^*(\cdot|\boldsymbol{\xi}_k); J^*(\boldsymbol{\xi}_k)\right)} = \begin{cases} \left(F_1\boldsymbol{\xi}_k + \mu_1; \boldsymbol{\xi}_k^\top \gamma_1 \boldsymbol{\xi}_k + \alpha_1 \boldsymbol{\xi}_k + \beta_1\right) & , \boldsymbol{\xi}_k \in \mathcal{R}_1; \\ & \dots \\ \left(F_M\boldsymbol{\xi}_k + \mu_M; \boldsymbol{\xi}_k^\top \gamma_M \boldsymbol{\xi}_k + \alpha_M \boldsymbol{\xi}_k + \beta_M\right) & , \boldsymbol{\xi}_k \in \mathcal{R}_M. \end{cases} \quad (2)$$

where $\mathcal{R}_j = \{\boldsymbol{\xi} : H_j\boldsymbol{\xi} \leq h_j\}$, the j-th polyhedral critical region, together with the parameters $F_j, \mu_j, \gamma_j, \alpha_j, \beta_j$ are numerically available within the parametric programming framework Borrelli, Takács et al., Zeilinger, Jones, and Morari, 2003, 2016, 2011.



Explicit MPC - the first simulation test

Table 1: Simulation result with Explicit MPC

System size	Prediction horizon	Number of critical regions	Offline computation time	Avg. online computation time	Storage size
6D	2	49897	≈ 24 hours	**1318.5 (ms)**	96 MB

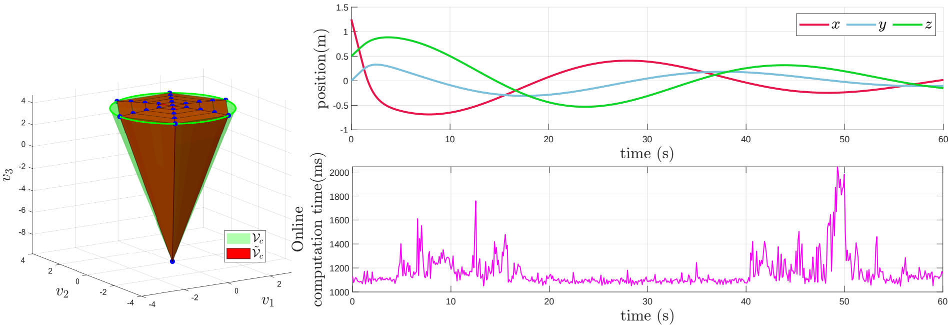


Figure 3: Simulation result with the explicit MPC

Explicit MPC - problem decoupling

Taking advantage of the three independent double integrators, we consider a set $\mathcal{B}$ as:

$$\mathcal{B} = \{[v_1, v_2, v_3] \in \mathbb{R}^3 : |v_i| \leq \bar{v}_i\}$$

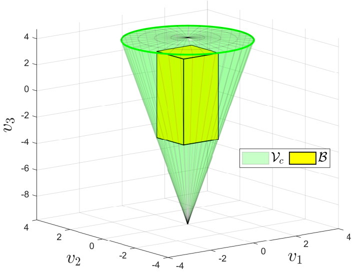


Figure 4: Largest box $\mathcal{B}$ inside $\mathcal{V}_c$

$$\ddot{\sigma}_i = v_i \text{ s.t } |v_i| \leq \bar{v}_i, i \in \{1, 2, 3\}$$

Table 2: Parameters for simulations and experimental tests

Parameters	Values
T_{max}	$1.45g \approx 14.22$ m/s^2
ϵ_{max}	0.1745 rad (10^o)
Sampling time t_s	100 ms
$\bar{v}_1, \bar{v}_2, \bar{v}_3$	0.8154,0.8154,3.27

Three double integrators with three independent constraints!

Simulation results

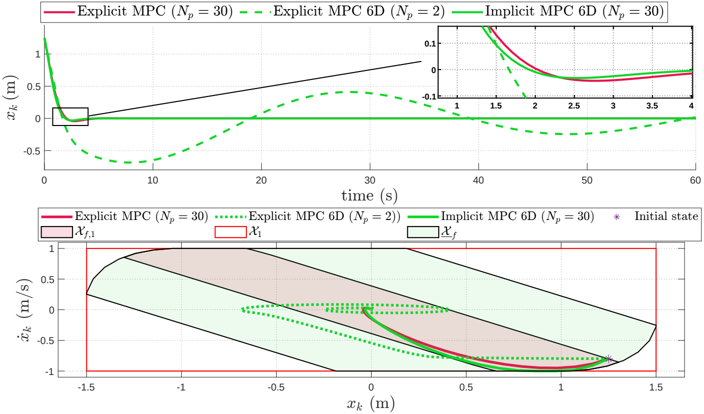
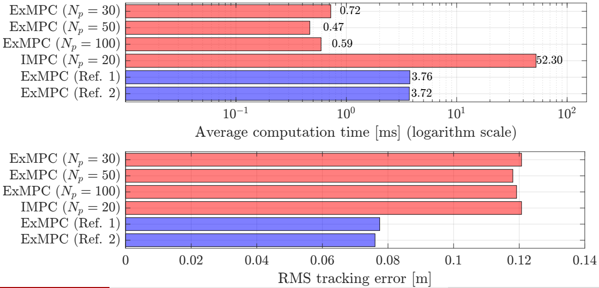


Figure 5: Stabilization with different MPC schemes.

System size	Prediction horizon	Number of critical regions	Computation time	Avg. online computation time	Storage size
6D	2	49897	$\approx$ 24 hours	**1318.5 (ms)**	96 MB
6D (Implicit)	30	-	-	**11.63 (ms)**	-
$(2D)^3$	30	(103,103,11)	$\leq$ 5 minutes	**0.3385 (ms)**	0.11 MB

Experimental results

Figure 6: Experiment video with the Crazyflie 2.1 platform



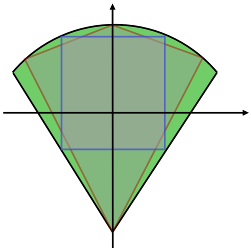
How to handle the intricate challenges of convoluted constraints arising from flat mappings?

MILP representation of a ReLU-ANN

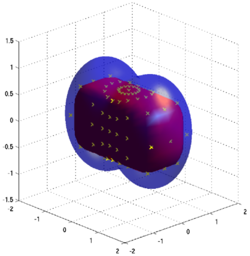
Flatness-based control

Flatness-based control includes coordinate change to *linearize the system in closed-loop*. The convoluted constraints in the new coordinates are usually handled:

- offline with constrained trajectory generation Mellinger and Kumar, Houari et al., Diwold, Kolar, and Schöberl, 2011, 2011, 2022.
- online with *convex inner approximations* such as hyperbox Greeff and Schoellig, Nguyen, Prodan, and Lefèvre, 2018, 2020, polytopes Do and Prodan, Faiz, Agrawal, and Murray, 2023, 2001, or super-ellipsoids Morio et al., 2008 for particular applications.



Greeff, M., & Schoellig, A. P. (2018)[1]
Do, H. T., & Prodan, I. (2023)[2]



Morio, V. (2009)[3]

[1] Greeff, M., & Schoellig, A. P. (2018). Flatness-based model predictive control for quadrotor trajectory tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE.

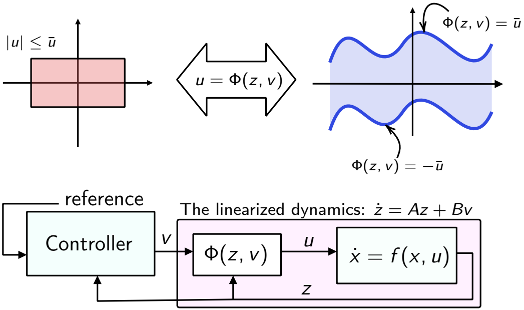
[2] Do, H. T., & Prodan, I. (2023). Indoor experimental validation of MPC-based trajectory tracking for a quadcopter via a flat mapping approach. *In 2023 European Control Conference*, IEEE.

Convoluted constraints by feedback linearization

Differentially flat systems Fliess et al., 1995

A system is differentially flat if and only if it is endogenous dynamic feedback linearizable.

$$\begin{cases} \dot{x} = f(x, u), \\ |u| \leq \bar{u} \end{cases} \qquad \begin{cases} \dot{z} = Az + Bv, \\ |\Phi(z, v)| \leq \bar{u} \end{cases}$$



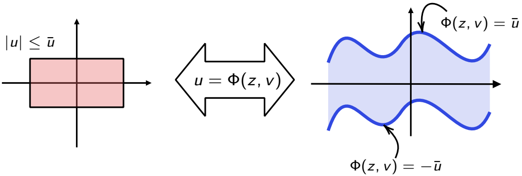
Fliess, M., Lévine, J., Martin, P., & Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, 61(6), 1327-1361.

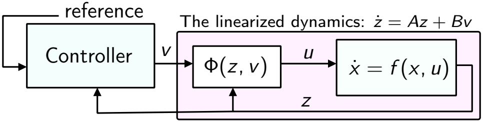
Convoluted constraints by feedback linearization

Differentially flat systems Fliess et al., 1995

A system is differentially flat if and only if it is endogenous dynamic feedback linearizable.

$$\begin{cases} \dot{x} = f(x,u), \\ |u| \leq \bar{u} \end{cases} \qquad \begin{cases} \dot{z} = Az + Bv, \\ |\Phi(z,v)| \leq \bar{u} \end{cases}$$

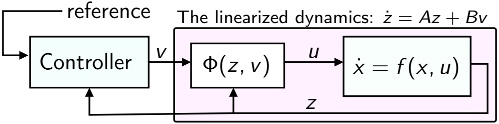




How to *approximate* the convoluted constraint and keeping the online control algorithm *tractable*?

Fliess, M., Lévine, J., Martin, P., & Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, 61(6), 1327-1361.

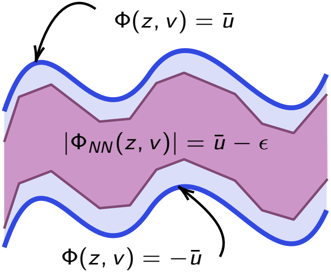
Our general idea for neural network-based inner approximation



Constraint: $|u| \leq \bar{u}$

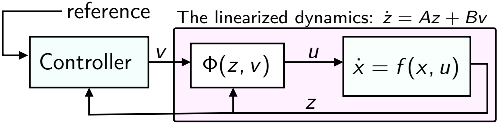
- Approximate the mapping $\Phi(z, v)$ by a neural network $\Phi_{NN}(z, v)$;
- Suppose that: $\sup |\Phi(z, v) - \Phi_{NN}(z, v)| \leq \epsilon$, impose the constraint:

$$|\Phi_{NN}(z, v)| \leq \bar{u} - \epsilon. \qquad (\star)$$



[4] Huu-Thinh Do and Ionela Prodan. On the constrained feedback linearization control based on the MILP representation of a ReLU-ANN, accepted to *IEEE Control Systems Letters*, 2024.

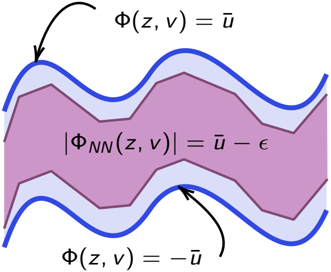
Our general idea for neural network-based inner approximation



Constraint: $|u| \leq \bar{u}$

- Approximate the mapping $\Phi(z, v)$ by a neural network $\Phi_{NN}(z, v)$;
- Suppose that: $\sup |\Phi(z, v) - \Phi_{NN}(z, v)| \leq \epsilon$, impose the constraint:

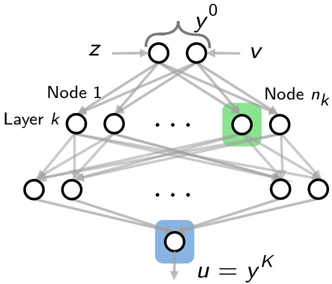
$$|\Phi_{NN}(z, v)| \leq \bar{u} - \epsilon. \qquad (\star)$$



If Φ_{NN} is a ReLU-ANN, $(\star)$ are mixed-integer linear constraints[4].

[4] Huu-Thinh Do and Ionela Prodan. On the constrained feedback linearization control based on the MILP representation of a ReLU-ANN, accepted to *IEEE Control Systems Letters*, 2024.

Convoluted constraints replaced by mixed-integer linear constraints



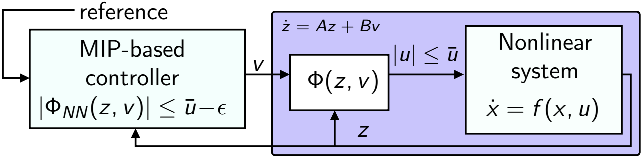
Given a network $\Phi_{NN}(z, v)$, the two following constraints are equivalent:

$$|\Phi_{NN}(z, v)| \leq \bar{u} - \epsilon \;\Leftrightarrow\; \begin{cases} y^0 = [z, v]^\top, k \in \{1, \ldots, K-1\}, j \in \{1, \ldots, n_k\}, \\ W^k y^{k-1} + b^k = \overline{y}^k - \underline{y}^k, \overline{y}^k \geq \mathbf{0}, \underline{y}^k \geq \mathbf{0}, \\ \alpha_j^k \in \{0, 1\}, \overline{y}_j^k \leq U_j^k \alpha_j^k, \underline{y}_j^k \leq -L_j^k(1 - \alpha_j^k), \\ y^K = W^K y^{K-1} + b^K, |y^K| \leq \bar{u} - \epsilon, \end{cases}$$

and

$$|\Phi_{NN}(z, v)| \leq \bar{u} - \epsilon \Rightarrow |\Phi(z, v)| \leq \bar{u}$$

Integration to optimization-based control



The tracking in the linearizing space can be handled by moving horizon MIP-based controller:

$$v^* = \arg\min_{v(t|kt_s)} \sum_{k=0}^{N_p-1} \ell(z(t|kt_s), v(t|kt_s))$$

$$|\Phi_{NN}(z(t|kt_s), v(t|kt_s))| \leq \bar{u} - \epsilon, \longrightarrow$$ mixed-integer linear constraints

$$z(t|(k+1)t_s) = A_d z(t|kt_s) + B_d v(t|kt_s), \longrightarrow$$ linearized dynamics

$$z(t|kt_s) \in \mathcal{X}_z, k \in \{0, ..., N_p - 1\}, \longrightarrow$$ state constraints!

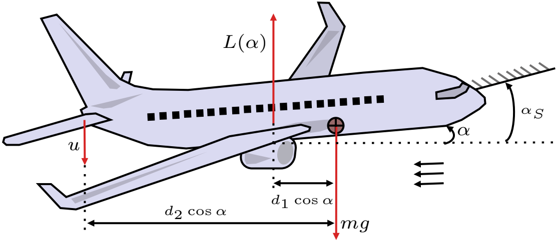
N_p is the prediction horizon, t_s is the sampling time, $\ell(z(t|kt_s), v(t|kt_s))$ is some cost function to minimize.

Longitudinal set-point tracking of an aircraft

Consider the dynamics Nicotra and Garone, 2018:

$$\ddot{\alpha} = (-d_1 L(\alpha) + u d_2)\cos\alpha,$$
$$\text{s.t } |u| \leq \overline{u}, \alpha \leq \alpha_S,$$

where $L(\alpha) = l_0 + l_1\alpha - l_3\alpha^3$ is the lift force generated by changing α, l_0, l_1, l_3, d_1, d_2 are the system's constant parameters, $\alpha_S = \sqrt{l_1/(3l_3)}$, $\overline{u}$ are the input amplitude bound.



⇒ *Feedback linearization solution*: $v \triangleq (-d_1 L(\alpha) + u d_2)\cos\alpha$.
The control problem becomes:

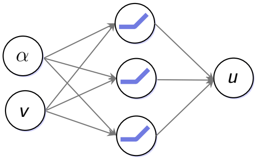
$$\ddot{\alpha} = v, \text{ s.t } \underbrace{\left| d_2^{-1}(v/\cos\alpha + d_1 L(\alpha)) \right| \leq \overline{u}}_{\text{non-convex constraint: } |\Phi(\alpha,v)| \leq \bar{u}} \text{ and } \alpha \leq \alpha_S.$$

Example

Consider a network of 1 hidden layer consisting of 3 nodes to approximate the function:

$$\Phi(\alpha, v) = d_2^{-1}(v/\cos\alpha + d_1 L(\alpha)).$$

- Approximating[1] network $\Phi(\alpha, v)$ with $\Phi_{NN}(\alpha, v)$, we have:



$$\begin{cases} y^1 = \sigma\left(\boldsymbol{W}^1 \begin{bmatrix}\alpha \\ v\end{bmatrix} + b^1\right) \\ \Phi_{NN}(\alpha, v) = \boldsymbol{W}^2 y^1 + b^2. \end{cases}$$

$$W^1 = \begin{bmatrix} 6.9544 & -0.7445 \\ -7.0939 & 0.0354 \\ 4.2001 & -0.0208 \end{bmatrix}, W^2 = [-1.5659 \;\; -1.2676 \;\; 2.1689],$$

$$b^1 = [14.9468 \;\; 1.4271 \;\; 0.8521], b^2 = 23.6044.$$

- Then the control problem is recast into:

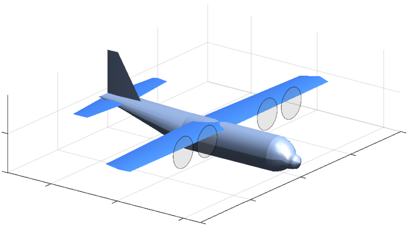
$$\ddot{\alpha} = v,$$

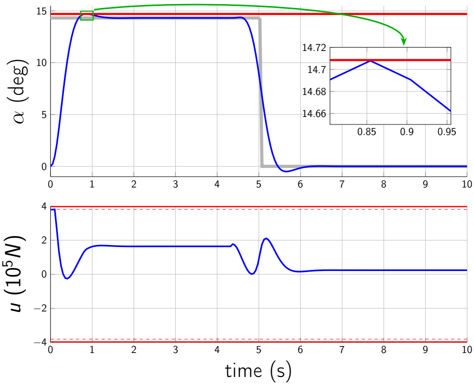
$$|\Phi_{NN}(\alpha, v)| \leq \bar{u} - \epsilon \text{ and } \alpha \leq \alpha_S,$$

with $\bar{u} = 4, \epsilon = 0.1897$.

[1] The neural network regression was done with `fitrnet` method of MATLAB

Simulation result[5]





Numerical specifications and simulation results

N_p	Sampling time t_s	Max CT	Mean CT	No. binary variables	Solver
15	50 ms	38.6 ms	30.8 ms	$N_p \times \sum_{k=1}^{K-1} n_k = 45$	Gurobi

[5] Animated with: "Draw a 3D airplane", Chad Greene (2024), MATLAB Central File Exchange. Retrieved May 26, 2024.

References I

[1] Mazen Alamir. "Stability proof for nonlinear MPC design using monotonically increasing weighting profiles without terminal constraints". In: *Automatica* 87 (2018), pp. 455–459.

[2] Andrzej Banaszuk and John Hauser. "Feedback linearization of transverse dynamics for periodic orbits". In: *Systems & control letters* 26.2 (1995), pp. 95–105.

[3] A. Bemporad et al. "The explicit linear quadratic regulator for constrained systems". In: *Automatica* 38.1 (2002), pp. 3–20.

[4] Francesco Borrelli. *Constrained optimal control of linear and hybrid systems*. Vol. 290. Springer, 2003.

[5] EM Bronstein. "Approximation of convex sets by polytopes". In: *Journal of Mathematical Sciences* 153.6 (2008), pp. 727–762.

[6] Bernard Charlet, Jean Lévine, and Riccardo Marino. "On dynamic feedback linearization". In: *Systems & Control Letters* 13.2 (1989), pp. 143–151.

[7] Mitchell R. Cohen, Khairi Abdulrahim, and James Richard Forbes. "Finite-Horizon LQR Control of Quadrotors on $SE_2(3)$". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5748–5755. DOI: 10.1109/LRA.2020.3010214.

[8] Bryan Convens et al. "A Terminal State Feasibility Governor for Real-Time Nonlinear Model Predictive Control Over Arbitrary Horizons". In: *IEEE Transactions on Control Systems Technology* (2024).

[9] C.R. Cutler et al. *Method for removal of PID dynamics from MPC models*. US Patent 7,263,473. 2007.

[10] Santosh Devasia, Degang Chen, and Brad Paden. "Nonlinear inversion-based output tracking". In: *IEEE Transactions on Automatic Control* 41.7 (1996), pp. 930–942.

[11] Johannes Diwold, Bernd Kolar, and Markus Schöberl. "Discrete-time flatness-based control of a gantry crane". In: *Control Engineering Practice* 119 (2022), p. 104980.

[12] Huu Thinh Do and Ionela Prodan. "Indoor experimental validation of MPC-based trajectory tracking for a quadcopter via a flat mapping approach". accepted to the 2023 European Control Conference (ECC), Available: https://youtu.be/1a1K6R6__3s, 2023.

[13] Huu-Thinh Do and Ionela Prodan. "Indoor experimental validation of MPC-based trajectory tracking for a quadcopter via a flat mapping approach". In: *2023 European Control Conference (ECC)*. IEEE. 2023, pp. 1–6.

[14] Nadeem Faiz, Sunil K Agrawal, and Richard M Murray. "Trajectory planning of differentially flat systems with dynamics and inequalities". In: *Journal of Guidance, Control, and Dynamics* 24.2 (2001), pp. 219–227.

[15] M. Fliess et al. "Flatness and defect of non-linear systems: introductory theory and examples". In: *International journal of control* 61.6 (1995), pp. 1327–1361.

[16] Simone Formentin and Marco Lovera. "Flatness-based control of a quadrotor helicopter via feedforward linearization.". In: *CDC-ECE*. 2011, pp. 6171–6176.

References II

[17] T. Geyer, F.D. Torrisi, and M. Morari. “Optimal complexity reduction of polyhedral piecewise affine systems”. In: *Automatica* 44.7 (2008), pp. 1728–1740. ISSN: 0005-1098.

[18] Mahmoud AK Gomaa et al. “Computationally Efficient Stability-Based Nonlinear Model Predictive Control Design for Quadrotor Aerial Vehicles”. In: *IEEE Transactions on Control Systems Technology* (2022).

[19] Eugene Gover and Nishan Krikorian. “Determinants and the volumes of parallelotopes and zonotopes”. In: *Linear Algebra and its Applications* 433.1 (July 2010), pp. 28–40.

[20] Melissa Greeff and Angela P Schoellig. “Flatness-based model predictive control for quadrotor trajectory tracking”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 6740–6745.

[21] P. Gritzmann and V. Klee. “On the complexity of some basic problems in computational convexity: I. Containment problems”. In: *Discrete Mathematics* 136.1-3 (1994), pp. 129–174.

[22] Lars Grüne and Jürgen Pannek. *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer-Verlag, London, 2011.

[23] Azeddine Houari et al. “Flatness-based control of three-phase inverter with output *LC* filter”. In: *IEEE Transactions on Industrial Electronics* 59.7 (2011), pp. 2890–2897.

[24] Li Jiajin et al. “Design of attitude controller using explicit model predictive control for an unmanned quadrotor helicopter”. In: *2017 Chinese Automation Congress (CAC)*. 2017, pp. 2853–2857. DOI: 10.1109/CAC.2017.8243262.

[25] M. Jünger et al. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Springer Verlag, 2009.

[26] O. Khatib. “Real-time obstacle avoidance for manipulators and mobile robots”. In: *The International Journal of Robotics Research* 5.1 (1986), p. 90. ISSN: 0278-3649.

[27] Cunjia Liu, Hao Lu, and Wen-Hua Chen. “An explicit MPC for quadrotor trajectory tracking”. In: *2015 34th Chinese Control Conference (CCC)*. IEEE. 2015, pp. 4055–4060.

[28] Hao Lu et al. “Constrained anti-disturbance control for a quadrotor based on differential flatness”. In: *International Journal of Systems Science* 48.6 (2017), pp. 1182–1193.

[29] Tom Lyche, Carla Manni, and Hendrik Speleers. “Foundations of spline theory: B-splines, spline approximation, and hierarchical refinement”. In: *Splines and PDEs: From Approximation Theory to Numerical Linear Algebra*. Springer, 2018, pp. 1–76.

[30] Bernhard M Maschke and Arjan J van der Schaft. “Port-controlled Hamiltonian systems: modelling origins and system theoretic properties”. In: *IFAC Proceedings Volumes* 25.13 (1992), pp. 359–365.

[31] David Q Mayne et al. “Constrained model predictive control: Stability and optimality”. In: *Automatica* 36.6 (2000), pp. 789–814.

References III

[32] Daniel Mellinger and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors". In: *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. Shanghai, China, 2011, pp. 2520–2525.

[33] Vincent Morio et al. "Onboard path planning for reusable launch vehicles application to the shuttle orbiter reentry mission". In: *International Review of Aerospace Engineering* 1.6 (2008), pp. 492–503.

[34] Mark W Mueller and Raffaello D'Andrea. "A model predictive controller for quadrocopter state interception". In: *Proceedings of the IEEE European Control Conference (ECC)*. 2013, pp. 1383–1389.

[35] Ngoc Thinh Nguyen, Ionela Prodan, and Laurent Lefevre. "A stabilizing NMPC design for thrust-propelled vehicles dynamics via feedback linearization". In: *2019 American Control Conference*. IEEE. 2019, pp. 2909–2914.

[36] Ngoc Thinh Nguyen, Ionela Prodan, and Laurent Lefèvre. "Flat trajectory design and tracking with saturation guarantees: a nano-drone application". In: *International Journal of Control* 93.6 (2020), pp. 1266–1279.

[37] Ngoc Thinh Nguyen, Ionela Prodan, and Laurent Lefèvre. "Multicopter Attitude Control through NMPC Design with Guaranteed Stability". In: *IFAC-PapersOnLine* 53.2 (2020). 21st IFAC World Congress, pp. 6101–6106. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2020.12.1685. URL: https://www.sciencedirect.com/science/article/pii/S2405896320322886.

[38] Marco M Nicotra and Emanuele Garone. "The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems". In: *IEEE Control Systems Magazine* 38.4 (2018), pp. 89–107.

[39] Mateus de Freitas Virgilio Pereira, Ilya V Kolmanovsky, and Carlos ES Cesnik. "Nonlinear model predictive control with aggregated constraints". In: *Automatica* 146 (2022), p. 110649.

[40] L. Piegl and W. Tiller. "B-spline Curves and Surfaces". In: *The NURBS Book*. Springer, 1995, pp. 81–116.

[41] Ionela Prodan et al. *Mixed-integer representations in control design: Mathematical foundations and applications*. Springer, 2015.

[42] A.I. Propoi. "Use of linear programming methods for synthesizing sampled-data automatic systems". In: *Automation and Remote Control* 24.7 (1963), pp. 837–844.

[43] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. 2009.

[44] J. Richalet and D. O'Donovan. *Predictive Functional Control: Principles and Industrial Applications*. Springer, 2009.

[45] Richard P Stanley et al. "An introduction to hyperplane arrangements". In: *Geometric combinatorics* 13 (2004), pp. 389–496.

[46] F. Suryawan, J. De Doná, and M. Seron. "Minimum-time trajectory generation for constrained linear systems using flatness and B-splines". In: *International Journal of Control* 84.9 (2011), pp. 1565–1585.

References IV

[47] Bálint Takács et al. "Python code generation for explicit MPC in MPT". In: *2016 European Control Conference*. IEEE. 2016.

[48] Guillem Torrente et al. "Data-driven MPC for quadrotors". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3769–3776.

[49] Dong Wang et al. "Efficient nonlinear model predictive control for quadrotor trajectory tracking: Algorithms and experiment". In: *IEEE Transactions on Cybernetics* 51.10 (2021), pp. 5057–5068.

[50] Andrea Zanelli et al. "Nonlinear model predictive control of a human-sized quadrotor". In: *2018 European Control Conference (ECC)*. IEEE. 2018, pp. 1542–1547.

[51] Melanie Nicole Zeilinger, Colin Neil Jones, and Manfred Morari. "Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization". In: *IEEE Transactions on Automatic Control* 56.7 (2011), pp. 1524–1534.