Computer-generated Control Lyapunov Function via offline linear programming

Huu-Thinh DO[1], Franco BLANCHINI[2], Stefano MIANI [2], Ionela PRODAN[1]

[1]{huu-thinh.do,ionela.prodan}@lcis.grenoble-inp.fr
Univ. Grenoble Alpes, Grenoble INP*, LCIS, F-26000, Valence, France.
*Institute of Engineering and Management Univ. Grenoble Alpes
[2]{franco.blanchini,miani.stefano}@uniud.it
Dipartimento di Matematica e Informatica, Università di Udine, 33100 Udine, Italy.

Outline

1. Computer generated Control Lyapunov Function
   - Motivation and main idea
   - Illustrative example
   - Experimental validation on quadcopter stabilization
2. Conclusion and future direction

Outline

Control Lyapunov Function

Consider a control affine system:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$$

CLF $V(\boldsymbol{x})$

$\Rightarrow$ find $\boldsymbol{u} \in \{\dot{V}(\boldsymbol{x}) \leq -\beta V(\boldsymbol{x})\}$

Control Lyapunov Function

Consider a control affine system:
$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$

Explicit control
via "universal" formulas

CLF $V(\boldsymbol{x})$

$\Rightarrow$ find $\boldsymbol{u} \in \{\dot{V}(\boldsymbol{x}) \leq -\beta V(\boldsymbol{x})\}$

Control Lyapunov Function

Consider a control affine system:

$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$

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

CLF $V(\boldsymbol{x})$

Explicit control
via "universal" formulas

$\Rightarrow$ find $\boldsymbol{u} \in \{\dot{V}(\boldsymbol{x}) \leq -\beta V(\boldsymbol{x})\} \bigcap \mathcal{U}$

Control Lyapunov Function

Consider a control affine system:

$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$

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

CLF $V(\boldsymbol{x})$

$\Rightarrow$ find $\boldsymbol{u} \in \{\dot{V}(\boldsymbol{x}) \leq -\beta V(\boldsymbol{x})\} \bigcap \mathcal{U}$

Explicit control
via "universal" formulas

$\mathcal{U} \leftarrow \mathcal{L}_p$ norm ball/polytope

[Sontag, 1989, Malisoff and Sontag, 2000]
[Yamashita et al., 2022, Solís-Daun and Leyva, 2011]
[Leyva et al., 2023]

Implicit control
(online constrained optimization)

[Freeman and Kokotovic, 1996]
[Suarez et al., 2001]

Control Lyapunov Function

Consider a control affine system:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$$
$$\boldsymbol{u} \in \mathcal{U}$$

CLF $V(\boldsymbol{x})$

$\Rightarrow$ find $\boldsymbol{u} \in \{\dot{V}(\boldsymbol{x}) \leq -\beta V(\boldsymbol{x})\} \bigcap \mathcal{U}$

Simple online implementation

Explicit control
via "universal" formulas

$\mathcal{U} \leftarrow \mathcal{L}_p$ norm ball/polytope

[Sontag, 1989, Malisoff and Sontag, 2000]
[Yamashita et al., 2022, Solís-Daun and Leyva, 2011]
[Leyva et al., 2023]

Implicit control
(online constrained optimization)

[Freeman and Kokotovic, 1996]
[Suarez et al., 2001]

Control Lyapunov Function

Consider a control affine system:

$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$

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

CLF $V(\boldsymbol{x})$

$\Rightarrow$ find $\boldsymbol{u} \in \{\dot{V}(\boldsymbol{x}) \leq -\beta V(\boldsymbol{x})\} \bigcap \mathcal{U}$

Simple online implementation

Explicit control
via "universal" formulas

$\mathcal{U} \leftarrow \mathcal{L}_p$ norm ball/polytope

[Sontag, 1989, Malisoff and Sontag, 2000]
[Yamashita et al., 2022, Solís-Daun and Leyva, 2011]
[Leyva et al., 2023]

Implicit control
(online constrained optimization)

[Freeman and Kokotovic, 1996]
[Suarez et al., 2001]

But how to find $V(\boldsymbol{x})$ a priori to the control design?

Control Lyapunov Function

Consider a control affine system:

$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$

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

CLF $V(\boldsymbol{x})$

$\Rightarrow$ find $\boldsymbol{u} \in \{\dot{V}(\boldsymbol{x}) \leq -\beta V(\boldsymbol{x})\} \bigcap \mathcal{U}$

Simple online implementation

Explicit control
via "universal" formulas

$\mathcal{U} \leftarrow \mathcal{L}_p$ norm ball/polytope

[Sontag, 1989, Malisoff and Sontag, 2000]
[Yamashita et al., 2022, Solís-Daun and Leyva, 2011]
[Leyva et al., 2023]

Implicit control
(online constrained optimization)

[Freeman and Kokotovic, 1996]
[Suarez et al., 2001]

But how to find $V(\boldsymbol{x})$ a priori to the control design?

Existing particular works: Polynomial systems [Prajna et al., 2004, Chesi, 2010],
PWA linear systems [Lazar and Jokić, 2010]

Generalized framework: space triangulation [Steentjes et al., 2020, Lavaei and Bridgeman, 2023],
Neural networks [Min et al., 2023]

Control Lyapunov Function

Consider a control affine system:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$$
$$\boldsymbol{u} \in \mathcal{U}$$

CLF $V(\boldsymbol{x})$

$\Rightarrow$ find $\boldsymbol{u} \in \{\dot{V}(\boldsymbol{x}) \leq -\beta V(\boldsymbol{x})\} \bigcap \mathcal{U}$

Simple online implementation

Explicit control via "universal" formulas

$\mathcal{U} \leftarrow \mathcal{L}_p$ norm ball/polytope

[Sontag, 1989, Malisoff and Sontag, 2000]
[Yamashita et al., 2022, Solís-Daun and Leyva, 2011]
[Leyva et al., 2023]

Implicit control (online constrained optimization)

[Freeman and Kokotovic, 1996]
[Suarez et al., 2001]

But how to find $V(\boldsymbol{x})$ a priori to the control design?

Existing particular works: Polynomial systems [Prajna et al., 2004, Chesi, 2010], PWA linear systems [Lazar and Jokić, 2010]

Generalized framework: space triangulation [Steentjes et al., 2020, Lavaei and Bridgeman, 2023], Neural networks [Min et al., 2023]

Computationally costly to artificially generate a CLF

Model Predictive Control

Consider a standard MPC design:

$$
\begin{aligned}
&\arg\min_{\boldsymbol{u}(\cdot)} \sum_{k=0}^{N_p} \left(\|\boldsymbol{x}(k)\|^2_{\boldsymbol{Q}} + \|\boldsymbol{u}(k)\|^2_{\boldsymbol{R}}\right) + \|\boldsymbol{x}(N_p)\|^2_{\boldsymbol{P}} \\
&\text{s.t} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \tau\left[f(\boldsymbol{x}(k)) + g(\boldsymbol{x}(k))\boldsymbol{u}(k)\right], \\ \tau \text{ is the sampling time, } \boldsymbol{u}(k) \in \mathcal{U}, \\ \boldsymbol{x}(k) \in \mathcal{X}, \boldsymbol{x}(N_p) \in \mathcal{X}_f, \end{cases}
\end{aligned}
$$

MPC: an alternative method to sidestep CLF design

Model Predictive Control

Consider a standard MPC design:

$$\arg\min_{\boldsymbol{u}(\cdot)} \sum_{k=0}^{N_p} \left(\|\boldsymbol{x}(k)\|^2_{\boldsymbol{Q}} + \|\boldsymbol{u}(k)\|^2_{\boldsymbol{R}}\right) + \|\boldsymbol{x}(N_p)\|^2_{\boldsymbol{P}}$$

$$\text{s.t} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \tau\left[f(\boldsymbol{x}(k)) + g(\boldsymbol{x}(k))\boldsymbol{u}(k)\right], \\ \tau \text{ is the sampling time, } \boldsymbol{u}(k) \in \mathcal{U}, \\ \boldsymbol{x}(k) \in \mathcal{X}, \boldsymbol{x}(N_p) \in \mathcal{X}_f, \end{cases}$$

MPC: an alternative method to sidestep CLF design

The online nonlinear program is generally complex

Model Predictive Control

Consider a standard MPC design:

$$\arg\min_{u(\cdot)} \sum_{k=0}^{N_p} \left(\|\boldsymbol{x}(k)\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}(k)\|_{\boldsymbol{R}}^2\right) + \|\boldsymbol{x}(N_p)\|_{\boldsymbol{P}}^2$$

$$\text{s.t} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \tau\left[f(\boldsymbol{x}(k)) + g(\boldsymbol{x}(k))\boldsymbol{u}(k)\right], \\ \tau \text{ is the sampling time, } \boldsymbol{u}(k) \in \mathcal{U}, \\ \boldsymbol{x}(k) \in \mathcal{X}, \boldsymbol{x}(N_p) \in \mathcal{X}_f, \end{cases}$$

MPC: an alternative method to sidestep CLF design

The online nonlinear program is generally complex

CLF-based control	versus	**MPC**
Complicated offline synthesis		Systematic offline design
Straightforward control implementation		Computationally demanding online implementation

Model Predictive Control

Consider a standard MPC design:

$$\arg\min_{\boldsymbol{u}(\cdot)} \sum_{k=0}^{N_p} \left(\|\boldsymbol{x}(k)\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}(k)\|_{\boldsymbol{R}}^2 \right) + \|\boldsymbol{x}(N_p)\|_{\boldsymbol{P}}^2$$

$$\text{s.t} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \tau \left[f(\boldsymbol{x}(k)) + g(\boldsymbol{x}(k))\boldsymbol{u}(k) \right], \\ \tau \text{ is the sampling time, } \boldsymbol{u}(k) \in \mathcal{U}, \\ \boldsymbol{x}(k) \in \mathcal{X}, \boldsymbol{x}(N_p) \in \mathcal{X}_f, \end{cases}$$

MPC: an alternative method to sidestep CLF design

The online nonlinear program is generally complex

CLF-based control versus **MPC**

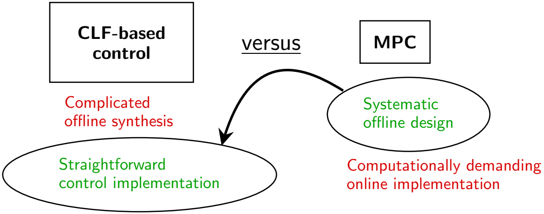
Complicated offline synthesis

Straightforward control implementation

Systematic offline design

Computationally demanding online implementation

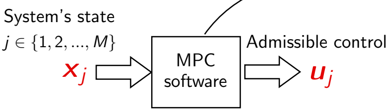
Main idea



Main idea: Use the admissible control from the MPC law to parameterize a CLF.

CLF offline synthesis based on Linear Program

- **Step 1: Generate and collect admissible points in state-input space**

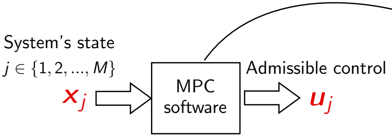


$$\arg\min_{\boldsymbol{u}(\cdot)} \sum_{k=0}^{N_p} \left(\|\boldsymbol{x}(k)\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}(k)\|_{\boldsymbol{R}}^2 \right) + \|\boldsymbol{x}(N_p)\|_{\boldsymbol{P}}^2$$

$$\text{s.t} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \tau \left[f(\boldsymbol{x}(k)) + g(\boldsymbol{x}(k))\boldsymbol{u}(k) \right], \\ \boldsymbol{u}(k) \in \mathcal{U}, \\ \boldsymbol{x}(k) \in \mathcal{X}, \boldsymbol{x}(N_p) \in \mathcal{X}_f, \end{cases}$$

CLF offline synthesis based on Linear Program

- **Step 1: Generate and collect admissible points in state-input space**



$$\underset{\boldsymbol{u}(\cdot)}{\arg\min} \sum_{k=0}^{N_p} \left(\|\boldsymbol{x}(k)\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}(k)\|_{\boldsymbol{R}}^2 \right) + \|\boldsymbol{x}(N_p)\|_{\boldsymbol{P}}^2$$
$$\text{s.t} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \tau \left[f(\boldsymbol{x}(k)) + g(\boldsymbol{x}(k))\boldsymbol{u}(k) \right], \\ \boldsymbol{u}(k) \in \mathcal{U}, \\ \boldsymbol{x}(k) \in \mathcal{X}, \boldsymbol{x}(N_p) \in \mathcal{X}_f, \end{cases}$$

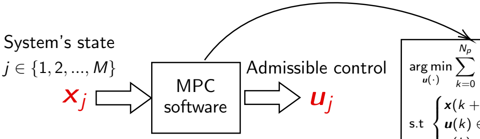
- **Step 2: Fix a collection of basis candidate CLFs $V_i(\boldsymbol{x})$ and define:**

$$V(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i V_i(\boldsymbol{x})$$

where $\alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i = 1$, are the scalar coefficients parameterizing the CLF to find later.

CLF offline synthesis based on Linear Program

- **Step 1: Generate and collect admissible points in state-input space**



$$\arg\min_{u(\cdot)} \sum_{k=0}^{N_p} \left(\|x(k)\|_Q^2 + \|u(k)\|_R^2 \right) + \|x(N_p)\|_P^2$$

$$\text{s.t} \begin{cases} x(k+1) = x(k) + \tau \left[f(x(k)) + g(x(k))u(k) \right], \\ u(k) \in \mathcal{U}, \\ x(k) \in \mathcal{X}, x(N_p) \in \mathcal{X}_f, \end{cases}$$

- **Step 2: Fix a collection of basis candidate CLFs $V_i(x)$ and define:**

$$V(x) = \sum_{i=1}^{N} \alpha_i V_i(x)$$

where $\alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i = 1$, are the scalar coefficients parameterizing the CLF to find later.

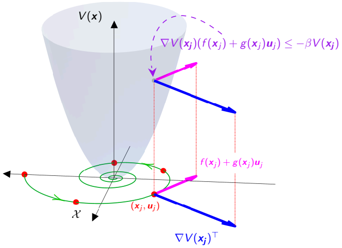
- **Step 3: Find the coefficients α_i via the following linear program, for some $\beta > 0$:**

$$\min_{\alpha_i} \sum_{i=1}^{N} \ell_i \alpha_i$$

$$\sum_{i=1}^{N} \alpha_i \left[\nabla V_i(x_j) \left[f(x_j) + g(x_j) u_j \right] + \beta V_i(x_j) \right] \leq 0,$$

$$\sum_{i=1}^{N} \alpha_i = 1, \alpha_i \geq 0, j \in \{1, ..., M\}.$$

where $\ell_i > 0$, are user-defined scalar weighting, prioritizing the dominance of $V_i(x)$.



Illustrative example

Consider the double integrator system:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu$$

$\boldsymbol{x} = [x_1, x_2]^\top, |u| \leq 0.8154,$

$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = [0 \quad 1]^\top.$

Illustrative example

Consider the double integrator system:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu$$

$\boldsymbol{x} = [x_1, x_2]^\top, |u| \leq 0.8154,$

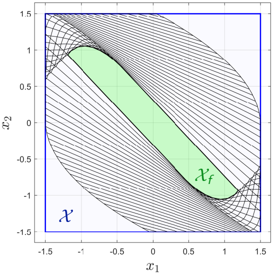
$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = [0 \quad 1]^\top.$

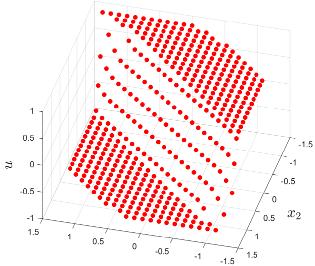
Step 1: collect admissible pairs from the MPC software

Discretized MPC design:

$$\arg\min_{u(\cdot)} \sum_{k=0}^{N_p} \left(\|\boldsymbol{x}(k)\|^2_{\boldsymbol{Q}} + \|\boldsymbol{u}(k)\|^2_{\boldsymbol{R}} \right) + \|\boldsymbol{x}(N_p)\|^2_{\boldsymbol{P}}$$

$$\text{s.t} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \tau \left[A\boldsymbol{x}(k) + Bu(k)\right], \\ \boldsymbol{u}(k) \in \mathcal{U}, \boldsymbol{x}(k) \in \mathcal{X}, \boldsymbol{x}(N_p) \in \mathcal{X}_f, \end{cases}$$





with $\boldsymbol{Q} = \text{diag}(50, 5), \boldsymbol{R} = 5, \boldsymbol{P} = \begin{bmatrix} 479.6118 & 181.0469 \\ 181.0469 & 155.5598 \end{bmatrix}$, $\mathcal{X}_f$ is the maximal positive invariant set with the local control $u = -[2.7617 \ 2.6491]\boldsymbol{x}$.

Illustrative example

Step 2: Fix a set of basis functions $V_i(\boldsymbol{x}), i \in \{1, ..., N\}$.

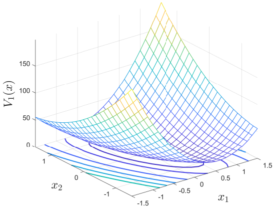
Illustrative example

Step 2: Fix a set of basis functions $V_i(\boldsymbol{x}), i \in \{1, ..., N\}$.

The first member:

$$V_1(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{P}_{lqr}\boldsymbol{x},$$

a quadratic CLF (for optimality).



Illustrative example

Step 2: Fix a set of basis functions $V_i(\boldsymbol{x}), i \in \{1, ..., N\}$.

The first member:

$$V_1(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{P}_{lqr} \boldsymbol{x},$$

a quadratic CLF (for optimality).

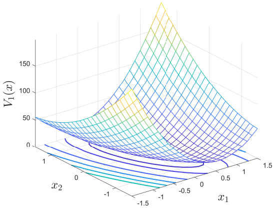
The other members: $2p$-norm CLF (for expansion of the domain of attraction) [Blanchini and Miani, 1999]:

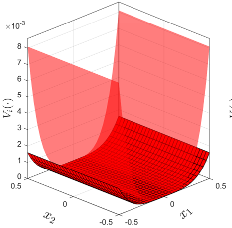
$$V_i(\boldsymbol{x}) = \|\boldsymbol{G}\boldsymbol{x}\|_{2p}^{2p},$$

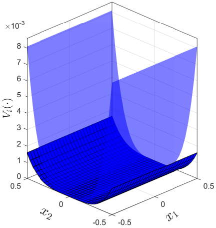
$\boldsymbol{G}$ is a full row rank.

$$V_{2\le i\le 100}(\boldsymbol{x}) \in \Big\{ \|\eta \boldsymbol{F}\boldsymbol{x}\|_4^4, \eta \in \text{linspace}(33, 0.4, 2.0)$$

$$\boldsymbol{F} \in \{[1 \quad 0], [0 \quad 1], [1 \quad 1]\} \Big\}.$$







Illustrative example

Step 3: Find the CLF

$V(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i V_i(\boldsymbol{x})$.

$$\alpha_i^* = \arg\min_{\alpha_i} \sum_{i=1}^{N} \ell_i \alpha_i$$

Parameters	Value
ℓ_1	0.05
$\ell_{2\le i\le 100}$	0.1
β	0.1275
N^o of basis functions N	100
N^o of sampled pairs M	625

$$\sum_{i=1}^{N} \alpha_i \left[\nabla V_i(\boldsymbol{x}_j)\left[f(\boldsymbol{x}_j) + g(\boldsymbol{x}_j)\boldsymbol{u}_j\right] + \beta V_i(\boldsymbol{x}_j)\right] \le 0; \sum_{i=1}^{N} \alpha_i = 1; \alpha_i \ge 0, j \in \{1, ..., M\}.$$

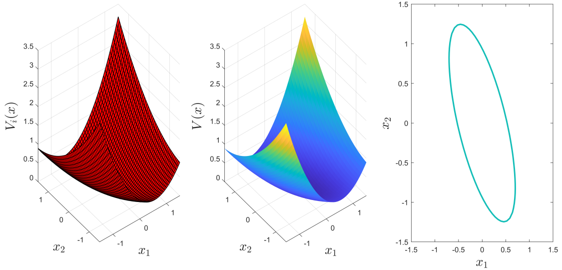


Figure 1: (left) Member functions (middle) the generated CLF and (right) the domain of attraction

$$V(\boldsymbol{x}) = 0.016\|\boldsymbol{x}\|^2_{\boldsymbol{P}_{lqr}} + 0.2293(2[1\ 0]\boldsymbol{x})^4 + 0.5406(2[0\ 1]\boldsymbol{x})^4 + 0.2141(2[1\ 1]\boldsymbol{x})^4.$$

solver: `linprog`, MATLAB 2021b, solver time: 0.0315 (sec).

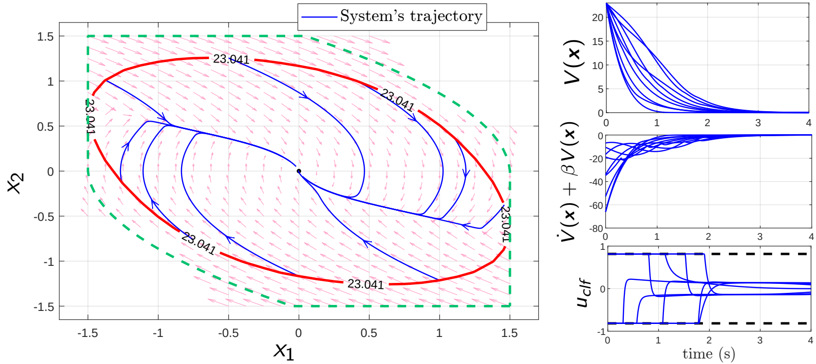
Simulation

Controller implementation:

$$u_{clf}(\boldsymbol{x}) = \arg\min_{u} \|\boldsymbol{u} - \boldsymbol{u}_d(\boldsymbol{x})\|^2$$

$$\text{s.t} \begin{cases} \boldsymbol{u} \in \mathcal{U}, \\ \nabla V(\boldsymbol{x})(f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}) \leq -\beta V(\boldsymbol{x}). \end{cases}$$

where $\boldsymbol{u}_d(\boldsymbol{x})$ is a user-defined desired control to avoid slow performance due to possibly small β.



Validation with quadcopter outer-loop control

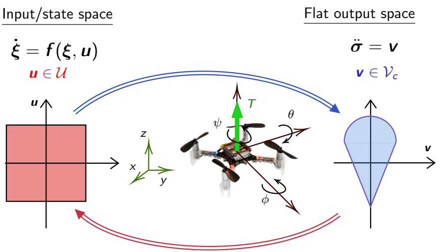


Figure 2: Constraint deformation due to the flatness-based feedback linearization.

Consider the quadcopter's outer-loop control:

$$\dot{\boldsymbol{\xi}} = A\boldsymbol{\xi} + B(\boldsymbol{R}_\psi \boldsymbol{f}(\boldsymbol{u}) - g\boldsymbol{e}_3)$$

$$\boldsymbol{\sigma} = [x, y, z]^\top \in \mathbb{R}^3,\ \boldsymbol{\xi} = [\boldsymbol{\sigma}^\top, \dot{\boldsymbol{\sigma}}^\top]^\top$$

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

Feedback linearization[1] $\quad \boldsymbol{u} = \boldsymbol{f}^{-1}(\boldsymbol{R}_\psi^{-1}(\boldsymbol{v} + g\boldsymbol{e}_3))$

$$\ddot{\boldsymbol{\sigma}} = \boldsymbol{v} = [v_1, v_2, v_3]^\top$$

$$\boldsymbol{v} \in \mathcal{V}_c = \left\{\|\boldsymbol{v} + g\boldsymbol{e}_3\| \le T_{max},\ \sqrt{v_1^2 + v_2^2} \le \tan \epsilon_{max}(v_3 + g)\right\}.$$

[1] H.T. Do and I. Prodan, "Indoor experimental validation of MPC-based trajectory tracking for a quadcopter via a flat mapping approach," 2023 European Control Conference, Bucharest, Romania.

Validation with quadcopter outer-loop control

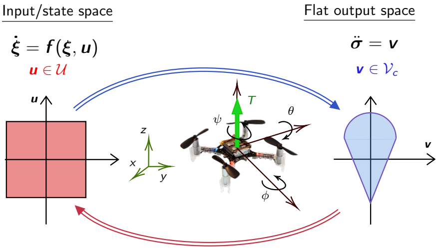


Figure 2: Constraint deformation due to the flatness-based feedback linearization.

Consider the quadcopter's outer-loop control:

$$\dot{\boldsymbol{\xi}} = A\boldsymbol{\xi} + B(\boldsymbol{R}_\psi \boldsymbol{f}(\boldsymbol{u}) - g\boldsymbol{e}_3)$$

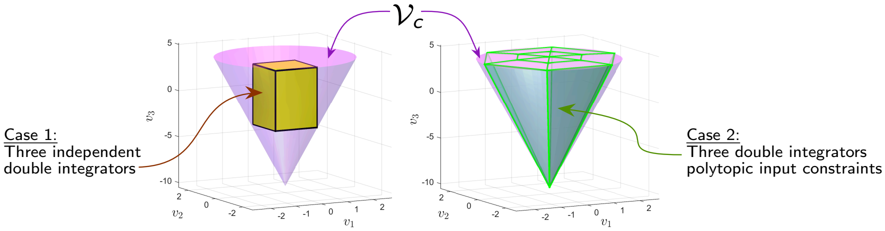
$$\boldsymbol{\sigma} = [x, y, z]^\top \in \mathbb{R}^3,\ \boldsymbol{\xi} = [\boldsymbol{\sigma}^\top, \dot{\boldsymbol{\sigma}}^\top]^\top$$

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

Feedback linearization[1] $\boldsymbol{u} = \boldsymbol{f}^{-1}(\boldsymbol{R}_\psi^{-1}(\boldsymbol{v} + g\boldsymbol{e}_3))$

$$\ddot{\boldsymbol{\sigma}} = \boldsymbol{v} = [v_1, v_2, v_3]^\top$$

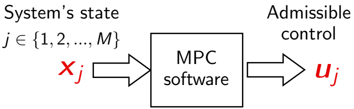
$$\boldsymbol{v} \in \mathcal{V}_c = \Big\{\|\boldsymbol{v} + g\boldsymbol{e}_3\| \le T_{max},\ \sqrt{v_1^2 + v_2^2} \le \tan\epsilon_{max}(v_3 + g)\Big\}.$$



[1] H.T. Do and I. Prodan, "Indoor experimental validation of MPC-based trajectory tracking for a quadcopter via a flat mapping approach," 2023 European Control Conference, Bucharest, Romania.

Validation with quadcopter outer-loop control

Step 1: Generate and collect admissible points in state-input space



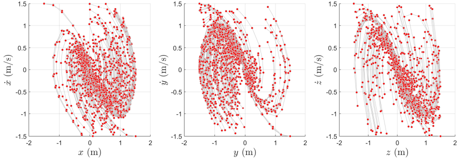


Figure 3: Sampled admissible pairs from MPC for Case 2.

Step 2 & 3: fix the basis functions and solve the LP for the CLFs

Case 1: The system is decoupled into three separated double integrators:

$$\ddot{x} = v_1, \ddot{y} = v_2, \ddot{z} = v_3,$$

with $|v_i| \leq \bar{v}_i, i \in \{1, 2, 3\}$. The CLFs:

- For $q \in \{x, y\}$ axis:
  $$V^q(\boldsymbol{x}) = 0.016\|\boldsymbol{x}\|^2_{\boldsymbol{P}_{lqr}} + 0.2293(2[1\,0]\boldsymbol{x})^4 + 0.5406(2[0\,1]\boldsymbol{x})^4 + 0.2141(2[1\,1]\boldsymbol{x})^4.$$
- For the z axis:
  $$V^z(\boldsymbol{x}) = 0.0119\|\boldsymbol{x}\|^2_{\boldsymbol{P}_{lqr}} + 0.3862(2[1\,0]\boldsymbol{x})^4 + 0.6019(2[1\,1]\boldsymbol{x})^4.$$

Case 2: The system as three concatenated double integrators:

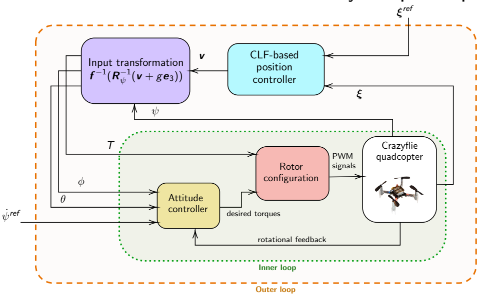
$$\dot{\boldsymbol{\xi}} = A\boldsymbol{\xi} + B\boldsymbol{v},$$

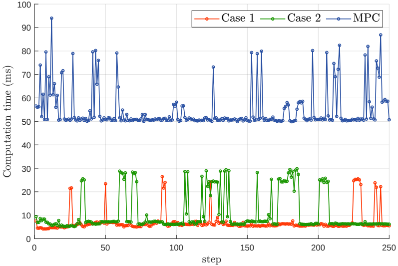
with $H\boldsymbol{v} \leq h$.

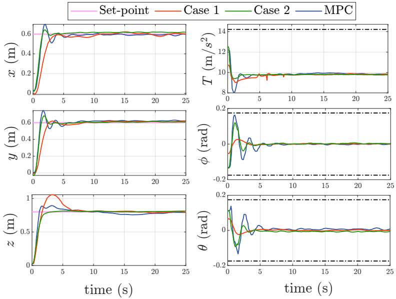
$$V(\boldsymbol{\xi}) = 0.0244\|\boldsymbol{\xi}\|^2_{\boldsymbol{P}_{lqr}} + 0.4508([0\,1\,0\,0\,0\,0]\boldsymbol{\xi})^4 + 0.0674([1\,1\,0\,0\,0\,0]\boldsymbol{\xi})^4 + 0.1631([0\,0\,1\,0\,0\,0]\boldsymbol{\xi})^4 + 0.1841([0\,0\,1\,1\,0\,0]\boldsymbol{\xi})^4 + 0.1103([0\,0\,0\,0\,0\,1]\boldsymbol{\xi})^4.$$

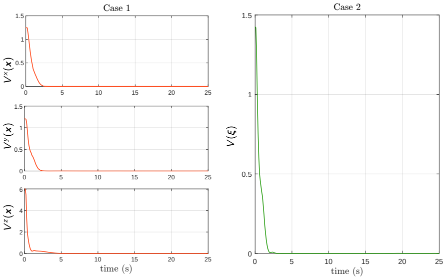
Set-point tracking, comparison with MPC

Hierarchical control structure of Crazyflie quadcopter:







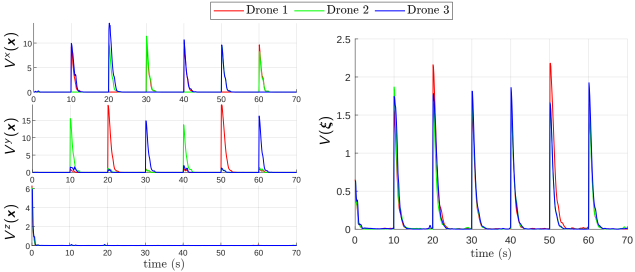


Multiple drones control

The low complexity of the CLF-based law allows controlling multiple drones in a centralized manner.

Table 1: Experiment results

	Controller	RMS tracking error (cm)			Avg. CT
Sce. 1	Case 1	14.11			7.02 ms
	Case 2	11.39			10.74 ms
	MPC	11.48			54.70 ms
Sce. 2		Drone 1	Drone 2	Drone 3	
	Case 1	24.09	22.20	21.33	24.09 ms
	Case 2	19.14	18.57	18.71	20.51 ms
	MPC	×	×	×	×



Experiment video: `https://youtu.be/PP5fZnIyH54`

Outline

Conclusion

Conclusion:

- A linear program-based technique was proposed to synthesize CLF with admissible state-input pairs collected from an MPC software.
- The result shows that the technique has low computational footprint during the offline synthesis, allowing the benefits of CLF-based online implementation.

Future work:

- Investigate how to efficiently choose the sampled data from MPC, providing theoretical guarantees over a finite set of constraints and allowing scalability.
- Include robustness analysis in the CLF synthesis.

References I

Blanchini, F. and Miani, S. (1999).
A new class of universal lyapunov functions for the control of uncertain linear systems.
IEEE Transactions on Automatic Control, 44(3):641–647.

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

Freeman, R. A. and Kokotovic, P. V. (1996).
Inverse optimality in robust stabilization.
SIAM journal on control and optimization, 34(4):1365–1391.

Lavaei, R. and Bridgeman, L. J. (2023).
Systematic, lyapunov-based, safe and stabilizing controller synthesis for constrained nonlinear systems.
IEEE Transactions on Automatic Control.

Lazar, M. and Jokić, A. (2010).
On infinity norms as lyapunov functions for piecewise affine systems.
In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pages 131–140.

Leyva, H., Aguirre-Hernández, B., and Espinoza, J. F. (2023).
Stabilization of affine systems with polytopic control value sets.
Journal of Dynamical and Control Systems, pages 1–13.

References II

Malisoff, M. and Sontag, E. D. (2000).
Universal formulas for feedback stabilization with respect to minkowski balls.
Systems & Control Letters, 40(4):247–260.

Min, Y., Richards, S. M., and Azizan, N. (2023).
Data-driven control with inherent lyapunov stability.
arXiv preprint arXiv:2303.03157.

Prajna, S., Papachristodoulou, A., and Wu, F. (2004).
Nonlinear control synthesis by sum of squares optimization: A lyapunov-based approach.
In *2004 5th Asian control conference (IEEE Cat. No. 04EX904)*, volume 1, pages 157–165. IEEE.

Solís-Daun, J. and Leyva, H. (2011).
On the global clf stabilization of systems with polytopic control value sets.
IFAC Proceedings Volumes, 44(1):11042–11047.

Sontag, E. D. (1989).
A 'universal'construction of artstein's theorem on nonlinear stabilization.
Systems & control letters, 13(2):117–123.

Steentjes, T. R. V., Lazar, M., and Doban, A. I. (2020).
Construction of continuous and piecewise affine feedback stabilizers for nonlinear systems.
IEEE Transactions on Automatic Control, 66(9):4059–4068.

References III

Suarez, R., Solis-Daun, J., and Aguirre, B. (2001).
Global clf stabilization for systems with compact convex control value sets.
In *Proceedings of the 40th IEEE conference on decision and control (Cat. No. 01CH37228)*, volume 4, pages 3838–3843. IEEE.

Yamashita, Y., Matsukizono, R., and Kobayashi, K. (2022).
Asymptotic stabilization of nonlinear systems with convex-polytope input constraints by continuous input.
Automatica, 138:110032.