

# Behavioral System Theory in Safe Predictive Control

Formal Methods for AI-Enabled Cyber-Physical Systems

Author: Ayse Aybüke Ulusarslan

Supervisor: Yongkuan Zhang, M.Sc.



05-FM

## Motivation

- Traditional control methods often rely on explicit modeling - not always practical
- Limits the usability of learning-based control systems
- Behavioral System Theory offers a solution

## Behavioral System Theory

### Key Assumptions

1. **Persistence of Excitation:** A signal  $u \in \mathbb{R}^{m \times T}$  is PE of order  $L$  if  $H_L(u)$  has full row rank.

2. **LTI'ness:** The system  $\Sigma = (T, W, \mathcal{B})$  satisfies:

(a) **Linearity:**  $\mathcal{B}$  is a vector space:  $\mathcal{B}(\alpha_1 \cdot w_1 + \alpha_2 \cdot w_2) = \alpha_1 \cdot \mathcal{B}(w_1) + \alpha_2 \cdot \mathcal{B}(w_2)$

(b) **Time Invariance:**  $B \subseteq \sigma B$ , where  $\sigma$  is the shift operator:  $(\sigma w)(t) = w(t+1)$

(c) **Completeness:**  $w|_{[t_0, t_1]} \in \mathcal{B}|_{[t_0, t_1]} \implies w \in \mathcal{B}$

3. **Sufficient trajectory length T:**  $T \geq (m+1)(T_{\text{ini}} + N + n(B)) - 1$ , where:

- $N$ : Prediction horizon.
- $n(B)$ : System state dimension.
- $m$ : Input dimension.

Traditional control relies on state-space models described by:

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t),$$

where  $(A, B, C, D)$  are system matrices. Behavioral System Theory (BST) offers a data-driven alternative using Hankel matrices:

$$H_L(u) = \begin{bmatrix} u_1 & u_2 & \cdots & u_{T-L+1} \\ u_2 & u_3 & \cdots & u_{T-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{L+1} & \cdots & u_T \end{bmatrix},$$

constructed from input trajectories over a window of length  $L$ . According to the Fundamental Lemma [3], if  $u^d$  is persistently exciting of order  $L + n(\mathcal{B})$ , then:

$$(u, y) \in \mathcal{B} \iff \exists \alpha \in \mathbb{R}^{T-L+1} \text{ s.t. } \begin{bmatrix} H_L(u^d) \\ H_L(y^d) \end{bmatrix} \alpha = \begin{bmatrix} u \\ y \end{bmatrix}.$$

## Data-Driven Safety Filter (DDSF) [1]

### Classical Safety Filter

$$\min_{u_{[0, N-1]}} \|u_0(t) - u_l(t)\|_R^2$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k,$$

$$x_0(t) = x_0,$$

$$x_k(t) \in \mathcal{X} \quad \forall k \in [0, N]$$

$$u_k(t) \in \mathcal{U} \quad \forall k \in [0, N-1].$$

### Behavioral Safety Filter

$$\min_{\alpha(t), \bar{u}(t), \bar{y}(t)} \|\bar{u}_0(t) - u_l(t)\|_R^2 \quad \text{s.t.}$$

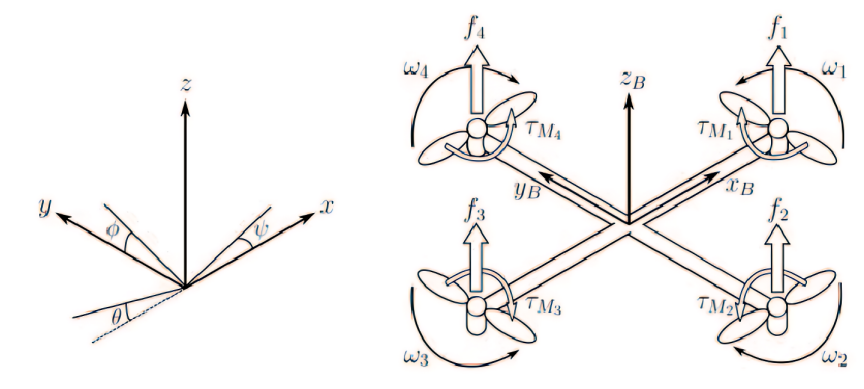
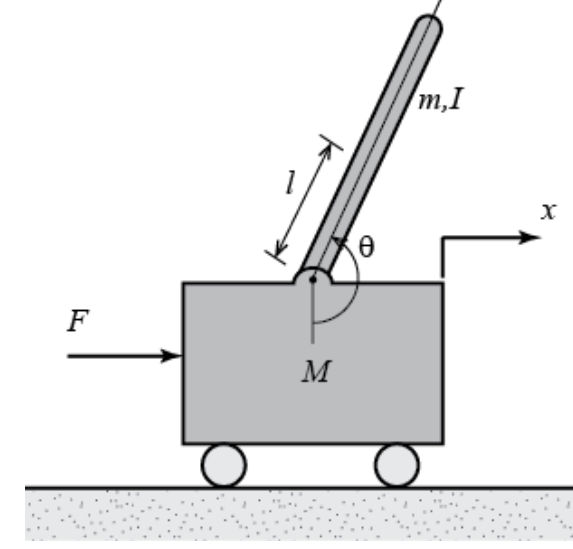
$$\begin{bmatrix} \bar{u}_{[-T_{\text{ini}}, N+T_{\text{ini}}-1]}(t) \\ \bar{y}_{[-T_{\text{ini}}, N+T_{\text{ini}}-1]}(t) \end{bmatrix} = \begin{bmatrix} H_L(u^d) \\ H_L(y^d) \end{bmatrix} \alpha(t),$$

$$\begin{bmatrix} \bar{u}_{[-T_{\text{ini}}, -1]}(t) \\ \bar{y}_{[-T_{\text{ini}}, -1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-T_{\text{ini}}, t-1]} \\ y_{[t-T_{\text{ini}}, t-1]} \end{bmatrix},$$

$$\bar{u}_k(t) \in \mathcal{U}, \bar{y}_k(t) \in \mathcal{Y} \quad \forall k \in [0, N-1],$$

$$\bar{u}_k(t), \bar{y}_k(t) \in S_f, \quad \forall k \in [N, N+T_{\text{ini}}-1].$$

### ► Systems: Inverted Pendulum, Quadrotor

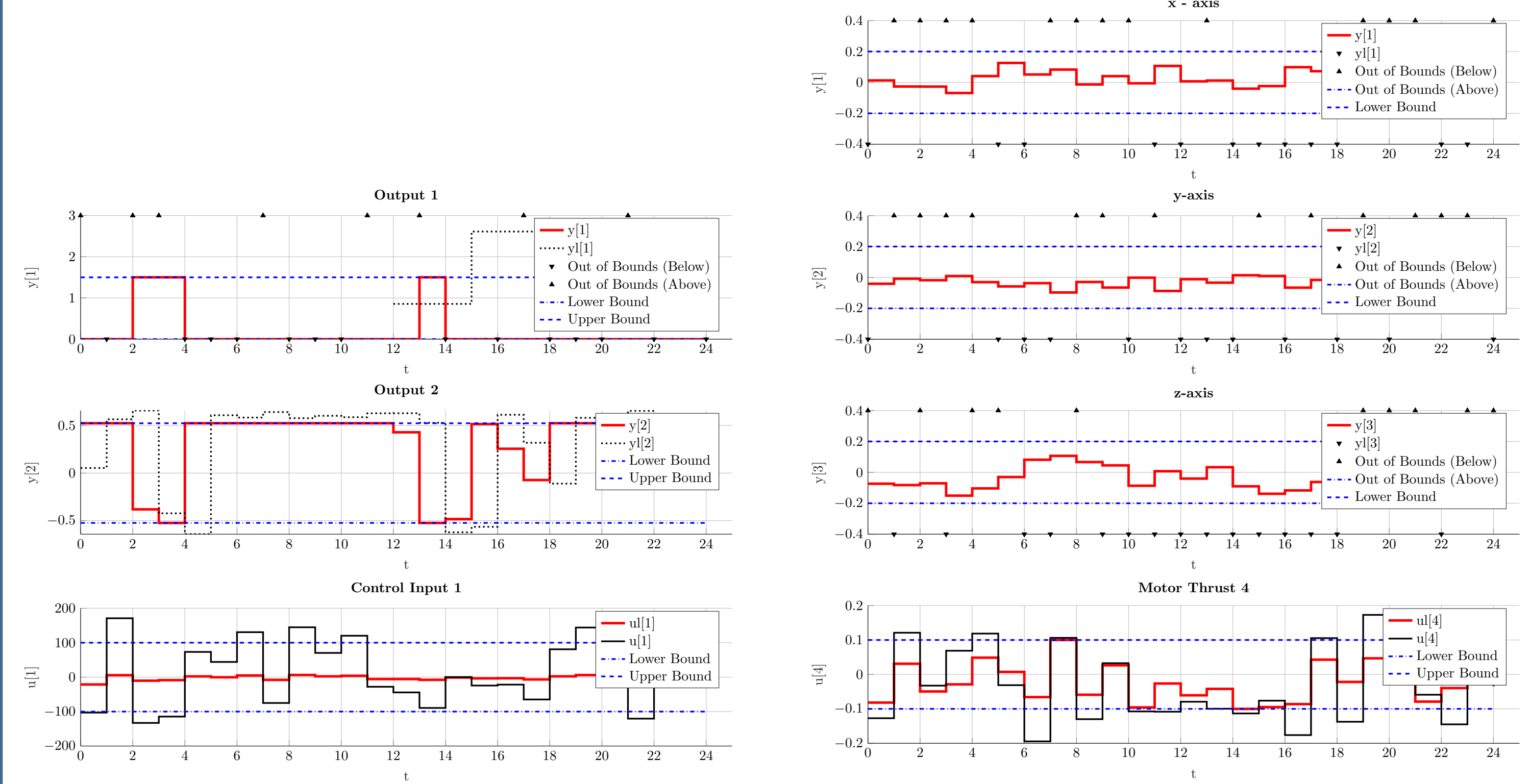


$$\vec{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}; u = F \in \mathbb{R}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \vec{x} = x \in \mathbb{R}$$

$$x = [\phi, \theta, \psi, p, q, r, u, v, w, x, y, z]^T,$$
$$u = [u_1, u_2, u_3, u_4]^T,$$
$$y = [\phi, \theta, \psi, x, y, z]^T$$

### ► Simulation Results



### Key Observations

- Overall, the DDSF effectively enforced safety limits
- The conservatism of the filter behavior varies with system complexity.

## Outlook

- Feasibility:** Translating classical control algorithms into the behavioral paradigm proved achievable
- Limitations**
  - Neither of the algorithms was tested extensively on nonlinear systems
  - Constrained by numerical stability issues
  - Continuing dependency on the state-space representation in the computation of system lag and equilibria
- Future Work**
  - Address numerical challenges
  - Extend the framework to nonlinear systems
  - Facilitate truly data-driven methodologies
  - Conduct robustness tests under progressively increasing levels of perturbation and time delay

## Data-Enabled Predictive Control (DeePC) [2]

### Classical Optimal Tracking Objective

$$\min_{u, y} \sum_{k=0}^{N-1} (\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2)$$

$$\text{s.t. } x_0 = \hat{x}(t),$$
$$x_{k+1} = Ax_k + Bu_k,$$
$$y_k = Cx_k + Du_k, \quad u_k \in \mathcal{U}, y_k \in \mathcal{Y} \quad \forall k \in [0, N-1]$$

### System vectors:

- (System state)  $\vec{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$ , where  $x$  is the vertical displacement
- (Input)  $u = F$ , force applied to the mass-spring damper system
- (Output)  $y = [1, 0]^T \cdot \vec{x} = x$ , vertical displacement
- System dimensions:  $n = 2, p = 1, m = 1$

### Input Constraints:

- Control inputs  $u \in [0, 100] \text{ [N]}$

### Output Constraints:

- System outputs  $y \in [-10, 10] \text{ [m]}$

### Initial and Target States:

- Initial State:  $x_{\text{ini}} = \begin{bmatrix} 9 & 2 \end{bmatrix}^T \text{ [m]}.$
- Target State  $y_{\text{target}} = 0 \text{ [m]}.$

### Filter Parameters:

- (Initial trajectory length)  $T_{\text{ini}} = 1.$
- (Prediction horizon)  $N = 10.$
- (Output weight matrix)  $Q = 100 \cdot \mathbf{I}_{2 \times 2}.$
- (Input weight matrix)  $R = 0.01 \cdot \mathbf{I}_{1 \times 1}.$

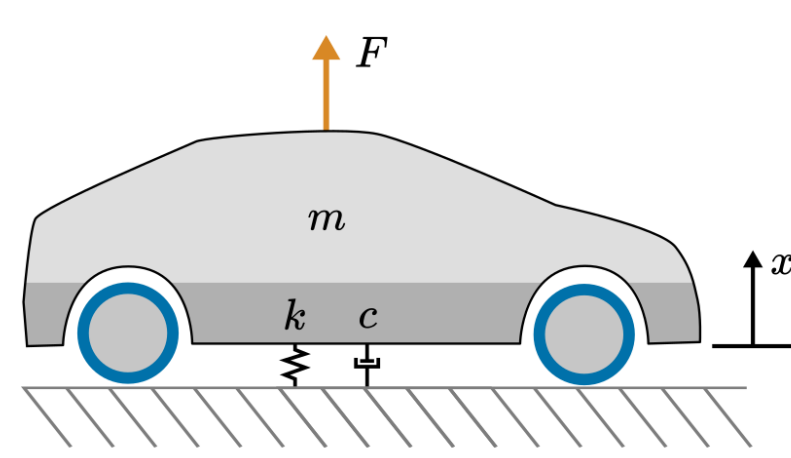
### ► DeePC Control Objective

$$\min_{g, u, y} \sum_{k=0}^{N-1} (\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2)$$

$$\text{s.t. } \begin{pmatrix} U_p \\ Y_p \\ U_f \\ Y_f \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{pmatrix},$$

$$u_k \in \mathcal{U}, y_k \in \mathcal{Y} \quad \forall k \in [0, N-1]$$

### ► System: Mass-Spring Damper

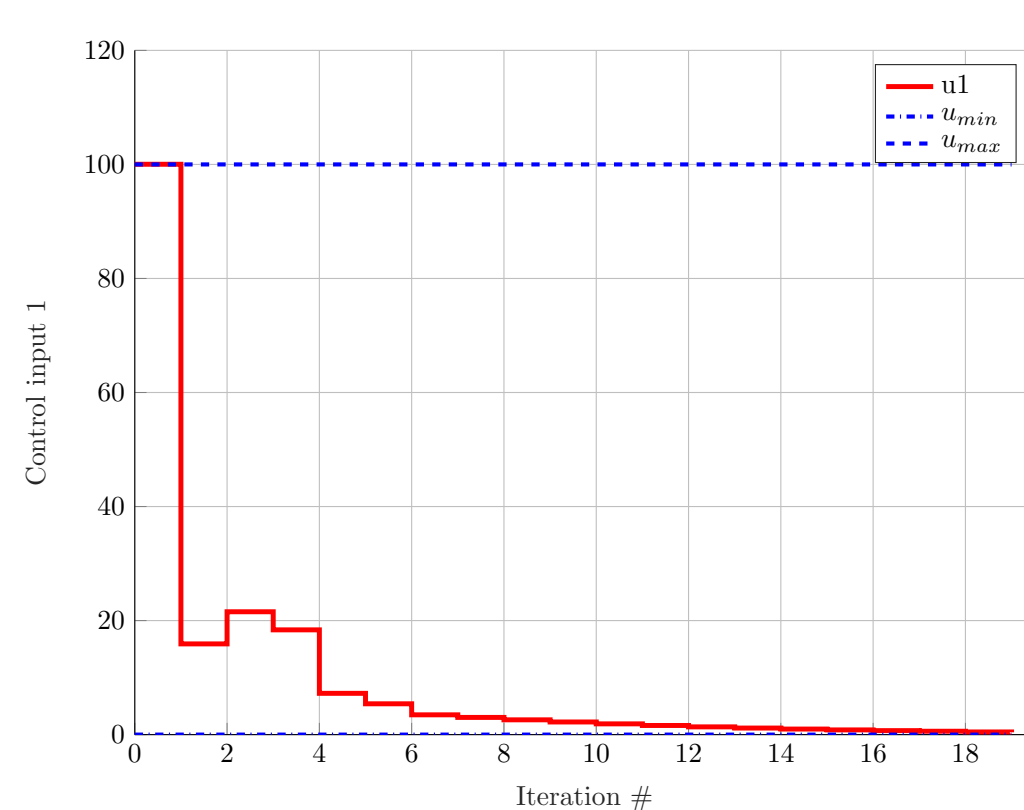


$$A = \begin{bmatrix} 1 & dt \\ -\frac{k \cdot dt}{m} & 1 - \frac{b \cdot dt}{m} \end{bmatrix},$$

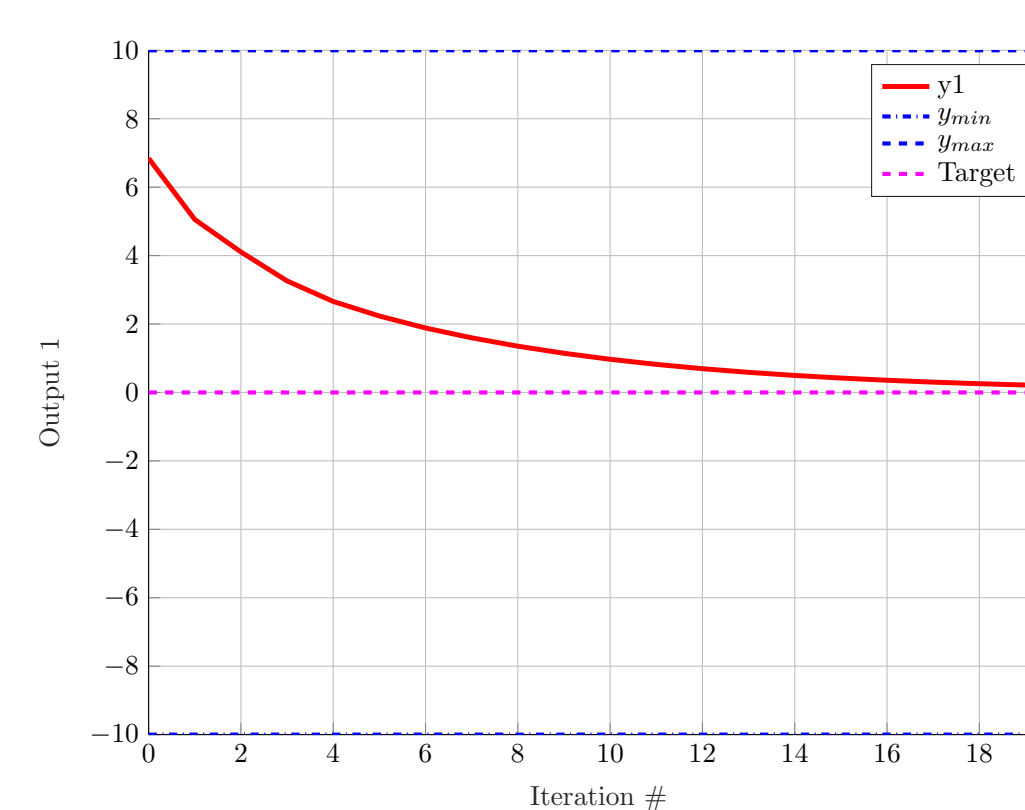
$$B = \begin{bmatrix} 0 \\ \frac{dt}{m} \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, D = 0.$$

### ► Simulation Results



(a) Control Inputs



(b) System Outputs

### ► The tracking problem is solvable in the BST framework

## KEY REFERENCES

- [1] Mohammad Bajelani and Klaske van Heusden. *Data-Driven Safety Filter: An Input-Output Perspective*. 2023.
- [2] Jeremy Coulson, John Lygeros, and Florian Dörfler. *Data-Enabled Predictive Control: In the Shallows of the DeePC*. 2019.
- [3] Jan C. Willems et al. "A note on persistency of excitation". In: *Systems Control Letters* 54.4 (2005), pp. 325–329.