Koopman-LQR Controller for Quadrotor UAVs from Data

Zeyad M. Manaa [1] Ayman M. Abdallah [1] Mohammad A. Abido [2] Syed S. Azhar Ali[1]

[1]Department of Aerospace Engineering, KFUPM, Dhahran 31261

[2]Department of Electerical Engineering, KFUPM, Dhahran 31261

IEEE International Conference on Smart Mobility (SM'24)
Crowne Plaza Niagara Falls-Fallsview, Ontario, Canada

September 16-18, 2024

Main Takeaway

Goal: To control highly unstable, nonlinear system (quadrotor) with the simplicity inherent in the linear control with global linear approximation.

Desired properties for our scheme:

1. **Global Linearity**. Converts nonlinear quadrotor dynamics into a globally linear model instead of local traditional linearization (e.g. Taylor linearization).
2. **Optimality**. Enables fast and computationally light optimal control design.
3. **Stability**. Ensures robust stabilization of highly unstable systems.

Main Takeaway

Goal: To control highly unstable, nonlinear system (quadrotor) with the simplicity inherent in the linear control with global linear approximation.

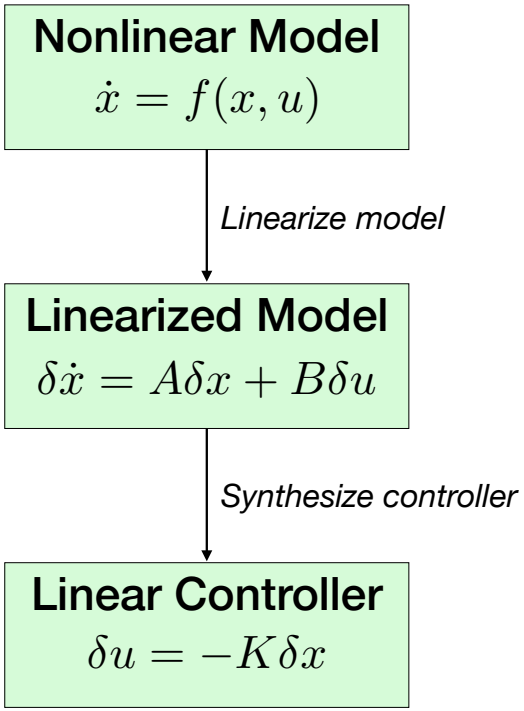
Desired properties for our scheme:

1. **Global Linearity**. Converts nonlinear quadrotor dynamics into a globally linear model instead of local traditional linearization (e.g. Taylor linearization).
2. **Optimality**. Enables fast and computationally light optimal control design.
3. **Stability**. Ensures robust stabilization of highly unstable systems.

Optimal Control Workflow

Optimal control is very famous in robotics and aerospace applications. One possible work flow would be:

1. **derive a nonlinear model** from the first principles
2. **linearize** the model about an equilibrium point or trajectory, then
3. synthesize a controller on top of the linearized model by posing an **optimal problem.**



Optimal Control Workflow

Advantages

- Leverages **knowledge of dynamics**.
- Is systematic and easy to tune.
- Has vast literature, with extensions like
  1. robust H_∞ control[1] and
  2. gain scheduling[2].

Disadvantages

- Relies on knowledge of dynamics. But
  1. model parameters and
  2. model form are **not exactly known**.
- Synthesizes a linear controller for a nonlinear system.

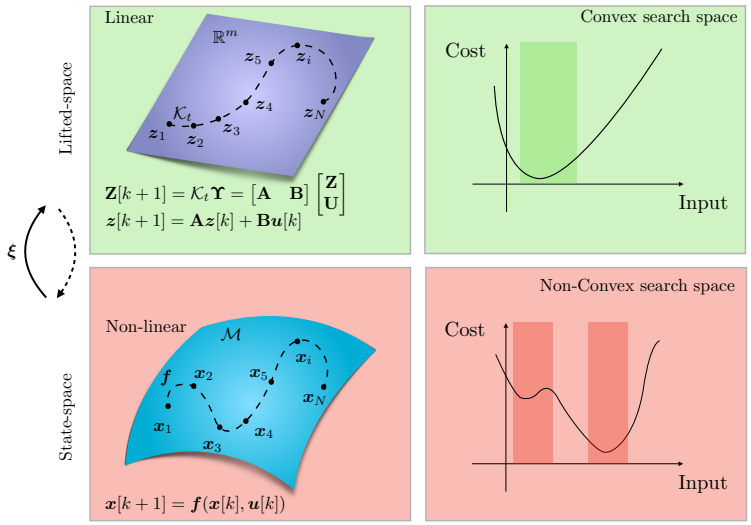
Enhancing the model can enhance the controller!

[1] D Jc et al. "State-space solutions to standard h/sub 2/and h/sub infinity/control problems". In: *IEEE Transactions on Automatic Control* 34.8 (1989), pp. 831–847.

[2] Wilson J Rugh and Jeff S Shamma. "Research on gain scheduling". In: *Automatica* 36.10 (2000), pp. 1401–1425.

The Koopman Operator

The **Koopman operator**[3] provides A linear representation of a nonlinear system in **infinite-dimensional** space.



Use **data-driven methods** to approximate a **finite-dimensional** Koopman operator

[3] Bernard O Koopman. "Hamiltonian systems and transformation in Hilbert space". In: *Proceedings of the National Academy of Sciences* 17.5 (1931), pp. 315–318.

Motivating Toy Example

$$\dot{x}_1 = \mu x_1 + u_1 \tag{1a}$$
$$\dot{x}_2 = \lambda(x_2 - x_1^2) \tag{1b}$$

If we choose the right **observables** such that

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \end{bmatrix} \rightarrow \frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \mu & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & 2\mu \end{bmatrix}}_{\hat{K}} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{B} u_1 \tag{2}$$

The system became **linear** but in a **higher** dimension!

Koopman Operator Theory

Consider the discrete time dynamical system:

$$x_k^+ = f(x_k, u_k),$$

and a *lifting function* $\xi : \mathbb{R}^n \mapsto \mathbb{R}$, where $\xi \in \mathcal{H}$. The Koopman operator $\mathcal{K}_t : \mathcal{H} \mapsto \mathcal{H}$ is an infinite-dimensional operator that advances the *lifting function* one time step such as:

$$(\mathcal{K}_t \xi)(x_k, u_k) = (\xi \circ f)(x_k, u_k) = \xi(x_{k+1}),$$

the dynamics are therefore equivalent to

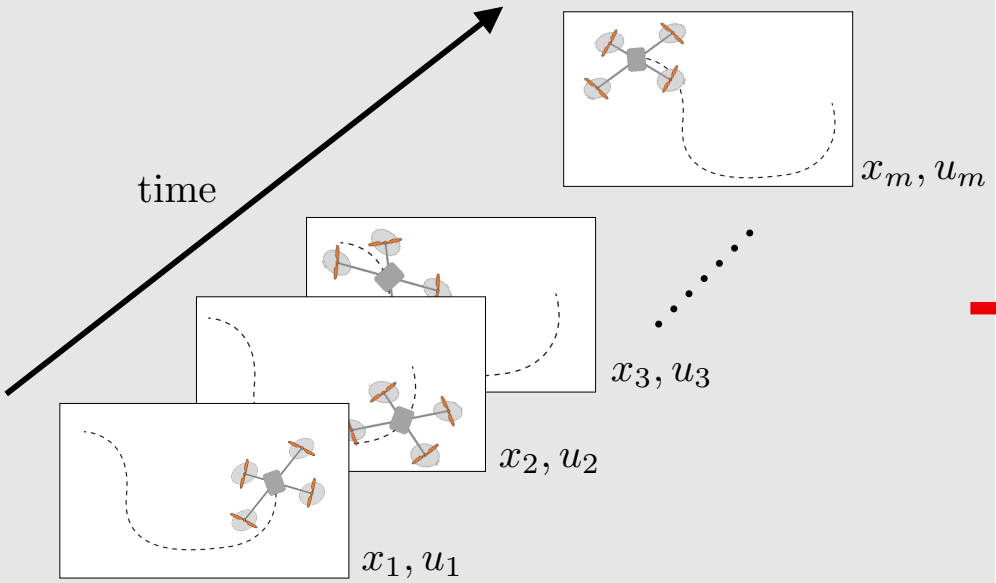
$$\xi(x_{k+1}, u_{k+1}) = (\mathcal{K}_t \xi)(x_k, u_k),$$

and in **finite-dimensional** form,

$$\xi(x_{k+1}, u_{k+1}) = (\hat{\mathcal{K}}_t \xi)(x_k, u_k) + r_k.$$

Approximating Koopman Operator using Dynamic Mode Decomposition

Numerical simulation (or) experiment



Snapshots

Collect data

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_{m-1} \\ | & | & & | \end{bmatrix}$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ x_2 & x_3 & \dots & x_m \\ | & | & & | \end{bmatrix}$$

$$\boldsymbol{\Upsilon} = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_{m-1} \\ | & | & & | \end{bmatrix}$$

Fit Data

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} + \mathbf{B}\boldsymbol{\Upsilon}$$
$$\approx \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Upsilon} \end{bmatrix} = \mathbf{K}\boldsymbol{\Omega}$$

$$\min_{\mathbf{K}} ||\mathbf{X}' - \mathbf{K}\boldsymbol{\Omega}||_{\mathrm{F}}$$

or

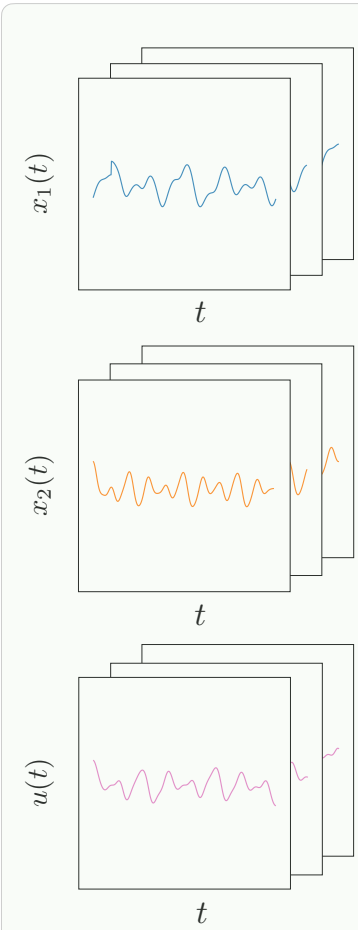
$$\min_{\mathbf{K}} ||\mathbf{X}' - \mathbf{K}\boldsymbol{\Omega}||_{\mathrm{F}} + \lambda||\mathbf{K}||_{*}$$

$$||\boldsymbol{\Delta}||_{*} = \mathtt{trace}(\sqrt{\boldsymbol{\Delta}\boldsymbol{\Delta}})$$

$$||\boldsymbol{\Delta}||_{\mathrm{F}} = \sqrt{\sum_{j=1}^{n}\sum_{k=1}^{m} \boldsymbol{\Delta}_{ik}^2}$$

The Entire Work Flow

(1) Collect data

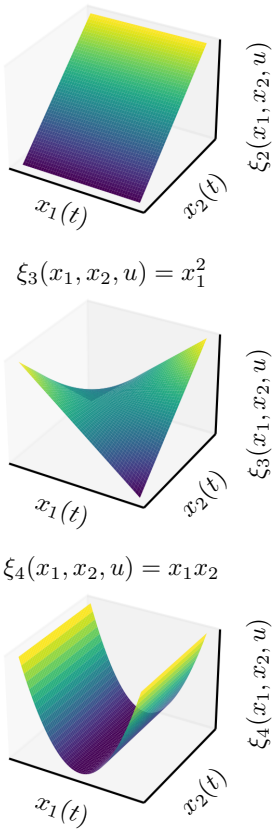


(2) Select lifting functions

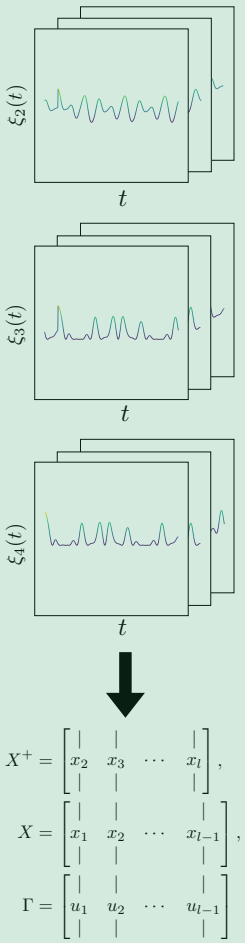
$\xi_2(x_1, x_2, u) = x_2$

$\xi_3(x_1, x_2, u) = x_1^2$

$\xi_4(x_1, x_2, u) = x_1 x_2$



(3) Lift and arrange snapshots



$$X^+ = \begin{bmatrix} | & | & & | \\ x_2 & x_3 & \cdots & x_l \\ | & | & & | \end{bmatrix},$$

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_{l-1} \\ | & | & & | \end{bmatrix},$$

$$\Gamma = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_{l-1} \\ | & | & & | \end{bmatrix}$$

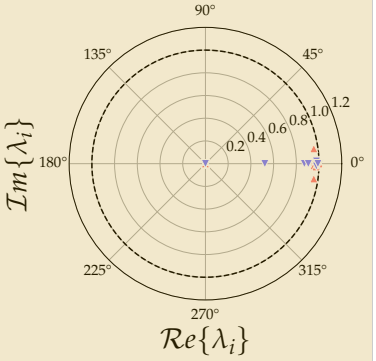
(4) Approximate Koopman

Regression using DMD

$$\Xi(X^+) = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \Xi(X) \\ \Gamma \end{bmatrix} = \hat{\mathcal{K}}_t \bar{\Omega},$$

$$\hat{\mathcal{K}}_t = \arg\min_{\hat{\mathcal{K}}_t} \|\Xi(X^+) - \hat{\mathcal{K}}_t \bar{\Omega}\|_{\mathrm{F}}^{\dagger}$$

$$^{\dagger}\|\mathbf{\Delta}\|_{\mathrm{F}} = \sqrt{\sum_{j=1}^{n}\sum_{k=1}^{m} \mathbf{\Delta}_{ik}^2}$$



Koopman Meets LQR

Consider a quadratic cost function:

$$\mathcal{J} = \min_{u_0,\dots,u_{N-1}} \sum_{\tau=0}^{N-1} x^\top(\tau) Q x(\tau) + u^\top(\tau) R u(\tau),$$

If Koopman linearization is considered, the cost function still holds but with minor modifications as follows

$$\bar{Q} = \begin{bmatrix} Q_{n\times n} & 0_{p-n\times p-n} \\ 0_{p-n\times p-n} & 0_{p-n\times p-n} \end{bmatrix}_{p\times p}, \quad \bar{R} = R$$

It is possible to have a matrix L, and a control law in the form of $u = -Lx$ such that the cost $\mathcal{J}$ is minimum with a Koopman linearized dynamics.

Application to Quadrotor

Koopman-LQR can be applied to control and stabilize very nonlinear topoligies like quadrotors. Consider a quadrotor dynamics given by the following

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} p_{WB} \\ \dot{p}_{WB} \\ q_{WB} \\ \omega_B \end{bmatrix} = f(x,u) = \begin{bmatrix} p_W \\ \frac{1}{m}\, q_{WB} \cdot T_B + g_W \\ \frac{1}{2} q_{WB} \otimes \omega_B \\ J^{-1}\left(\tau_B - \omega_B \times J\omega_B\right) \end{bmatrix},$$

and

$$T_B = \begin{bmatrix} 0 \\ 0 \\ \sum T_i \end{bmatrix} \quad \text{and} \quad \tau_B = \begin{bmatrix} l(-T_0 - T_1 + T_2 + T_3) \\ l(-T_0 + T_1 + T_2 - T_3) \\ c_\tau(-T_0 + T_1 - T_2 + T_3) \end{bmatrix},$$

the goal is to derive a linear formulation of this system then stabilize it.

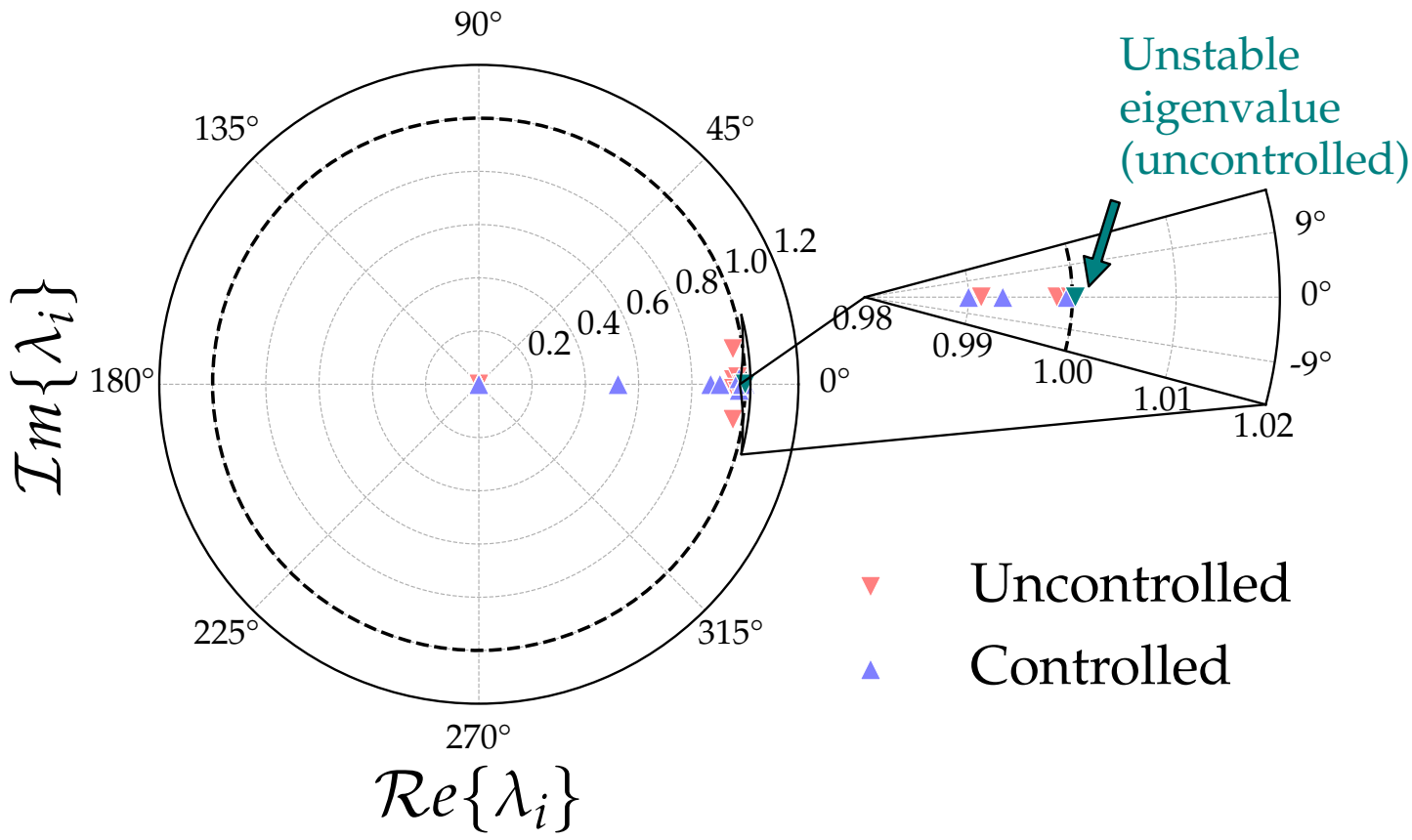
Koopman Approximation for Quadrotor

- We collected data by simulating the quadrotor following various **trajectories**
- **Randomization** was introduced to ensure **variability** in the dataset
- We used these simulated data to **train our Koopman model**
- We employed a set of **observable functions** based on our own **knowledge about the system** such as

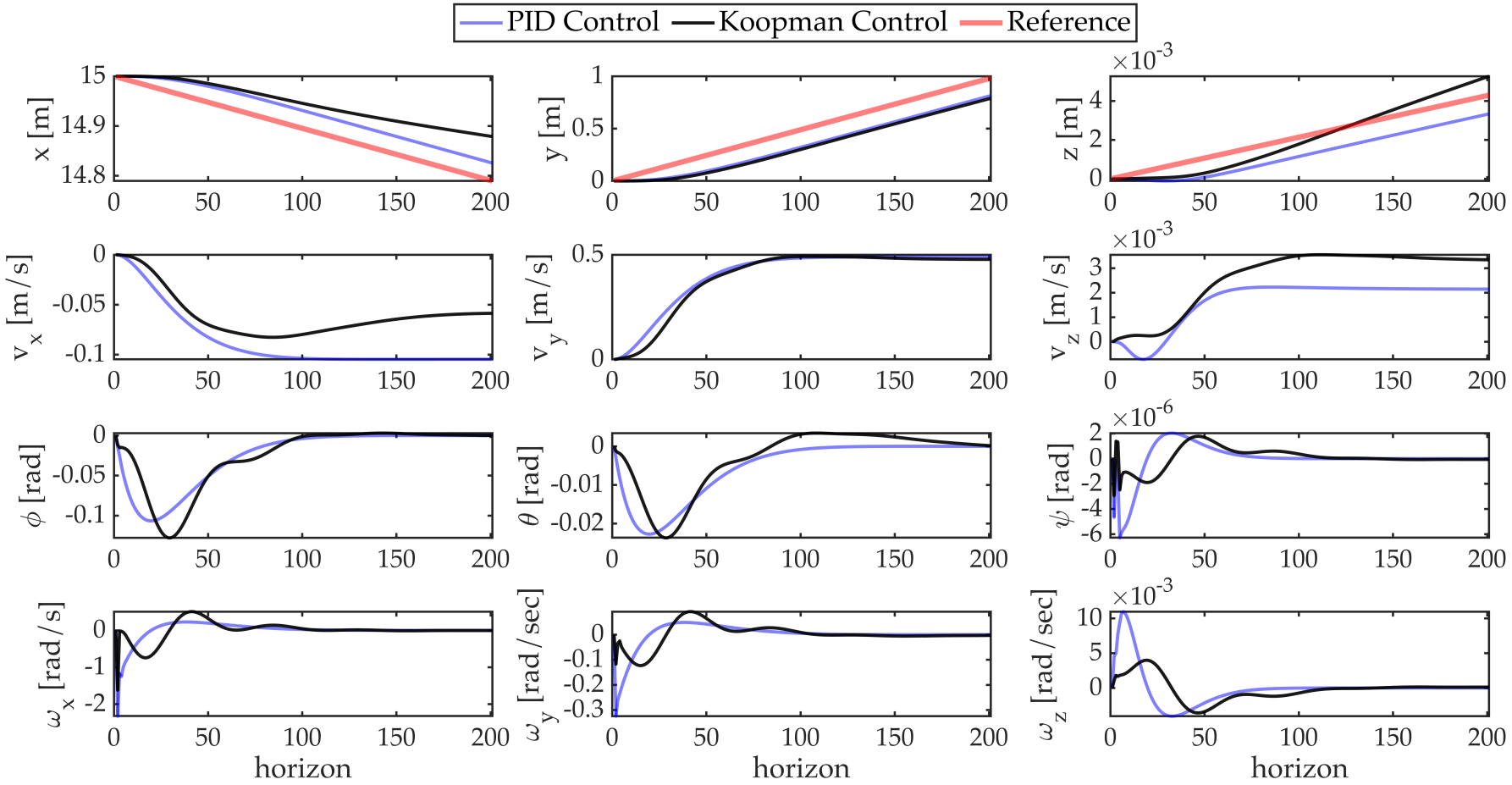
$$\Xi(x) = \left[1,\ x,\ p_{WB},\ \dot{p}_{WB},\ \sin(p_{WB}),\ \cos(p_{WB}),\ \text{vec}(R \times \omega_{WB})\right] \in \mathbb{R}^{39}.$$

- then, the **Koopman operator** is approximated using **DMD**.

Stability Analysis



Performance Analysis



Performance Analysis

The problem uses **39 observable functions**. Learning the Koopman operator on an **M2 MacBook Air** took **0.1393 seconds**. Inference over **200 timesteps** totaled **0.0035 seconds**, averaging $\mathbf{1.74 \times 10^{-5}}$ seconds per iteration.

states	%NRMSE
p_{WB} – Position	3.2529 ± 2.3216
$\dot{p}_{WB}$ – Velocity	4.8129 ± 3.8608
$\mathcal{E}_{WB}$ – Euler angles	2.4398 ± 2.4263
ω_B – Angular velocity	7.8525 ± 6.6367
Mean	4.5895 ± 3.8114

Future Directions

- **Further refining** the set of **observable functions** to improve **accuracy**.
- Extending this framework to handle **noisy or uncertain data**.
- **Testing the approach** on **physical quadrotors**.

Questions?