University of Stuttgart

Institute for Adaptive Mechanical Systems

Term paper
Master's thesis

Physics-Informed Learning Of The Koopman Generator

Topic Areas:	Nonlinear Dynamics, Machine Learning, Simulation
Advisors:	Maximilian Raff, raff@iams.uni-stuttgart.de
Responsible Professor:	Prof. C. David Remy
Prerequisites/Prior Knowledge:	Mechanical Engineering, Python or Matlab, Machine Learning

Finding optimal control laws for dynamical systems, such as legged robots, is particularly challenging due to their nonlinear dynamics $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$. If the system instead had linear dynamics, it would enable the effective design of globally optimal control policies.

The Koopman generator offers an approach to exactly linearize these dynamics by lifting the system into a higher-dimensional space, resulting in $\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{u}$, using a so-called observable function $\boldsymbol{\psi}(\boldsymbol{x})$ (Figure 1). Unfortunately, exact linearization generally requires $\boldsymbol{\psi}(\boldsymbol{x})$ to be infinite-dimensional. As a result, any finite-dimensional (i.e., truncated) choice of $\boldsymbol{\psi}(\boldsymbol{x})$ introduces model errors, which we aim to minimize [1].

Your task is to explore various strategies for constructing such approximate linear systems. This includes developing efficient sampling methodologies and identifying suitable observable functions $\boldsymbol{\psi}$ using tools from machine learning, such as neural networks [2].

To begin, you will implement an autonomous system ($\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$), as illustrated in Figure 1 (preferably in Python). A neural network will then be used to learn the observable function $\boldsymbol{\psi}$. Since it is often beneficial in control applications to maintain authority over physical states in the lifted space, e.g., to enforce path constraints, you will explore how these states can be explicitly incorporated into $\boldsymbol{\psi}$ using so-called physics-informed neural networks.
Finally, you will investigate how this methodology can be extended to nonlinear dynamics with inputs ($\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$).

Nonlinear Dynamics

$$\dot{x}_1(t) = x_1(t)$$
$$\dot{x}_2(t) = a x_1^2(t) + x_2(t)$$

$$\boldsymbol{z} = \boldsymbol{\psi}(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{z}$$

Linear Dynamics

$$\dot{\boldsymbol{z}}(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & a \\ 0 & 0 & 2a \end{bmatrix} \boldsymbol{z}(t)$$

Figure 1: Example of a Koopman Lift

[1] M. Abou-Taleb, et al., *Koopman Based Trajectory Optimization with Mixed Boundaries,* 7th Annual Learning for Dynamics& Control Conference. PMLR, 2025.

[2] Y. Han, W. Hao, U. Vaidya, *Deep learning of Koopman representation for control,,* 59th IEEE Conference on Decision and Control (CDC). IEEE, 2020.