Studying the Damped Harmonic Oscillator with Physics Informed Neural Networks

Daniele Montagnani

12 September 2024

Introduction

In this project I tested the Physics-Informed approach to training neural networks in a simple setting: learning the dynamics of the damped harmonic oscillator.

Reasons:

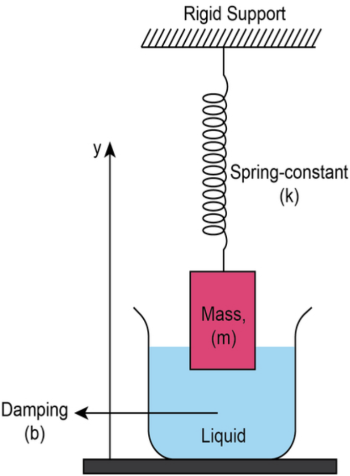
- simple and well understood dynamical system
- features interesting behaviors (oscillatory, exponential decay)
- defined in a one-dimensional space, easy visualisation

Damped Harmonic Oscillator I

Describe a system formed by a mass **m** attached to a spring of constant **k** whose movement is antagonised by a damper with coefficient **d**.
It is described by the following differential equation:

$$m\ddot{x} + d\dot{x} + kx = 0$$

.



Damped Harmonic Oscillator II

In order to have more insight during the ML workflow, I choose to perform a reduction of order by introducing an auxiliary variable $v = \dot{x}$.

Convert the second order differential equation to a system of first order ones:

$$\frac{d}{dt}\begin{pmatrix} x \\ v \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{pmatrix}\begin{pmatrix} x \\ v \end{pmatrix}$$

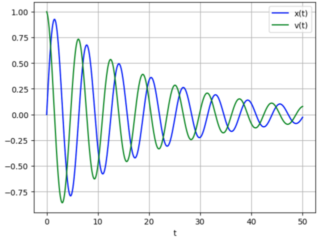
With general solution: $x(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t}$.

Obtain three qualitative behaviors:

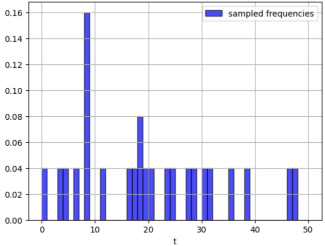
- overdamped
- critically damped
- **underdamped**

Simulation and Sampling

I simulated the system by considering an underdamped setting with parameters: $m = 1, d = 0.1, k = 1.$ Furthermore, I set initial conditions as: $x_0 = 0, v_0 = 1.$

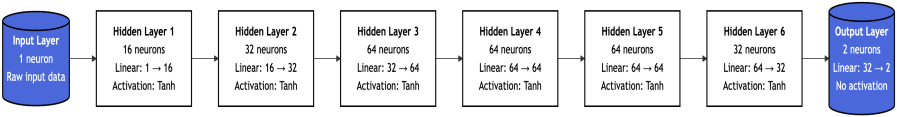


I wanted my model to operate in a "small data regime". So, I created my dataset by sampling 5% of the simulation data points uniformly over the simulated range



ML: Architectures

I trained neural networks on the system dynamics following different approaches: a Traditional Approach, a Pure Physics Informed Approach, and an Hybrid Approach.
For all three approaches I maintained the same network architecture: a simple multilayer perceptron with fully connected layers, upsampling up to 64 neurons, and a tanh activation.



ML: Loss Functions

I framed the problem as a $\mathbb{R} \to \mathbb{R}^2$ regression where I asked my models to predict both position and velocity from time.

- Traditional Loss:

$$L_{MSE}(\hat{v}, \hat{x}) = \frac{1}{N} \sum_{i=1}^{N} \left((x_i - \hat{x}_i)^2 + (v_i - \hat{v}_i)^2 \right)$$

- Traditional Loss:

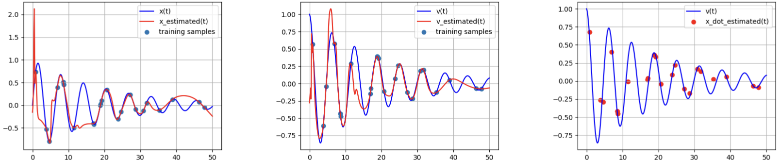
$$L_{PHYS}(\dot{\hat{v}}, \dot{\hat{x}}) = L_{MSE}(\hat{v}_0, \hat{x}_0) + \frac{1}{N} \sum_{i=1}^{N} \left(\left(\dot{\hat{x}}_i - \dot{x}_i \right)^2 + \left(\dot{\hat{v}}_i - \dot{v}_i \right)^2 \right)$$

- Hybrid Loss:

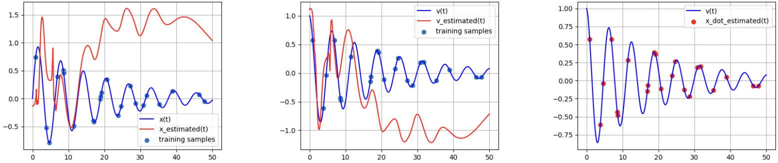
$$L_{PHYS}(\hat{v}, \hat{x}, \dot{\hat{v}}, \dot{\hat{x}}) = L_{MSE}(\hat{v}, \hat{x}) + L_{PHYS}(\dot{\hat{v}}, \dot{\hat{x}})$$

Comparing Performances

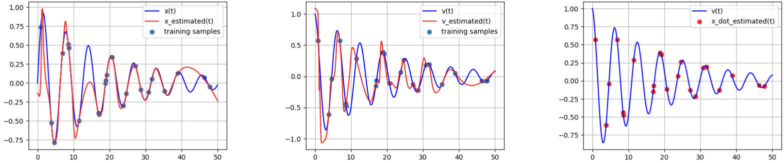
Traditional Approach



Pure Physics Informed Approach

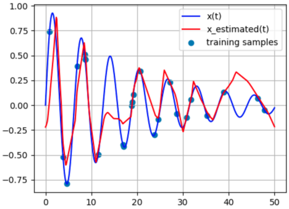


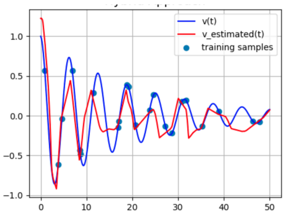
Hybrid Approach



Activation Function

It is important to impose a proper inductive bias by using a nonlinear activation function whose set of derivatives is “rich enough”.
In fact, by using a ReLu activation my network was performing considerably worse:





Data Regimes

In a setting where more data is available, the physics informed approach seems to lose its edge wrt the traditional approach.