Physics-Informed Trajectory Learning Using Neural Networks

Mohammad Ashifur Rahman
BSc in Physics (3rd Year), Khulna University

Abstract

This paper presents a novel framework for physics-informed trajectory optimization using neural networks, specifically designed for motion planning, drone navigation, and learning systems governed by physical laws. We construct a spatio-temporal deep neural network that infers dynamic paths while respecting Euler–Lagrange and Hamiltonian constraints. Applications range from robotics to embedded systems in autonomous vehicles.

1 Introduction

Trajectory planning problems appear in robotics, aerospace, autonomous vehicles, and smart electronics. Traditional approaches rely on explicit dynamical models or purely data-driven heuristics. Physics-Informed Neural Networks (PINNs) offer a powerful hybrid method: they encode physical laws (e.g., Lagrangian or Hamiltonian mechanics) into neural learning.

2 Model Overview

We propose a temporal physics neural network built with 1D CNNs that maps time and trajectory features to spatial paths. Let $\theta(t)$ be a coordinate trajectory, $\dot{\theta}$ and $\ddot{\theta}$ its derivatives.

2.1 Euler–Lagrange Residual

Let $M(t)$ be a learned mass matrix and $V(\theta)$ a learned potential. The Euler–Lagrange equation:

$$\frac{d}{dt}\Big(M(t)\dot{\theta}(t)\Big) + \frac{\partial V}{\partial \theta} = 0 \tag{1}$$

2.2 Hamiltonian Conservation

We define total energy $H(t)$:

$$H(t) = \frac{1}{2}\sum_i \frac{(M_i\dot{\theta}_i)^2}{M_i} + V(\theta) \tag{2}$$

We enforce conservation: $\frac{dH}{dt} \approx 0$

2.3 Brachistochrone Cost

Time-optimal paths satisfy:

$$\mathcal{L}_{BC} = \int \frac{ds}{\sqrt{2gy}} \quad \text{where } ds^2 = dx^2 + dy^2 \tag{3}$$

3 Learning and Optimization

Using PyTorch, we train the network with gradients of physics-based losses:

- $\mathcal{L}_{EL}$: Euler–Lagrange loss
- $\mathcal{L}_{H}$: Hamiltonian conservation
- $\mathcal{L}_{BC}$: Brachistochrone time cost
- $\mathcal{L}_{RC}$: Relation constraint from clustering ($\theta^T R\theta$)

4 Applications

- Embedded navigation systems in drones
- Optimized control in robotic arms
- Edge devices with energy-efficient pathing
- Smart electronics (IoT) with time-optimal coordination

5 Conclusion

This framework integrates physical laws with neural function approximation to achieve interpretable, efficient, and generalizable path learning.