ECE 445 Project Proposal - Spring 2019

Virtually Trained Self-Balancing Pendulum

Group 31: Kishora Adimulam, Henry Thompson, Mason Ryan
TA: Amr Martini

1. Introduction

i. Objective

There is a growing use for virtual reality as a training environment for AI for applications in the real world. Game engines like Unity have even released machine learning tool-kits for researchers and developers to experiment training AI inside games and simulations. There has been past work in translating these simulation-trained models to physical systems, such as the project done by OpenAI which taught a robot to stack different colored blocks in a specific order only after seeing it once in a virtual simulation [1]. However, there has been no past projects translating AI models trained in the Unity Game Engine to real physical systems.

Our solution would be to create a self-balancing inverted pendulum system, which would be trained as a simulation in Unity and uploaded into a physical system which then would gain the ability to balance the pendulum. Specifically, we would create a 3D simulation replicating all of the attributes of the physical system itself, and train the agent to learn to balance the pendulum using the Python API and Tensorflow. The trained Tensorflow model would then be uploaded into a Raspberry Pi, which would then use the control signals of the agent to operate motors to balance the real physical pendulum.

ii. Background

The ability of virtual reality to model physics and interactions between materials positions it to become an ideal tool for stimulating environments for artificial intelligence. This allows experiments to be carried out on a much larger scale, at a fraction of the time and resources required to carry out physical tests. Such an agent trained in a virtual simulation could be uploaded into an autonomous vehicle or factory robot, for example. Since the applications for virtual trained AI agents are numerous, it begs the question how easily such systems can practically be implemented using existing software. One of the most popular current game engines, Unity, has made it easy for developers to train their own AI agents using Tensorflow through the ML-Agents toolkit [2]. Furthermore, since Tensorflow 1.9 is supported by the Raspberry Pi, we aim to create a solution where the Tensorflow model of an agent trained on a 3D simulation in Unity can be deployed and perform in a physical replica of the system.

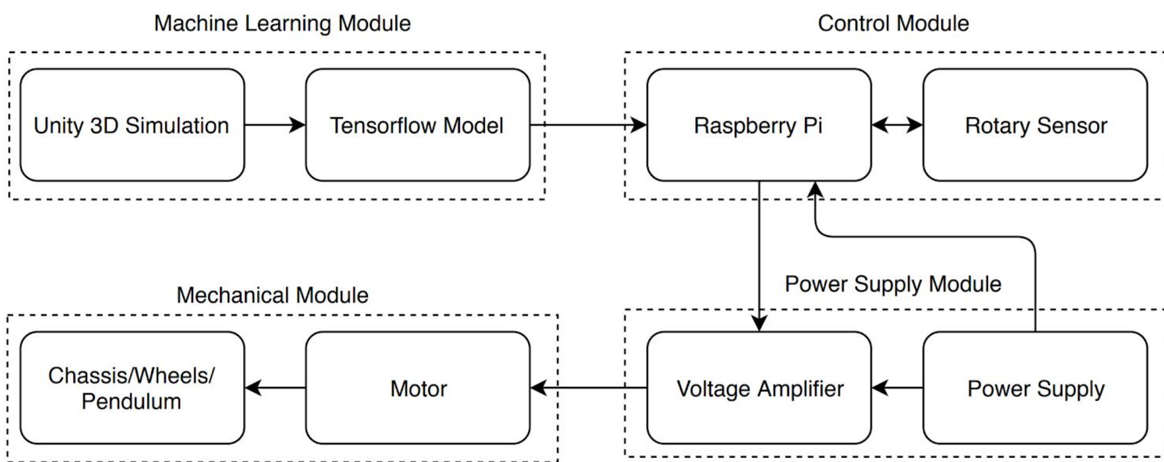
iii. High level Requirements

A. The agent in the Unity simulation must be able to successfully learn to balance the pole
B. The agent in the physical system must be able to balance the pole with no prior experience outside of the simulation

2. Design

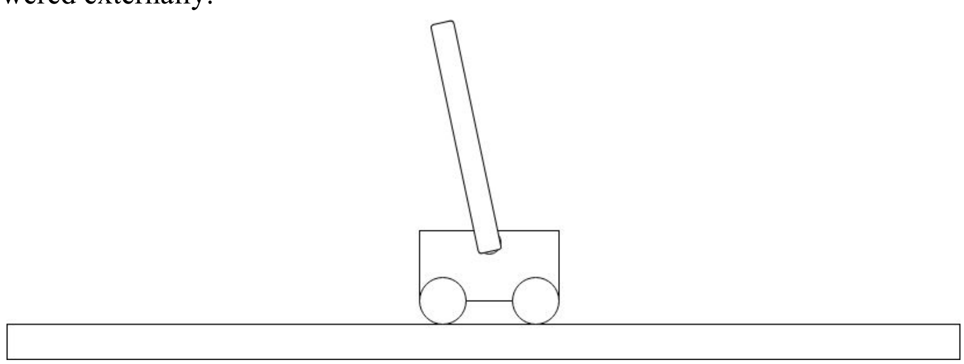
i. Block Diagram

There are four major subsystems in our project: the machine learning module, the control module, the mechanical module, and the power supply module. The machine learning module contains the Unity 3D simulation and the Tensorflow model produced from the simulation. The model will be uploaded to the control module where it can determine the appropriate control signals to send based off of this data. The control module consists of our microcontroller, a Raspberry Pi, and our rotary sensor. The Raspberry Pi will take in the sensor and Tensorflow model data, and will then determine what voltage it should supply to the motor in the physical system. The physical system consists of motor, cart and pendulum. It will be controlled by the Raspberry Pi, which will feed it input voltages to the motor through the voltage amplifier. The pendulum on the cart will then move according to the cart's velocity. The power supply module will supply power to the Raspberry Pi, but also to a voltage amplifier. The amplifier is needed to provide enough voltage for the motor, since the microcontroller has only a limited power output.



ii. Physical Design

The physical design will consist of a pendulum attached to a cart. The cart will be restrained to a one dimensional track. The Raspberry Pi and rotary sensor will be attached to the cart and they will be powered externally.



iii. Functional Overview of Modules

A. Machine Learning Module

- **Unity Simulation**

The simulation will be a 3D replica of our physical system, which will be a motor on a one-dimensional track, which can move in two directions and has an inverted pendulum attached. The brain of agent to be trained will be assigned to controlling the motor.
Requirement: Must be able to learn to balance the inverted pendulum with no prior instructions.

- **Tensorflow Model**

This model will be the output of the Unity simulation, and will be trained to balance the inverted pendulum using the Proximal Policy Optimization algorithm [3].
Requirement: Can be used by the virtual agent to balance the pendulum in simulation, as well as used by the Raspberry Pi to balance the physical pendulum.

B. Control Module

- **Rotary Sensor**

The rotary position sensor will be used to calculate the pendulum's angle and angular velocity. It will send a signal to the Raspberry Pi for analysis.
Requirement: The sensor requires a 4.5-5.5V source to function. The sensor must be small and attach to the pivot while minimally increasing the pivot's friction.

- **Raspberry Pi**

The Raspberry Pi will function as the project's microcontroller. It will take input data from the rotary position sensor and Tensorflow model to produce appropriate control signals to the motor.
Requirement: The Raspberry Pi needs 5.1V from a power supply to function properly.

C. Mechanical Module

- **Motor**

The motor will be used to turn the wheels so that the cart can move. It will be powered by the voltage amplifier which will be controlled by the control system. Using an H Bridge and the control signals the DC motor will either move at clockwise, counterclockwise, or will be off.
Requirement: The motor requires a 12V source. The motor must be able to rotate at 250 RPM with no load attached.

- **Chassis/Wheels/Pendulum**

The chassis of the cart will provide a pivot for the pendulum to be attached securely. It will hold the microcontroller, motor, sensor in place while the cart moves. The wheels will be attached to the motors and provide grip so that the cart can move.
Requirement: The chassis must be designed to house the motors and the sensors without inhibiting the carts movement. The wheels must provide enough grip to the track so that it moves the cart smoothly when the motor is on.

D. Power Module

○ **Power Supply**

The Power supply will be used to power the Raspberry Pi as well as the operational amplifier.
Requirement: must supply 5.1V DC at 2.5A to the Raspberry Pi. Must be able to provide enough volts to the op amp to function. Must provide 5V DC to the rotary sensor.

○ **Voltage Amplifier**

The voltage amplifier is required because the Raspberry Pi only outputs a voltage max voltage of 3 volts and we need a 12V DC signal to power the motor.
Requirement: The amplifier should take in a 3V signal from the Raspberry Pi and output a 12V signal to power the motors.

iv. Risk Analysis

The interface between the Raspberry Pi and the Tensorflow model will pose the greatest risk to successful completion of the project. While there is provided support for running Tensorflow on a Raspberry Pi, the task of mapping the input state of the system to the model in a format which it will understand will be a challenge that we will have to overcome. Furthermore, mapping the output of the model to the correct control signals output by the Raspberry Pi will be the other side of the challenge, to ensure that the model will seamlessly integrate with our physical system.

3. Ethics and Safety

The ethical and safety issues in our project mainly pertain to the safety of the different moving mechanical components in the design. Since there is movement of a cart on a track and the attached swinging pendulum, this poses a safety risk for anyone standing too close or putting their body parts in the system. During operation, the cart and pendulum could cause injury to someone too close to the system.

The safety precautions we would take to handle these situations refer to #9 on the IEEE Code of Ethics, to "avoid injuring others, their property, reputation, or employment by false or malicious action" [4]. To prevent the motor from hurting someone's fingers, we would add a rubber bumper to each side of the motor, as well as bumpers to the end of the track to prevent the motor from flying off. We would also ensure that the heavy end of the pendulum is rounded and not sharp, to avoid the risk of serious injury if it was to strike anyone standing too close. Lastly, we will be able to wirelessly connect to the Raspberry Pi to start and stop operation of the system, enabling everyone to be a safe distance away during the entirety of operation.

4. References

[1] Clark, Jack. “Robots That Learn.” *OpenAI Blog*, OpenAI Blog, 28 Nov. 2017, blog.openai.com/robots-that-learn/.

[2] Juliani, Arthur. “Introducing: Unity Machine Learning Agents Toolkit – Unity Blog.” *Unity Technologies Blog*, 19 Sept. 2017, blogs.unity3d.com/2017/09/19/introducing-unity-machine-learning-agents/.

[3] OpenAI. “Proximal Policy Optimization.” *OpenAI Blog*, OpenAI Blog, 20 July 2017, blog.openai.com/openai-baselines-ppo/.

[4] “IEEE Code of Ethics.” *IEEE - Advancing Technology for Humanity*, www.ieee.org/about/corporate/governance/p7-8.html.