Bianca Champenois
Department of Mechanical Engineering
University of California, Berkeley
blc@berkeley.edu

Control of a Pendulum Energy Converter that is Excited by Ocean Waves

Abstract

Rotating pendulums can generate energy when they are vertically excited at the correct frequency. One solution is to use ocean waves as a source of vertical excitation. Because the frequency of ocean waves is random and constantly changing, we can achieve the desired output energy of the pendulum by controlling its motion with a second actuated pendulum arm. In this project, a reinforcement learning algorithm is tested as a control method.

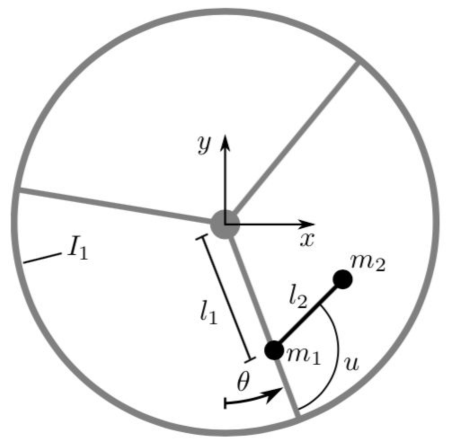

Figure 1. Diagram of the Acrobot pendulum with the actuated arm (l_2, m_2).

Introduction

In a world with a rapidly increasing demand for energy, we need to find new ways to harvest clean and renewable energy. Ocean gravity waves can be used to vertically excite a pendulum that can generate electricity via electromagnetic induction. The energy from the pendulum can be maximized when the frequency of the vertical excitation and the frequency of the pendulum achieve parametric resonance.

The frequency of ocean waves is continuously and stochastically changing. To make up for this, we can control the frequency of the pendulum with a second actuated pendulum arm. In this project, I study two control techniques: reinforcement learning and sliding mode control. The first technique calculates the probability of transitions between states and assigns rewards to the actions of the actuated arm based on the output energy. The second technique uses the known dynamics of the system to determine the best action to reach the desired output.

Figure 2. The system moves up and down on rails. This motion is controlled by a CompactRIO. The Raspberry Pi is directly mounted onto the pendulum so that the cables don't get tangled when it is rotating. The actuated arm is attached to a low-cost servo motor that is mounted on one of the arms of the pendulum.

Experimental Setup

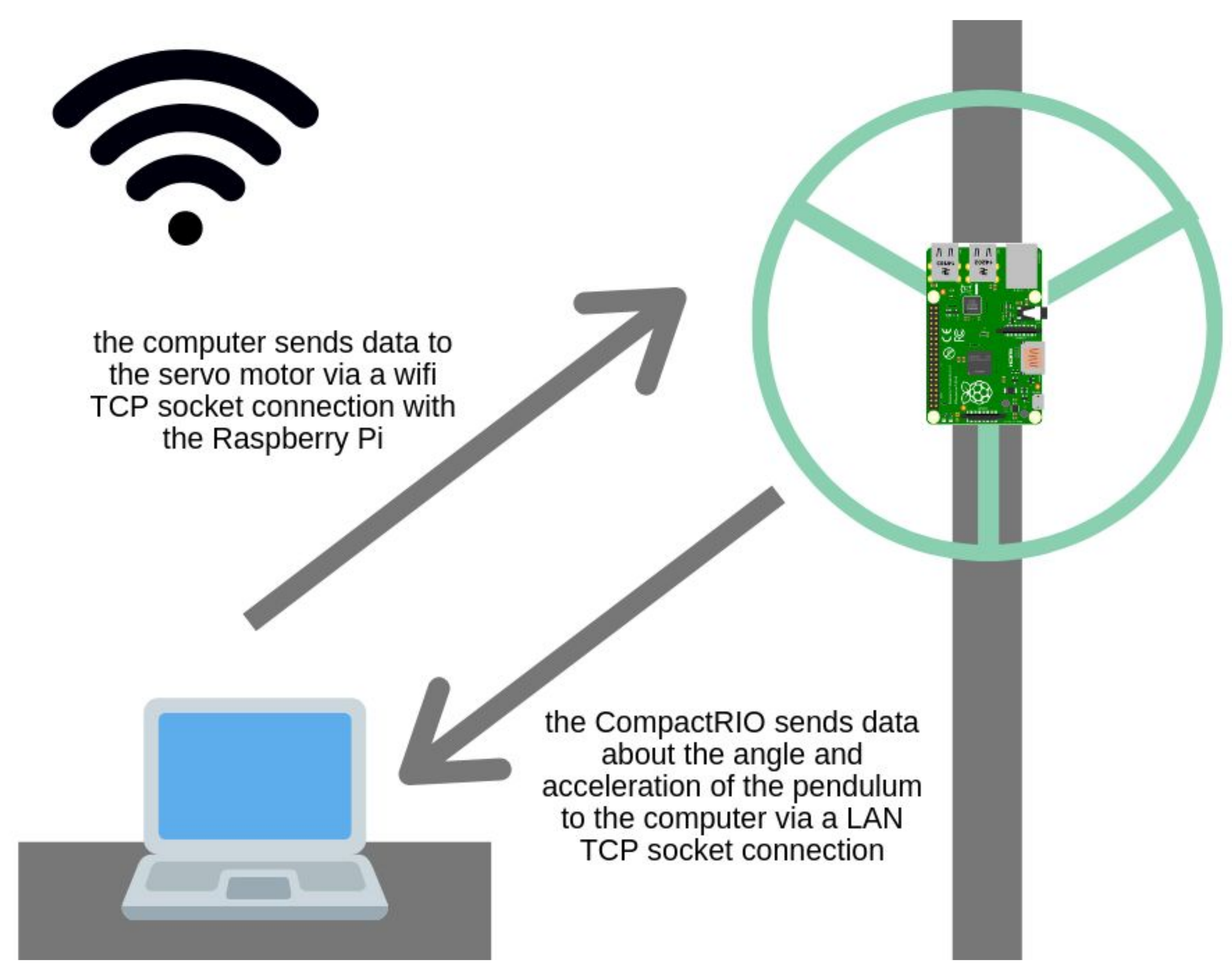

Figure 3. The main control algorithms are run on the computer. The computer sends instructions to the Raspberry Pi via a TCP connection over wifi to actuate the arm with the servo motor. The LabVIEW program actuates a motor that vertically excites the pendulum according to an input time series of desired position data. The LabVIEW program also sends data to the computer about the angle and angular velocity of the pendulum. This data is used to updates the states of the system in the control algorithms.

Results

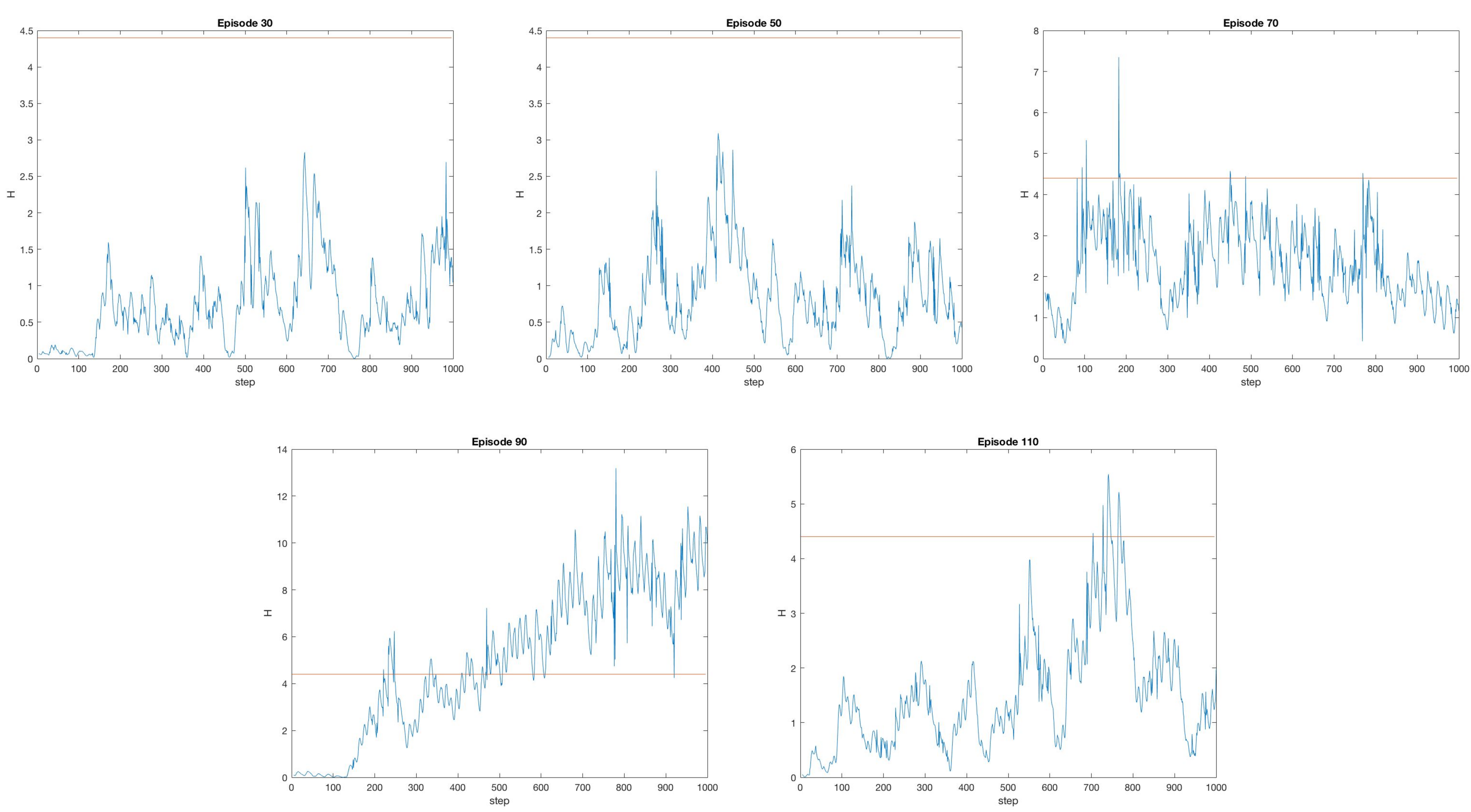

Figure 4. The experiment was run during 110 episodes that each last 1000 steps. The energy of each step in is plotted throughout the episode. During each step, the angle of the actuated arm is controlled by the reinforcement learning algorithm according to the transition matrices and calculated rewards. As the number of episode increases, the output energy gets closer to the desired energy (indicated by the red line). However, the pendulum is never able to perfectly achieve the desired output due to the stochastic nature of the simulated ocean waves.

Methods

I attached a servo motor to the second pendulum arm, and I connected it to the Raspberry Pi for easy control. I set up a TCP connection between the Raspberry Pi and the computer to send instructions to the servo motor over a wifi network.

Next, I set up a program in LabView to actuate a motor that vertically excites the pendulum according to an input time series of position data. The program also collected data on the angle and angular velocity of the pendulum. The CompactRIO sent this data to the computer via a TCP connection over a wifi network. The time series for the vertical excitation were generated from real models of the ocean.

I wrote Python scripts for the reinforcement learning and sliding mode control algorithms. The algorithms used the data from the sensors to control the actuated arm. The reinforcement learning algorithm adhered to the following procedure:

procedure RLMAIN
  while true **do**
    $s = \text{getState}(\theta, \dot{\theta})$
    $a = \text{selectNextAction}(s, v_k(s), \boldsymbol{P}(s'|s,a), \boldsymbol{R}(s,a))$
    perform action a
    $s' = \text{getState}(\theta, \dot{\theta})$
    get reward r
    $\text{updateSystemDynamicsModel}(r, s, s', \boldsymbol{C}(s, s', a), \boldsymbol{R}(s,a), \boldsymbol{R}_c(s,a))$
    if $s' = s_T$ **then**
      $\text{policyIteration}(v_k(s), \boldsymbol{P}(s'|s,a), \boldsymbol{R}(s,a))$
      re-initiate next episode
    end if
  end while
end procedure

Conclusions and Further Work

The reinforcement learning algorithm worked on a static system. However, it was unable to adapt to the stochastic vertical excitation. The pendulum approached the desired energy level, but it was never able to remain at the desired energy level during an entire episode. The parametric resonance would either make the pendulum undershoot or overshoot the desired energy level.

Given more time, I would run more experiments to see if the reinforcement learning algorithm needs more tests to correctly calculate the transition matrices. I would also test different discretizations of the states to see how this affects the learning.

In the future, my experimental setup can easily be used to test other control algorithms. The TCP connections between each part of the system allow for smooth adjustments. I would like to use the experimental setup to test neural networks, model predictive control, and other control strategies.

References

C. Cyr, L. Dostal, D. A. Dueker and E. Kreuzer, "Towards Reinforcement Learning-based Control of an Energy Harvesting Pendulum," *2019 18th European Control Conference (ECC)*, Naples, Italy, 2019, pp. 3934-3939.

Acknowledgements

I would like to thank Dr. Leo Dostal, Marten Hollm, and the rest of the Institute of Mechanics and Ocean Engineering at the Technische Universität Hamburg for providing me with the support and resources to conduct this project. This project was funded by The German Academic Exchange, or DAAD.