

# Hopper: Reinforcement Learning-Based Control on a Wheeled Bipedal Robot

Project Proposal

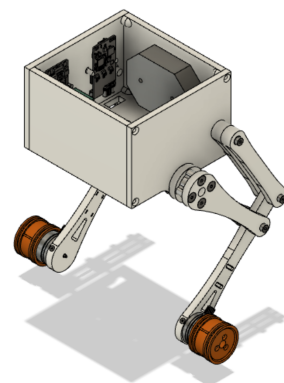
Category: Theory and Reinforcement Learning

Parthiv Krishna ([parthiv](#)) and Aaron Schultz ([azs](#))

CS 229 Spring 2021

## 1 Motivation

The problem that we intend to tackle is the application of reinforcement learning techniques to control Hopper, a small robot that the Extreme Mobility team within Stanford Student Robotics has been developing over the past several months. A CAD render of Hopper can be seen below.



Hopper is a wheeled bipedal robot; in other words, it moves by making use of driven wheels that are attached to the end of two actuated legs. To move effectively, Hopper must maintain its balance while also responding to the desired control input to reach a target position or velocity. Additionally,

Hopper can theoretically jump by rapidly extending its legs, and land after jumping by controlling the wheels to maintain stability.

We propose a project involving the application of a reinforcement learning-based control approach to the robot. As we intended for Hopper to be a platform to test and benchmark different control algorithms, we would like to compare the results obtained by reinforcement learning with those from more traditional control methods like PID.

## 2 Method

We will use reinforcement learning techniques during this project. To actually implement the project, we intend to use some form of physics simulator like PyBullet/OpenAI Gym with a simplified, but dynamically near-accurate, model of Hopper. After the completion of this project, we hope to test the algorithm that we develop on a physical robot as soon as we are able to be on campus. One example of prior research is:

Morimoto, J., et al. A Simple Reinforcement Learning Algorithm for Biped Walking. *IEEE International Conference on Robotics and Automation*, 2004. [doi:10.1109/robot.2004.1307522](https://doi.org/10.1109/robot.2004.1307522).

## 3 Intended Experiments

We plan to test the algorithm in a few different scenarios, which may include:

1. Balancing while staying still
2. Driving at some fixed velocity
3. Landing from a jump

We will evaluate our machine learning algorithm by comparing it to a traditional method of control like PID. We will compare these different algorithms on the basis of some chosen metrics, which may include:

1. Maximum angle of perturbation from which Hopper can return to stable balancing
2. Maximum height obstacle that Hopper can drive over
3. RMS error of Hopper's angle from vertical