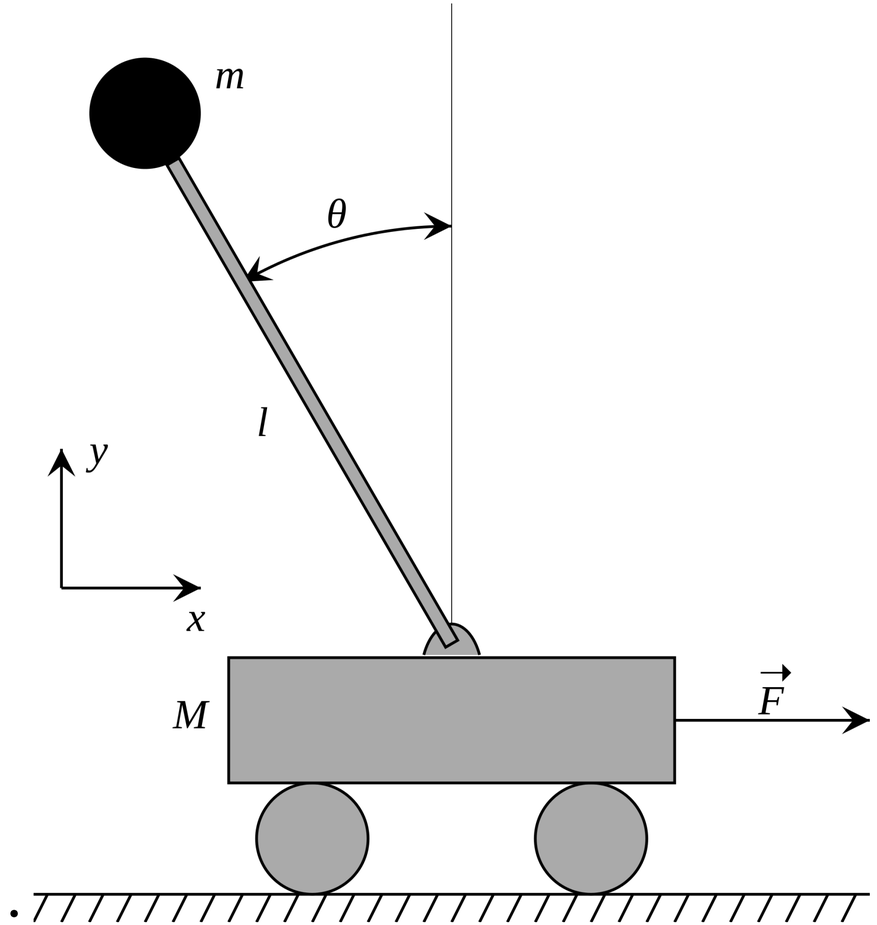
Balancing Bot
PID vs. RL

Chenning Yu (Solo)

Overview

- Problem
- Hardware - Car Chasis
- Software - Control Algorithm
  - PID
  - Reinforcement Learning
- Results
- Demo
- Lessons

Problem - Inverted Pendulum



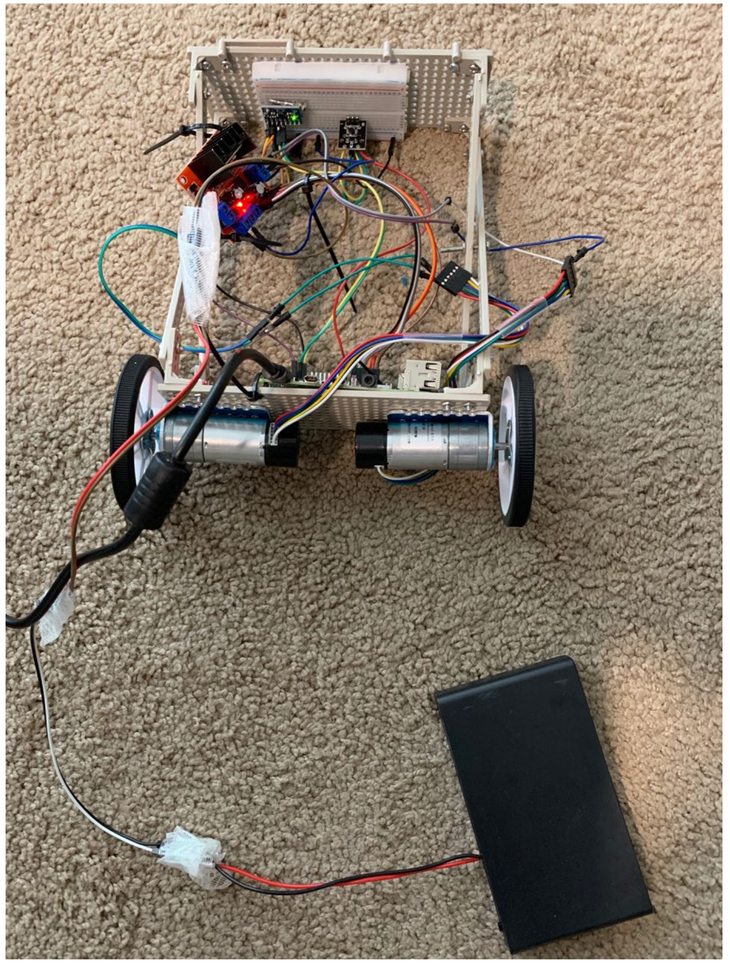
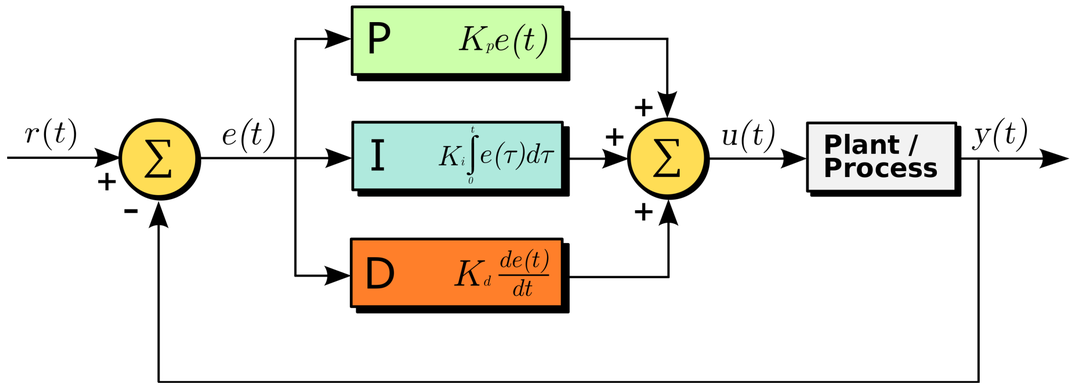


Image: https://en.wikipedia.org/wiki/Inverted_pendulum

Problem -
PID vs RL

- PID - not feasible to unstable & complex systems
- RL - not easy to experiment on noisy environments



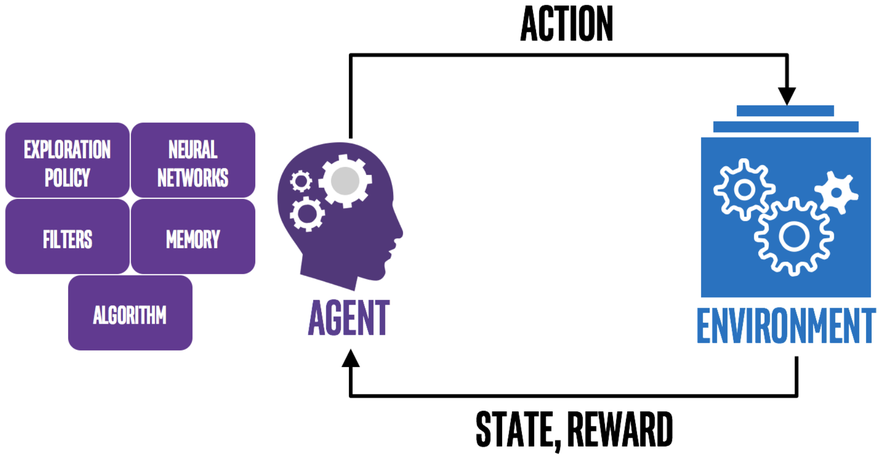
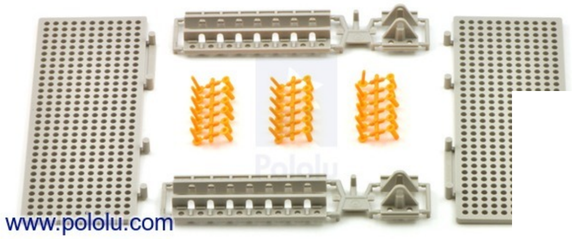
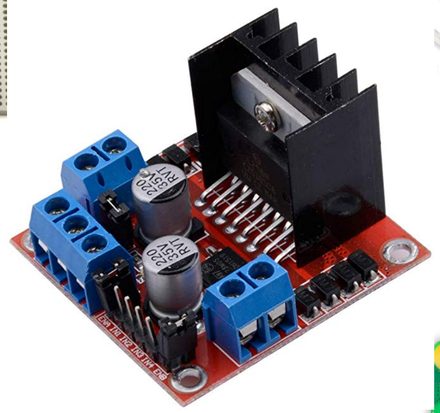


Image: https://en.wikipedia.org/wiki/PID_controller, https://medium.com/ai³-theory-practice-business/reinforcement-learning-part-1-a-brief-introduction-a53a849771cf

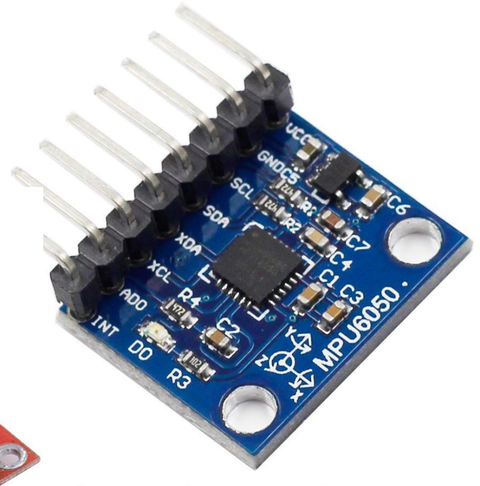
Hardware



Car Chasis

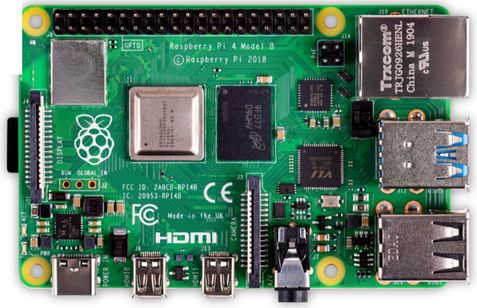


Micro Controller



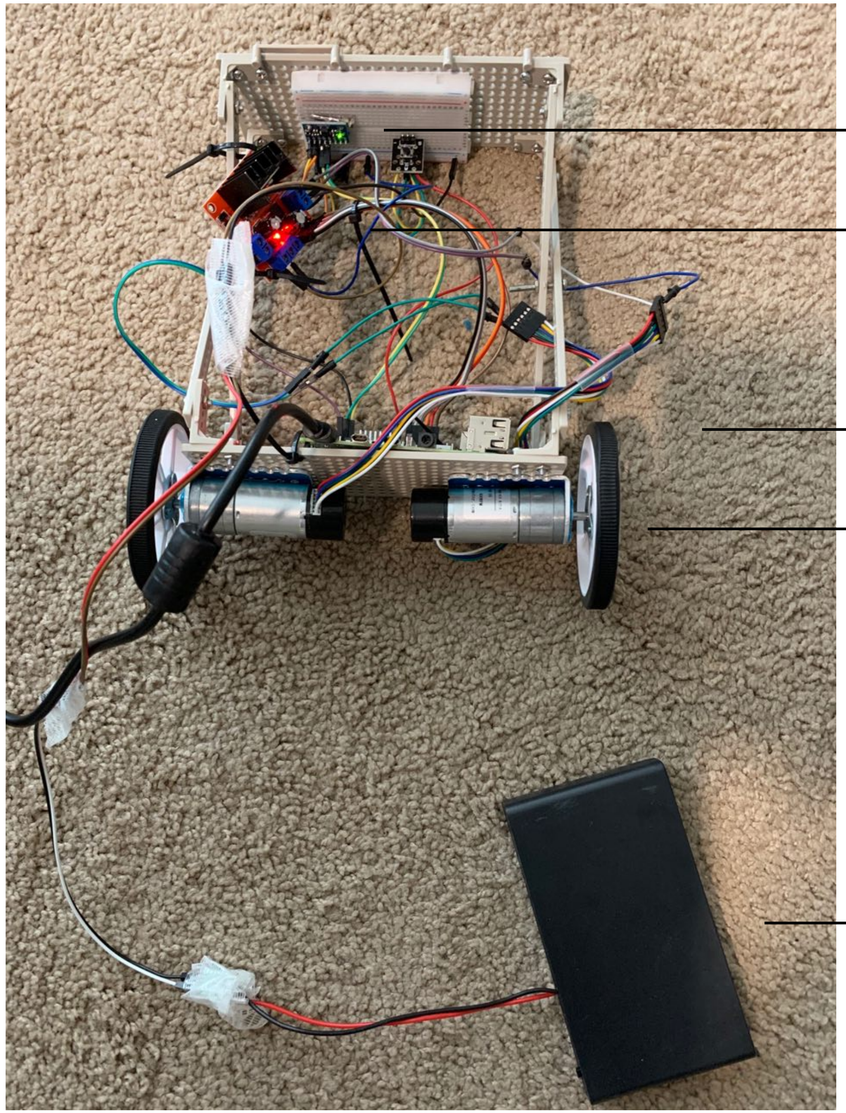
Gyro & Accelerator

Motor x2



RPi 4

Put It Together



MPU 6050 & Button

Microcontroller

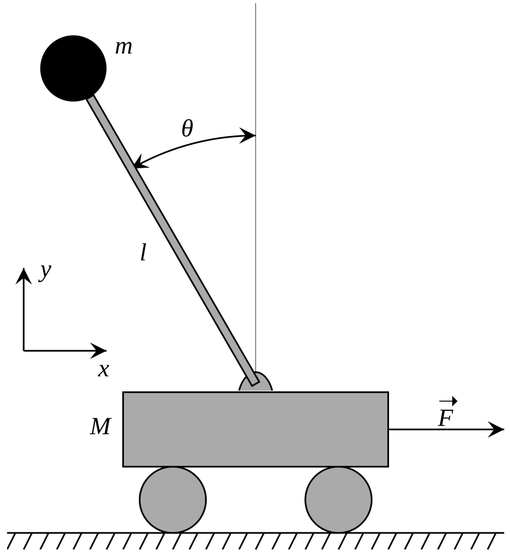
RPi 4

Motors & Wheels

Battery

Software - PID

- Input: angle & angular acceleration
- Program:
  - Estimate angle using complementary filter
  - Calculate PID term based on errors
- Output: speed [-100, 100]
- Ku: 3.0, Tu: 2 seconds
- Use PD instead of PID



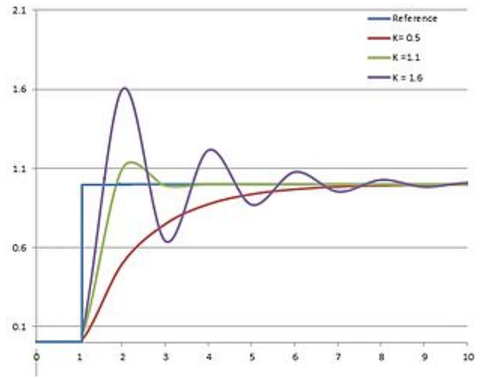
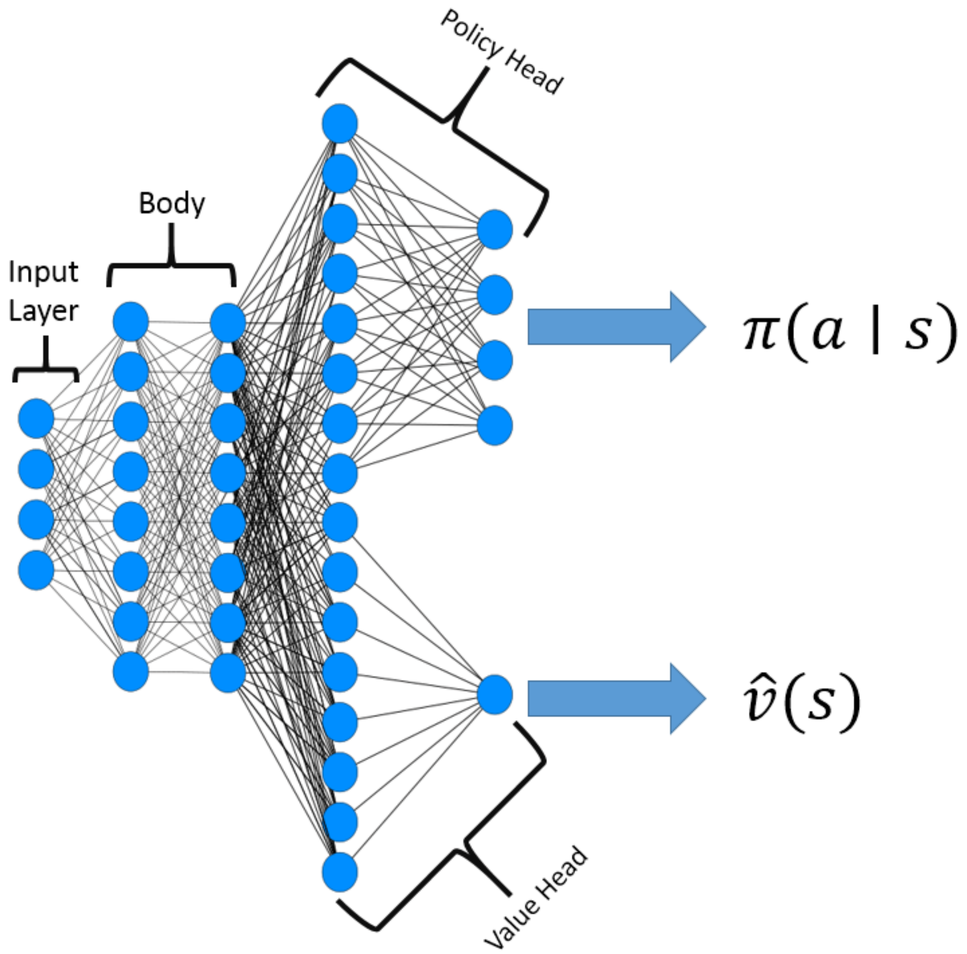


Image: https://en.wikipedia.org/wiki/PID_controller

Software - Reinforcement Learning



- Input: angle & angular acceleration
- Output (1): policy as probability on 21 discrete actions
- Output (2): value estimation of the current state
- Reward Design:
  - 1 when abs(angle)<10; -1 when abs(angle)>20; else 0.
- Sample and retrain with interacting with the environments

Image: https://www.datahubbs.com/two-headed-a2c-network-in-pytorch/

Results

- TL;DR - to get the same performance, PID easy, reinforcement learning hard
- PID is stable, and easy to tune parameters.
- Failed to find a way to make RL work (200 samplings)
- Reasons:
  - Reward Design
  - Network Design

Demo

- PID with Kp=5, Ki=0, Kd=12.8565
- RL like a baby

Lessons

- Hardware
  - Torque determines performance
  - Connect the battery GROUND with the RPi 4 GROUND
- Software
  - Control looping time (sleep if there is time rest)
  - Smooth inputs over time (using Complementary / Kalman Filter)

Thank you!