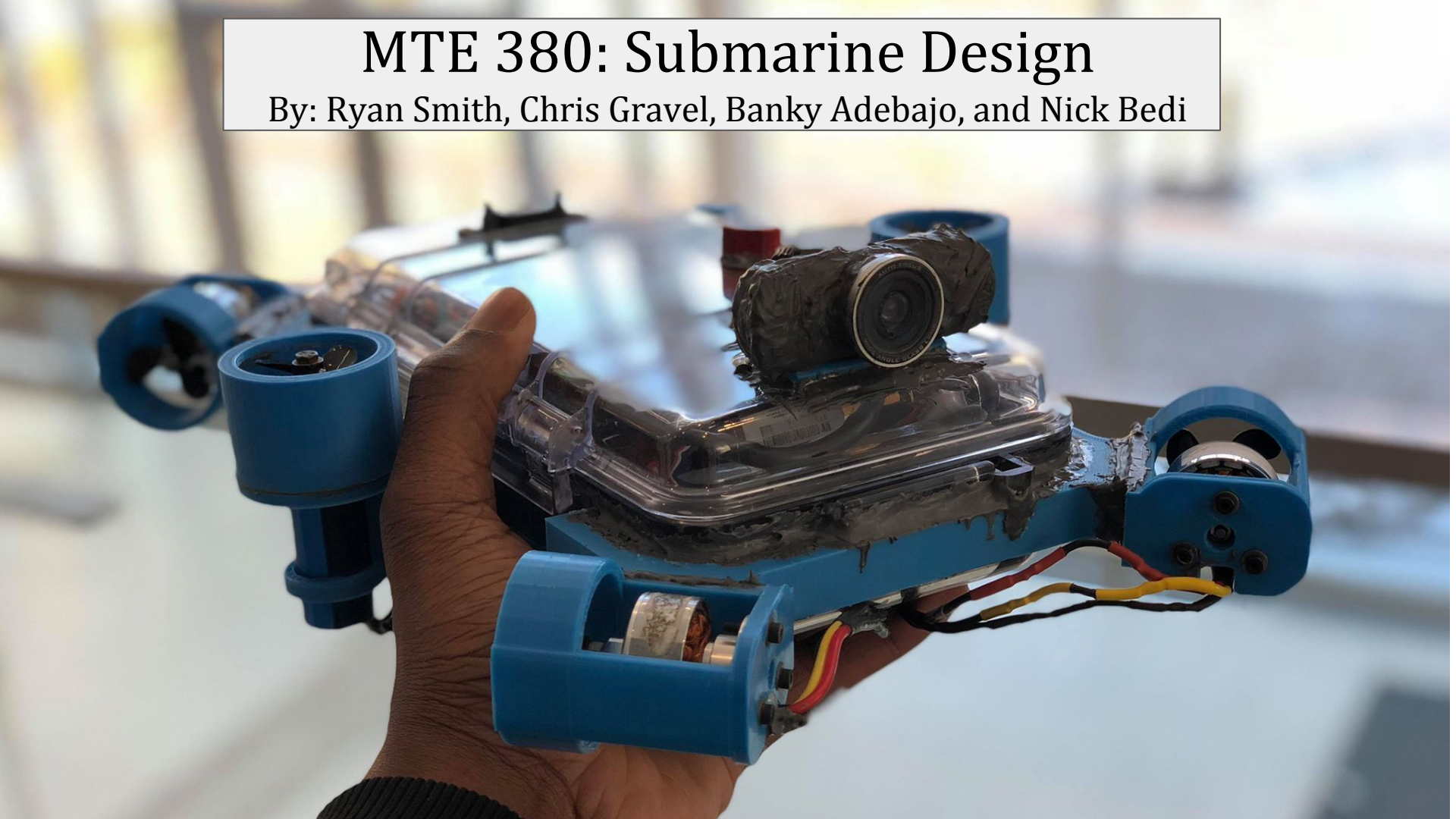
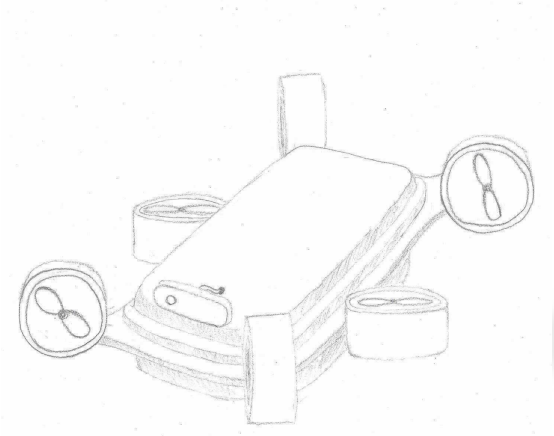
MTE 380: Submarine Design

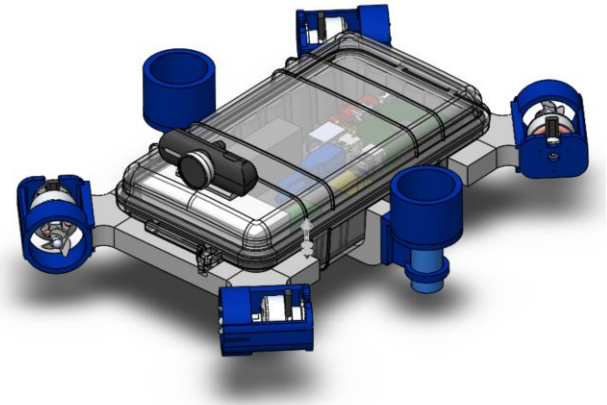
By: Ryan Smith, Chris Gravel, Banky Adebajo, and Nick Bedi



Mechanical Design



Concept Design



CAD Model

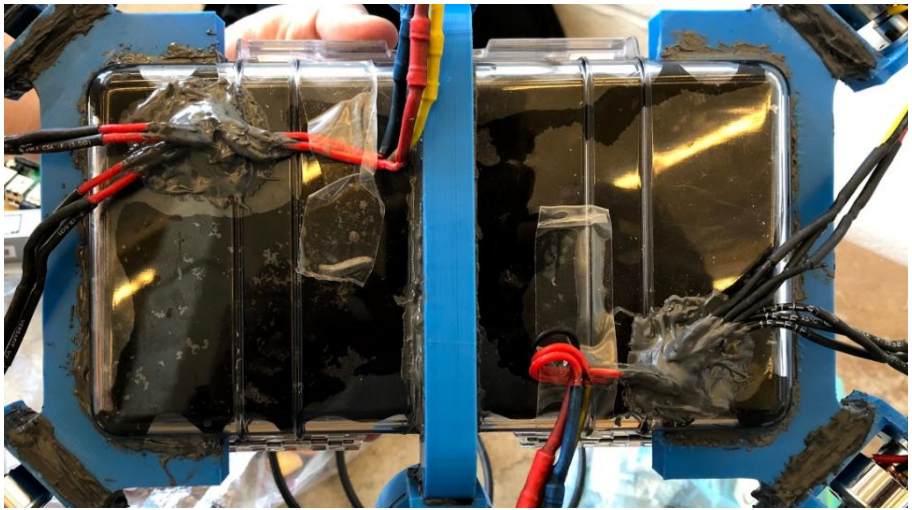
Current Build

Design Functionality

- Ease of control in 3D space
- Ease of electronics access
- Outrunner shrouds

Leak Proofing - JB Weld

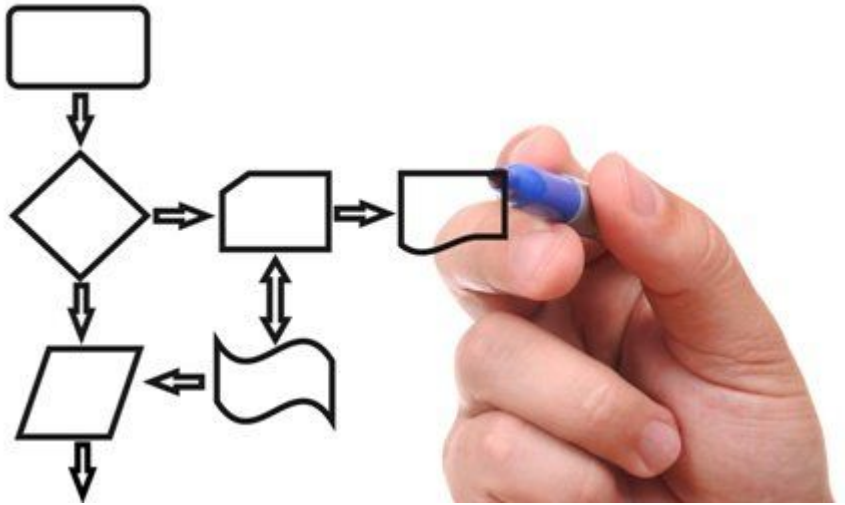
- Rated for marine and plumbing uses
- Lost hold, peeled, fell apart in water
- Caused parts to separate, created leaks



Software Design

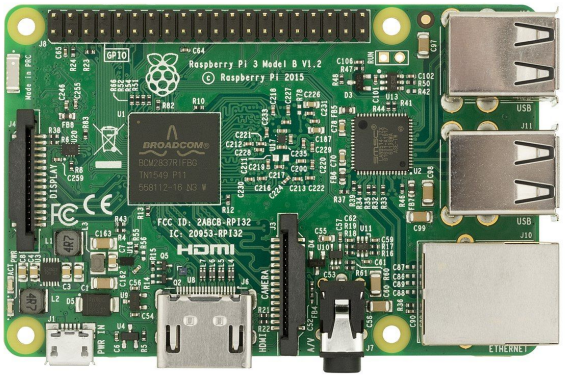
Design Considerations:

- Limited resources
- How to interface



What Are Our Resources?

- Raspberry Pi 3 B
- 4 Cores
- 1Gb RAM
- GPU not supported by machine learning platforms

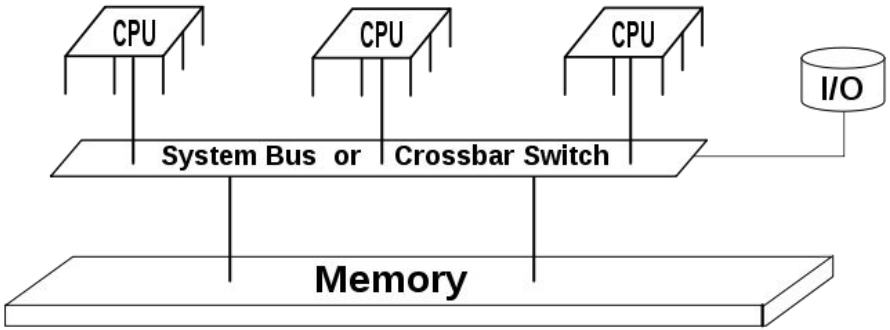


Specifications

SoC: Broadcom BCM2837
CPU: 4× ARM Cortex-A53, 1.2GHz
GPU: Broadcom VideoCore IV
RAM: 1GB LPDDR2 (900 MHz)
Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage: microSD
GPIO: 40-pin header, populated
Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

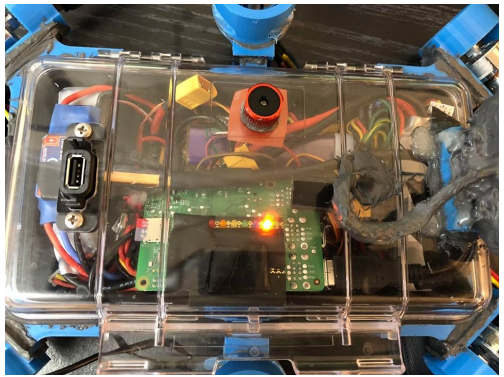
How Do We Utilize Multiple Cores?

- Roadblocks:
  - Python doesn’t allow multiple cores to be used in the same process
  - If using multiple processes, need to design it for inter-process communication
- Solution:
  - Make lots of processes and let Linux handle the priorities



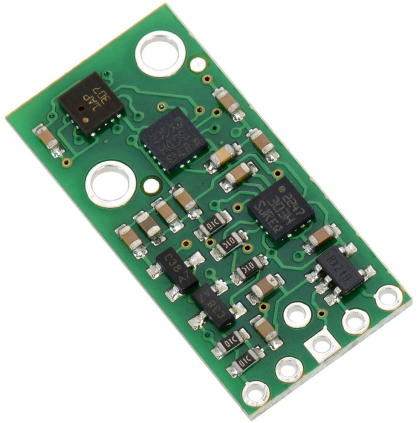
How Do We Interface With The ROV?

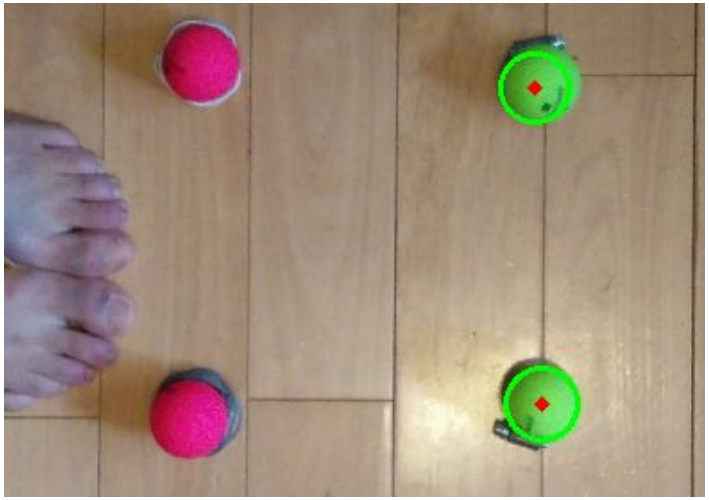
- Do everything through one interface
- Don’t require opening the case
- Need feedback to know the state of the robot
- Cheatcodes!!!



Localization

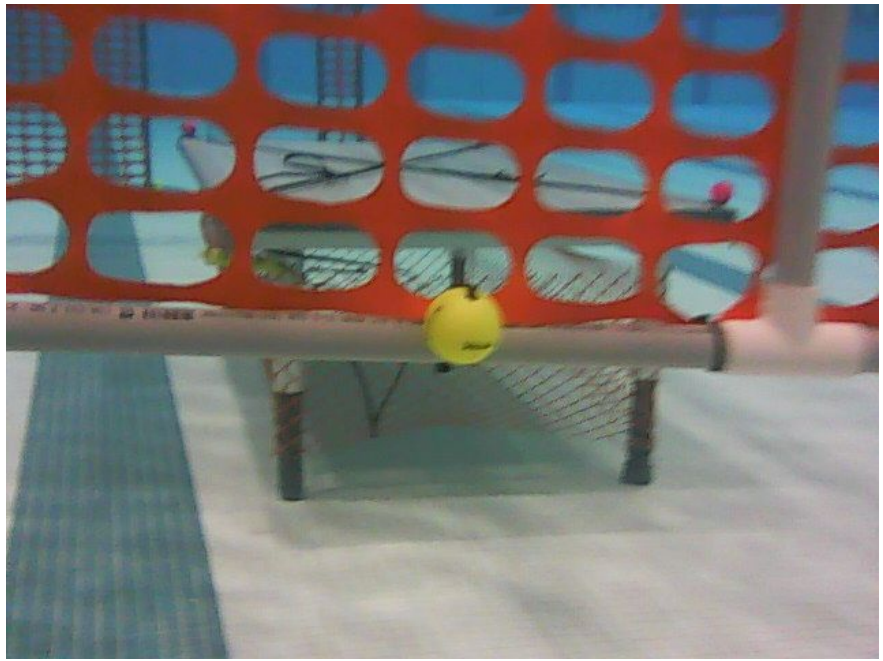
- Depth Sensor
  - 2mm resolution
  - Waterproof
  - Temperature Compensated
- IMU
  - 3-axis gyro
  - 3-axis accelerometer
  - 3-axis magnetometer
- Dead Reckoning
- Computer Vision
  - Colour Filtering + Blob Detection





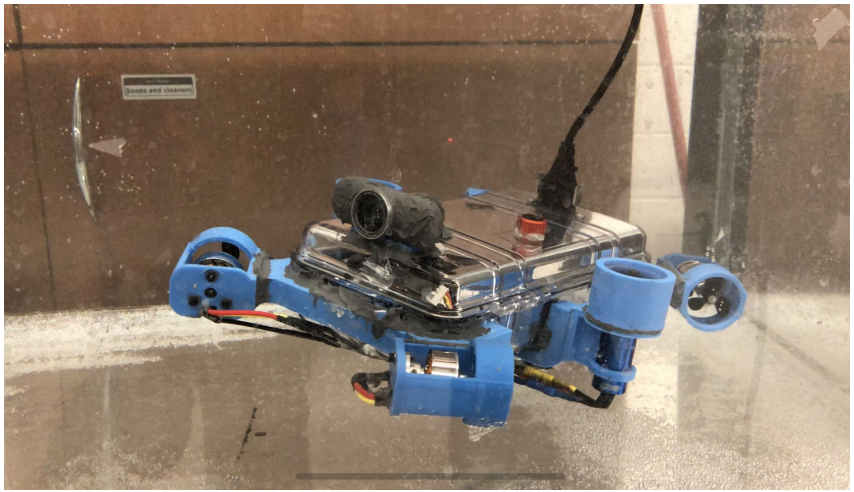
State Transitions

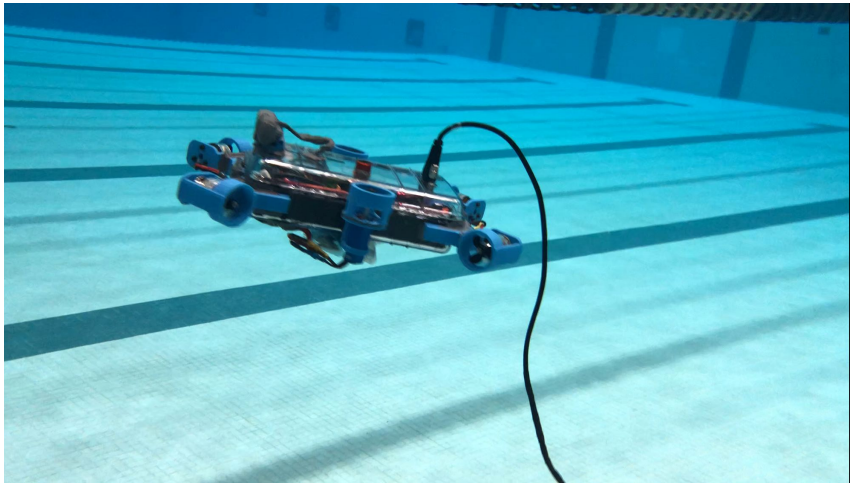
- Objectives Broken into smaller tasks
- Move forward until certain objective
- Eg. Travel at set depth until under mesh



Automatic Controls

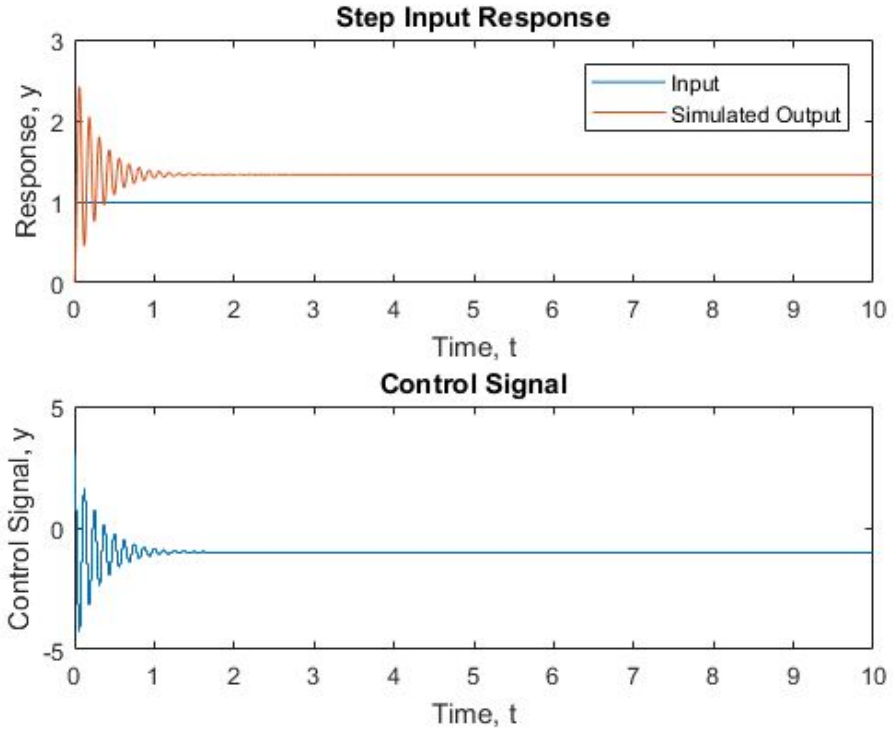
- Depth control based on height is required to navigate course
- The input to the control system is a pressure/temperature sensor
- The output is the thrust produced by the side mounted motors





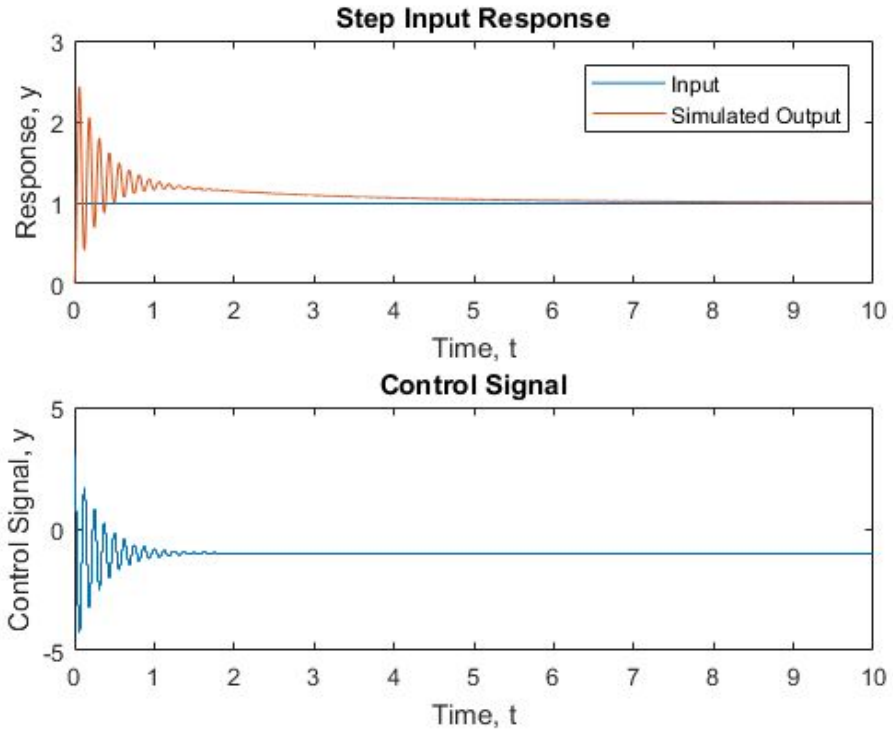
P-Control Only

- System responds quickly, but significant oscillatory behavior
- Disturbance input causes a steady state error



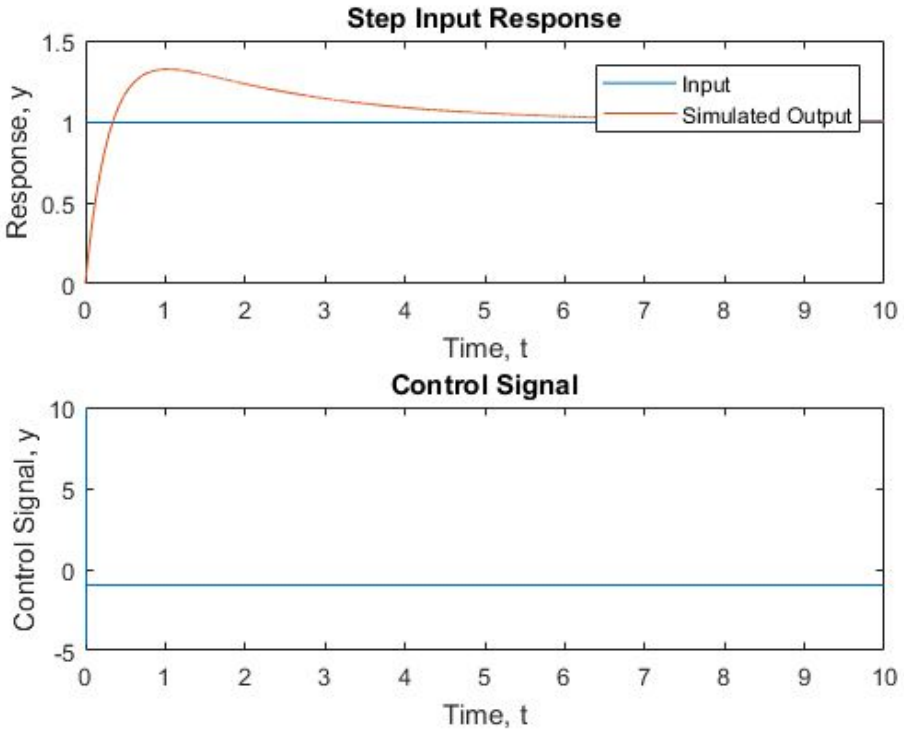
PI-Control Only

- Integral term removes steady state error over time
- There is still a lot of oscillatory behavior



PID-Control

- Smooth response without any oscillations
- No steady state error



PID controller

- The plant parameters are not fully characterized
- Thus PID controller is tuned manually using the Ziegler–Nichols method
- K_p is increased until oscillation, then K_I and K_d are set accordingly

Gain values for controller			
Control Type	K_p	K_i	K_d
P	$0.5K_u$	-	-
PI	$0.45K_u$	$0.54K_u/T_u$	-
PID	$0.6K_u$	$1.2K_u/T_u$	$3K_uT_u/40$