Robotics 311 : How to build robots and make them move

Prof. Elliott Rouse

GSI Yves Nazon MS

Fall 2022

Photo credit: Brenda Ahearn / U-M CoE

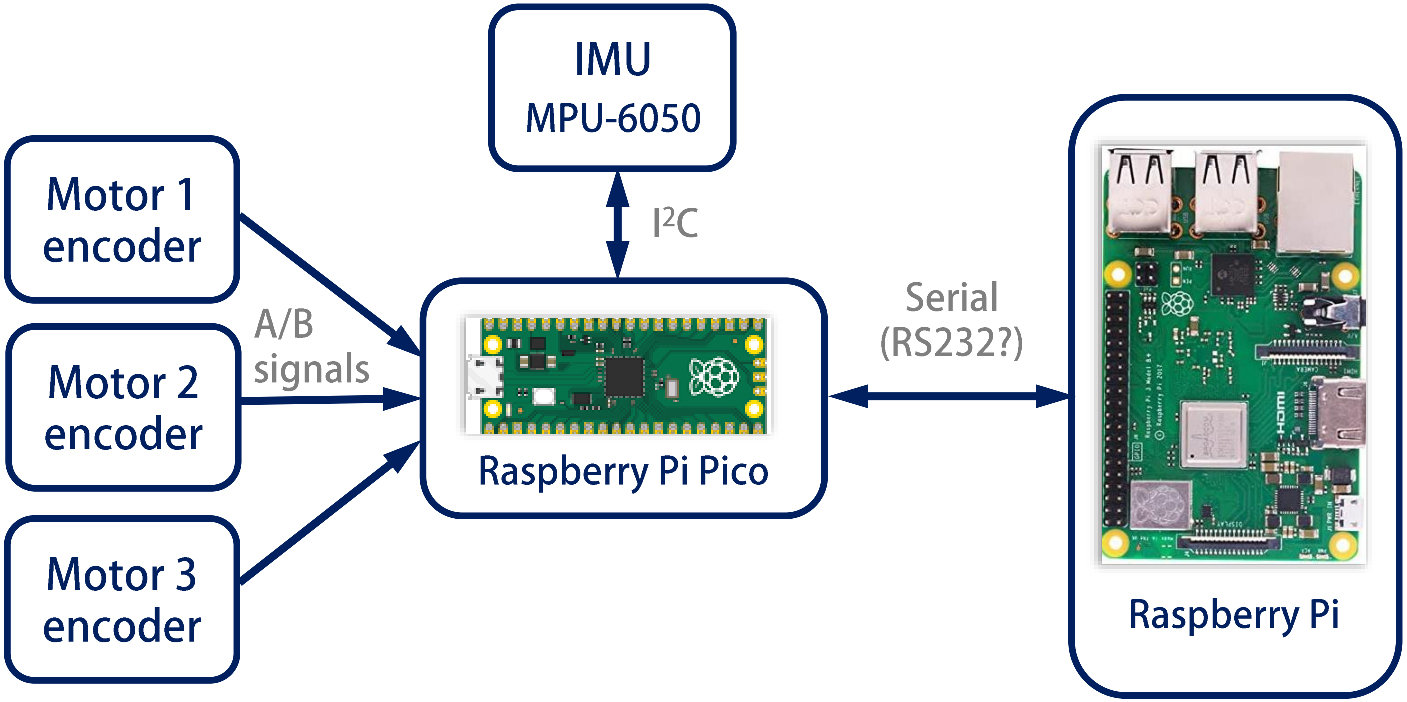
ROB 311 – Lab 10

- Today we will begin control
- Overview of control architecture
- Introduction of balance controller
- Explanation of code

- Announcements
  - HW 4 posted, due 11/10 at class start
  - Midterm exam – 11/8
  - HW4 Q5 – don't worry about getting z-axis velocity with the IMU

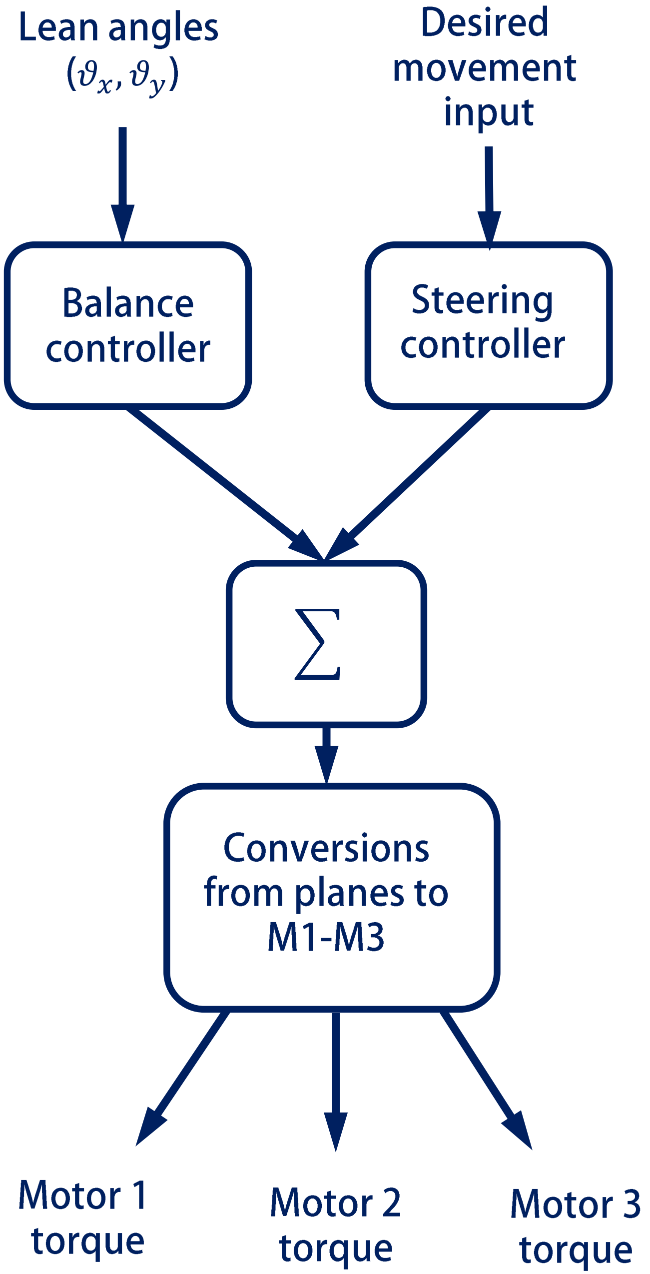
System Layout Review

- As a reminder, our system is setup as shown below
- We receive encoder / IMU information each loop cycle, and we need to convert this into intelligent instructions to balance and move



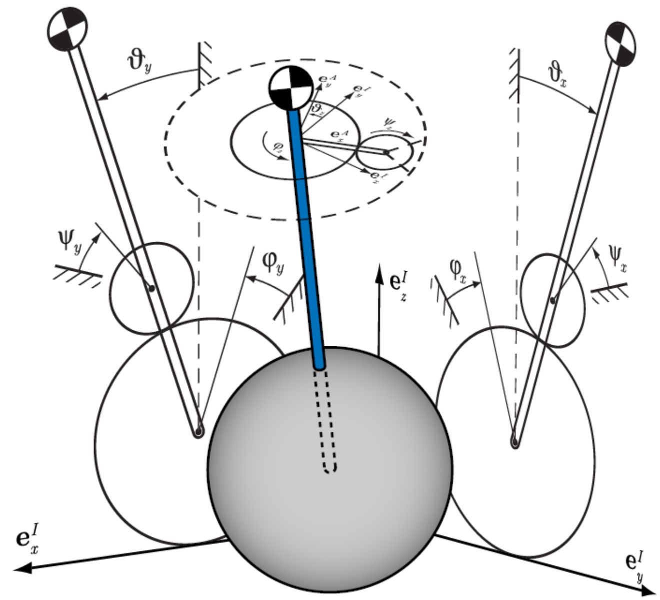
Balance and Steering Controllers

- We break the controller into the two planes
- Each plane will be handled independently
- Each plane has two controllers that run in parallel
  - Balance controller / steering controller
  - They will be separate but will run simultaneously
- There will be four total controllers in parallel
- We will superimpose the torques from the balance and steering controllers
- Simultaneous balance and steering
- We will begin with the **balance controller**
- Let's think about how this controller should be designed



Balance Controller

- To develop the balance controller, we want the ball-bot to right itself
- What state variables can we use to determine how well the system is balancing? ϑ_x and ϑ_y
- We can measure ϑ_x and ϑ_y directly using the IMU
- We need to apply a counter torque to bring the system upright
- If we know ϑ_{axis} , we can use this to determine a counter torque to balance the ball-bot
- What should the controller effort be at vertical? Zero—so the controller provides a lot of effort when leaned and no effort when vertical
- We can use ϑ_x multiplied by a controller gain to define the controller effort—it satisfies these requirements



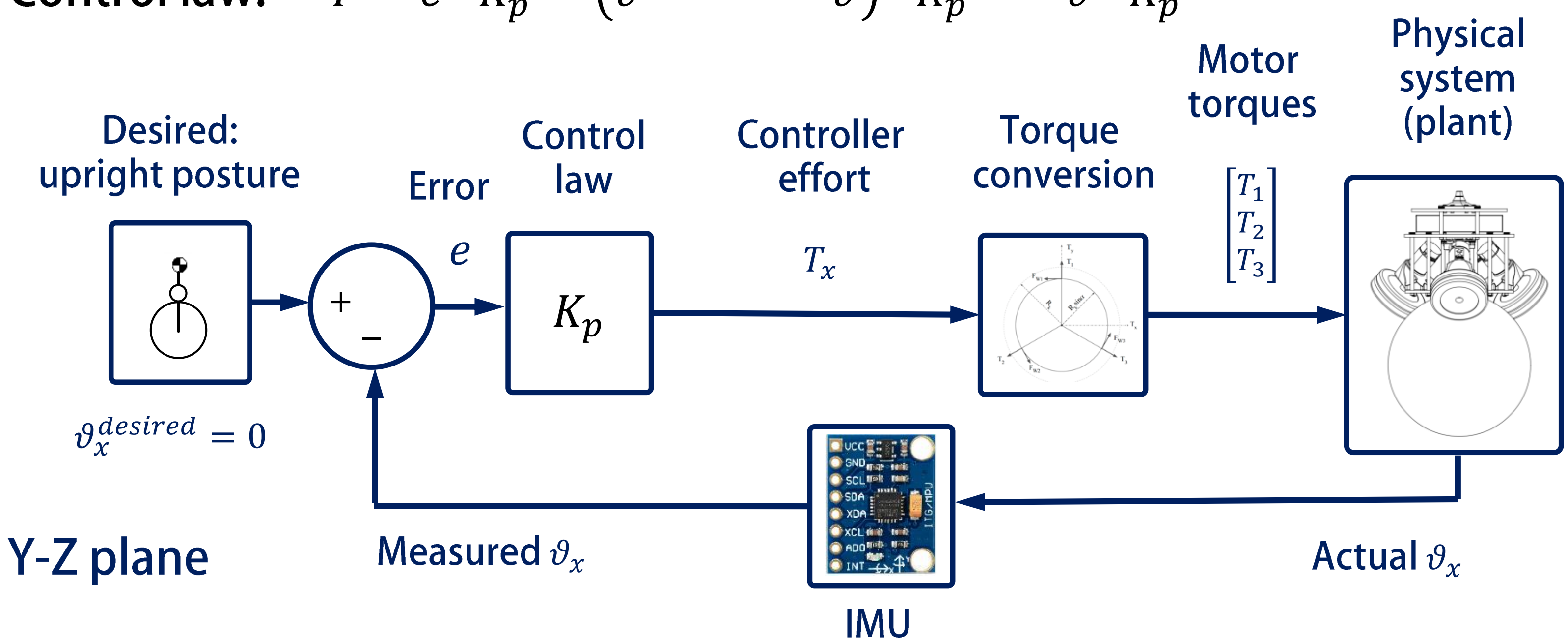
Balance Controller

- Let's build a basic feedback controller using chassis lean angle
- We know we want controller effort to be

Controller gain or 'proportional gain'

$$T = -K_p \cdot \vartheta_{axis}$$

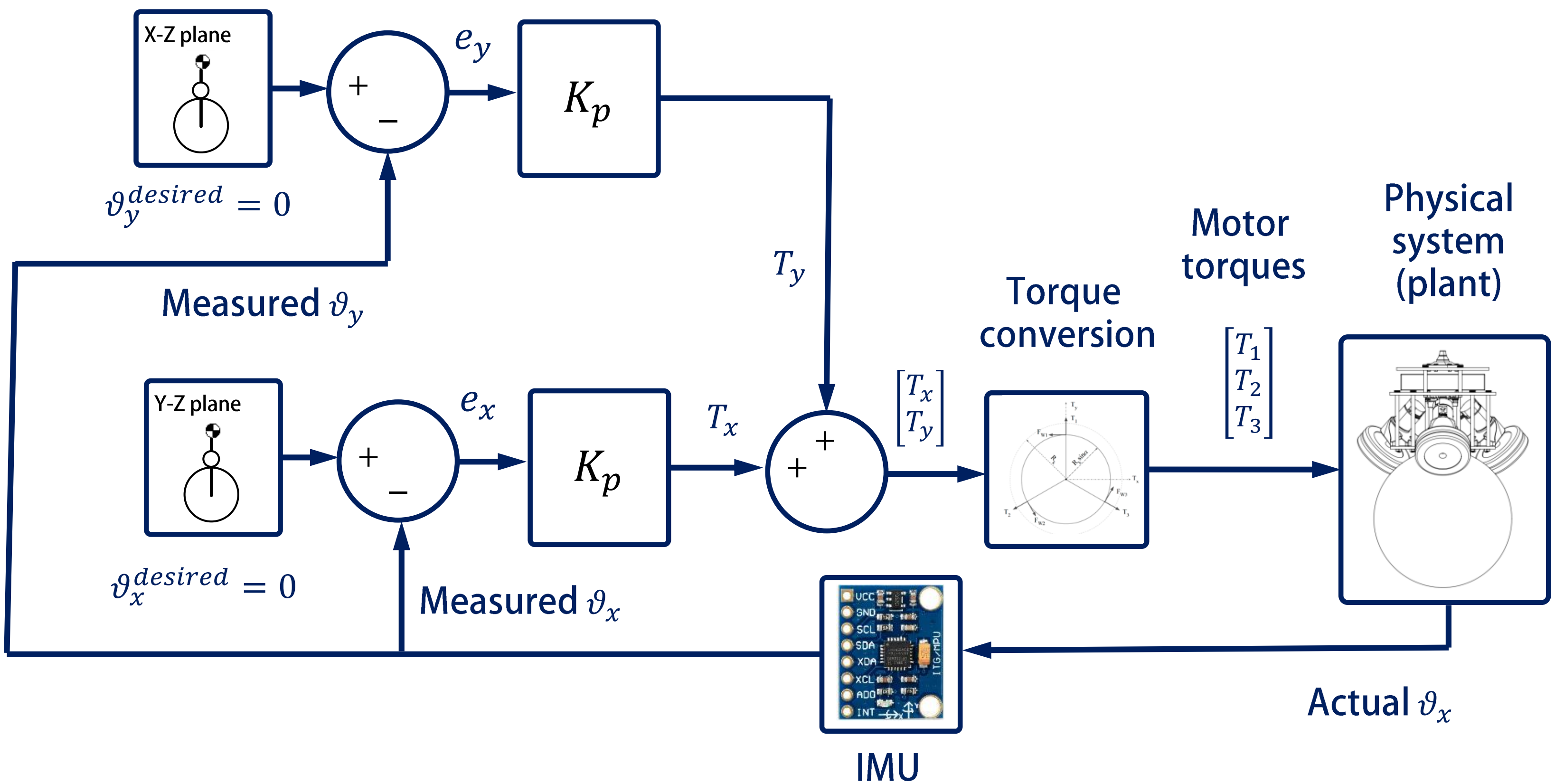
- Lets define a reference trajectory of upright posture ($\vartheta_x^{desired} = \vartheta_y^{desired} = 0$)
- By defining a reference trajectory, we can make our approach more general
- Control law: $T = e \cdot K_p = \left(\vartheta^{desired} - \vartheta\right) \cdot K_p = -\vartheta \cdot K_p$



Signals get multiplied by blocks

Balance Controller

- How about for both planes?
- Control law: $T = e \cdot K_p = \left(\vartheta^{desired} - \vartheta\right) \cdot K_p = -\vartheta \cdot K_p$
- How do we choose K_p? This process is called 'tuning' a controller



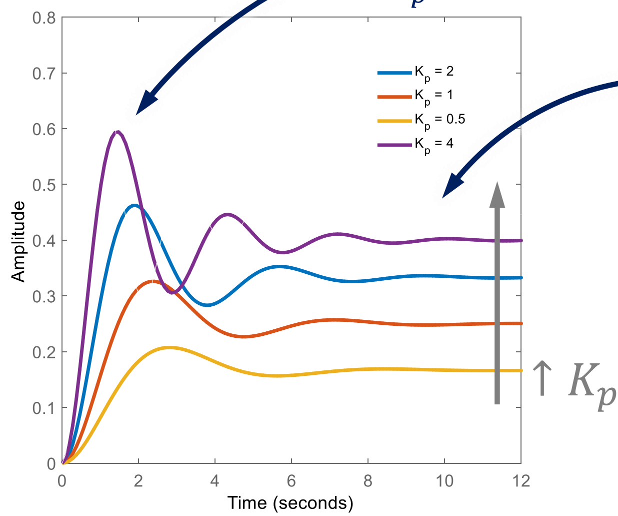
Tuning Your Controller

- A proportional controller is like a virtual spring (when controlling around position)
- The greater the K_p (stiffness), the greater the restoring torque to balance
- But it has to be selected carefully
- A controller gain that's too high will cause oscillations and instability
- Let's look at an example 2nd order system under proportional control trying to go from 0 to 1 at $t = 0$

More oscillations as K_p is increased

The responses get closer to the goal as K_p is increased

These systems are trying to reach the goal of 1 at $t = 0$



These responses are known as 'step responses'

Tuning Your Controller

- You will use code on Canvas/Lab 10 – `ROB311_stability_controller_kp_walkthrough.py`
- Code walk through
- You will need to iteratively adjust your controller gains

Adjust these gains

```
# ---------------------------------------------------------------------------
###############  THESE WILL NEED TO BE CAREFULLY ADJUSTED ##################
# ---------------------------------------------------------------------------

# Proportional gains for the stability controllers (X-Z and Y-Z plane)

KP_THETA_X = 0.0                                  # Adjust until the system balances
KP_THETA_Y = 0.0                                  # Adjust until the system balances

# ---------------------------------------------------------------------------
#############################################################################
```

- Begin with a small number and increase until it starts to balance
- This will take time and you may need to make many adjustments
- Be careful and be ready to ctrl+c to exit if it begins doing something bad
- Tuning a controller can be very dangerous when working with powerful machines—for the ball-bot, we don't need to worry too much