Robotics 311 : How to build robots and make them move

Prof. Elliott Rouse

GSI Yves Nazon MS

Fall 2022

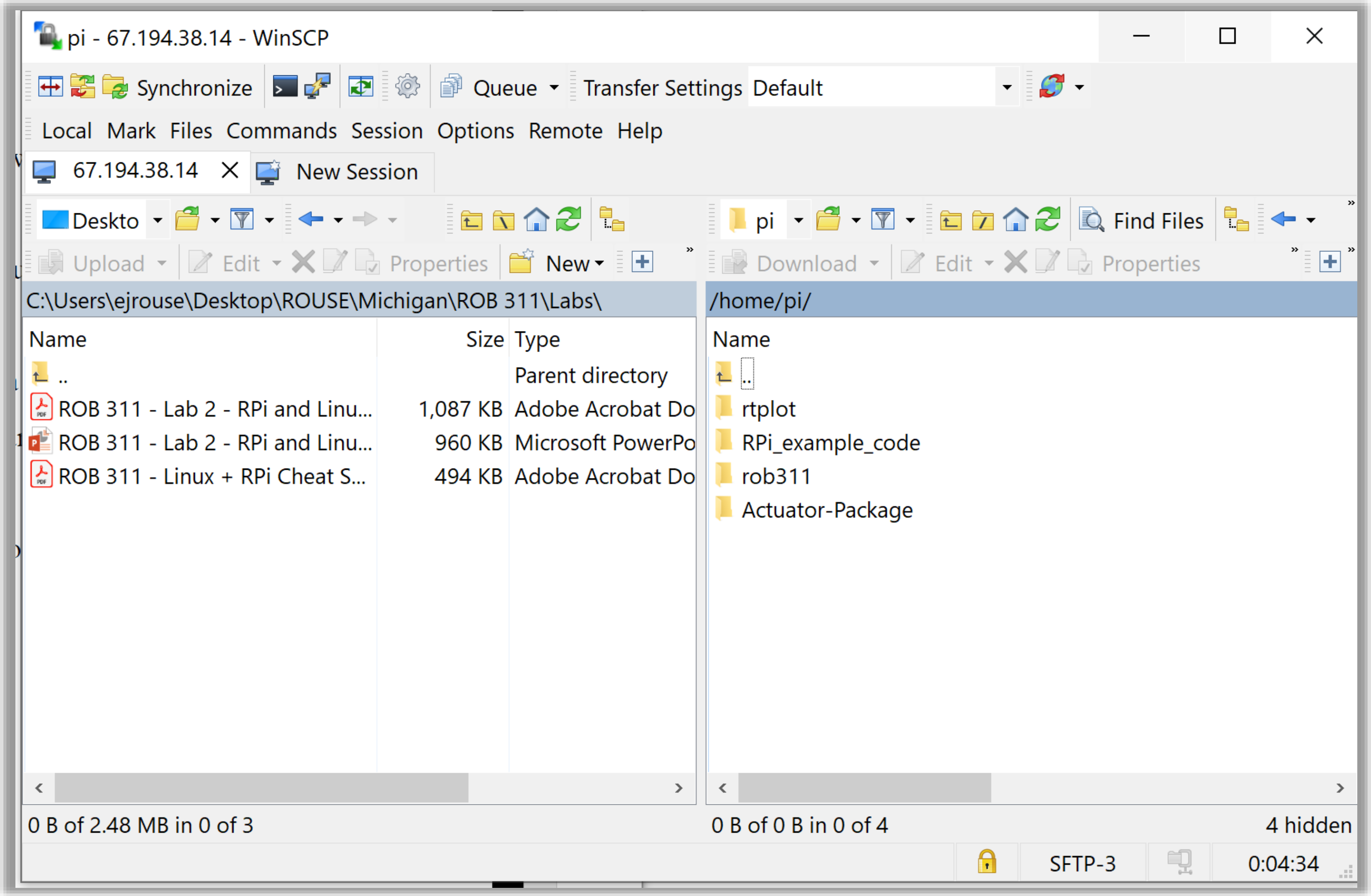
ROB 311 – Lecture 4

- Today:
  - Review Lab 2
  - Show how to adjust settings in Raspi-config
  - Describe layout / file structure
  - Review common Linux commands
  - Review planar ball-bot mechanics
  - Show motor selection MATLAB code
  - In-class example exercise

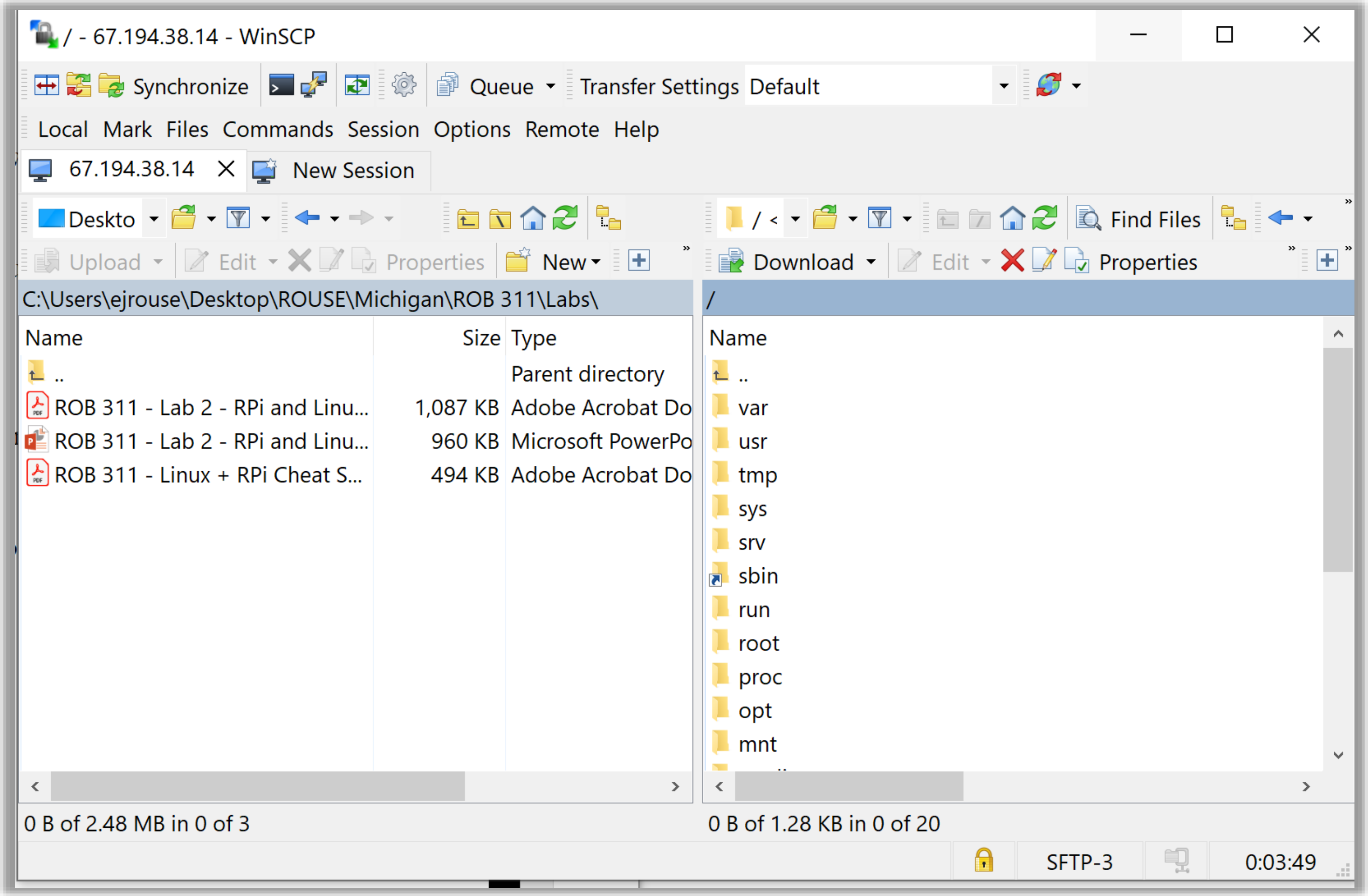
ROB 311 – Lab 2

- What did we do with Git Actions yesterday?
- Next, we downloaded programs—what did they do?
- How did we connect to the Raspberry Pi?
- What is ‘nano’ ?
- What does `sudo` do?
- Those that got the access point working, what did you do?

ROB 311 – Lab 2


pi - 67.194.38.14 - WinSCP
Synchronize
Queue
Transfer Settings Default
Local Mark Files Commands Session Options Remote Help
67.194.38.14
New Session
Deskto
Upload
Edit
Properties
New
pi
Find Files
Download
Edit
Properties
C:\Users\ejrouse\Desktop\ROUSE\Michigan\ROB 311\Labs\
Name
Size
Type
..
Parent directory
ROB 311 - Lab 2 - RPi and Linu...
1,087 KB
Adobe Acrobat Do
ROB 311 - Lab 2 - RPi and Linu...
960 KB
Microsoft PowerPo
ROB 311 - Linux + RPi Cheat S...
494 KB
Adobe Acrobat Do
/home/pi/
Name
..
rtplot
RPi_example_code
rob311
Actuator-Package
0 B of 2.48 MB in 0 of 3
0 B of 0 B in 0 of 4
4 hidden
SFTP-3
0:04:34

ROB 311 – Lab 2


/ - 67.194.38.14 - WinSCP
Synchronize
Queue
Transfer Settings Default
Local Mark Files Commands Session Options Remote Help
67.194.38.14
New Session
Deskto
Find Files
Upload
Edit
Properties
New
Download
Edit
Properties
C:\Users\ejrouse\Desktop\ROUSE\Michigan\ROB 311\Labs\
/
Name
Size
Type
..
Parent directory
ROB 311 - Lab 2 - RPi and Linu... 1,087 KB Adobe Acrobat Do
ROB 311 - Lab 2 - RPi and Linu... 960 KB Microsoft PowerPo
ROB 311 - Linux + RPi Cheat S... 494 KB Adobe Acrobat Do
Name
..
var
usr
tmp
sys
srv
sbin
run
root
proc
opt
mnt
0 B of 2.48 MB in 0 of 3
0 B of 1.28 KB in 0 of 20
SFTP-3
0:03:49

ROB 311 – Lab 2

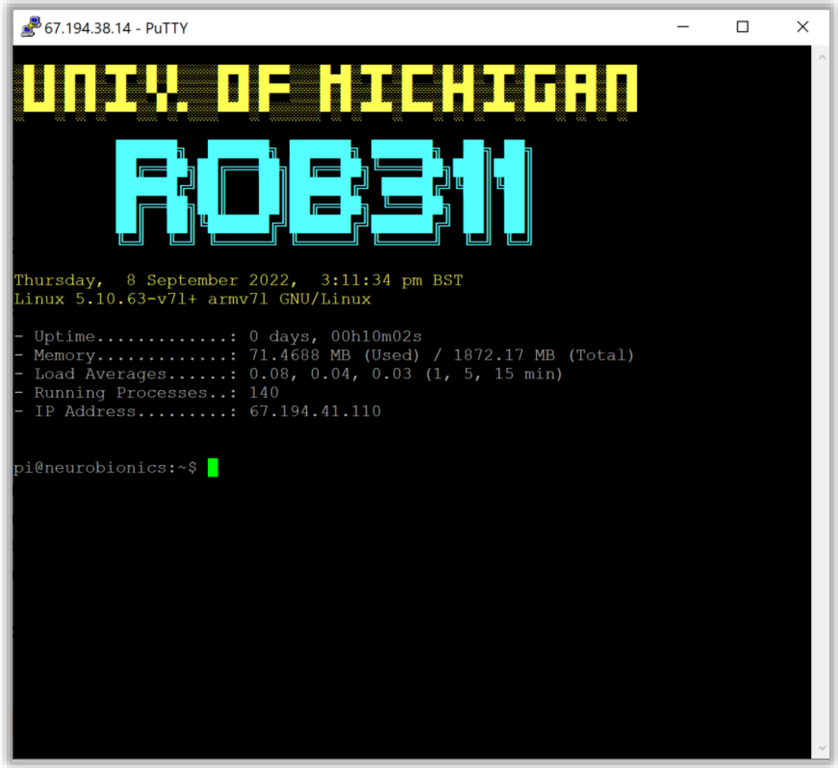
- Important file locations
  - Startup mailer: `/etc/startup_mailer.py`
  - Wireless network configuration file:
    `/etc/wpa_supplicant/wpa_supplicant-wlan0.conf`
  - ROB 311 folder: `/home/pi/rob311`
  - Example Python files: `/home/pi/RPi_example_code`
  - Real time plotter (for later): `/home/pi/rtplot`
  - Ignore: `/home/pi/Actuator-Package`

Cheat sheet uploaded to Canvas

- Common Linux commands
  - `ls` – list files in a directory (`ls -a` shows hidden files as well)
  - `cd DIRECTORY` – change directory to `DIRECTORY`
  - `mkdir DIRECTORY` – creates a folder named `DIRECTORY`
  - `touch FILENAME` – creates a file named `FILENAME`
  - `cat FILENAME` – displays contents of `FILENAME` in terminal
  - `nano FILENAME` – opens `FILENAME` in text editor named Nano
  - `ping www.google.com` – pings google to verify internet connectivity
  - `Sudo shutdown now or sudo reboot` – shuts down or reboots

ROB 311 – Lab 2

- SSH demonstration
- File structure overview
- Updating configuration in `raspi-config`
- We need to enable a setting (SPI) on your Raspberry Pis
- It would be a good time to learn the layout of your RPi, watch the introduction video if needed, and try some of the commands



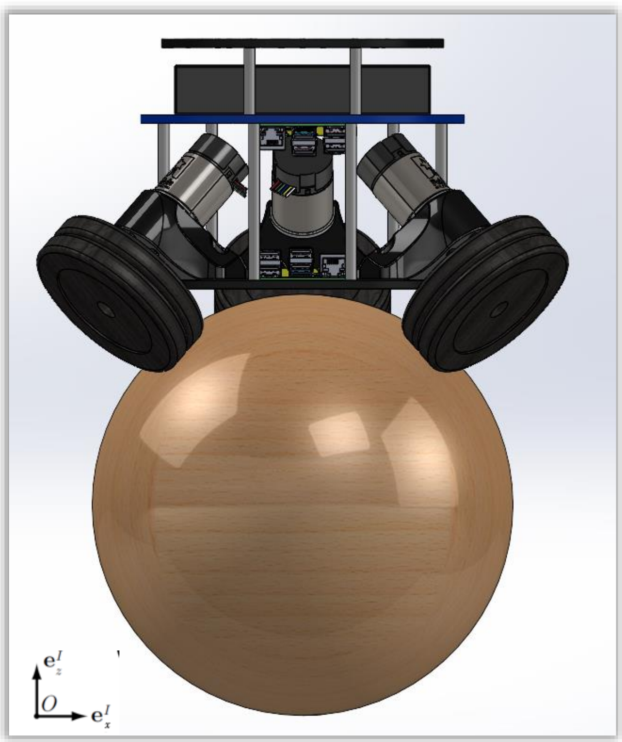
File structure / SSH demo

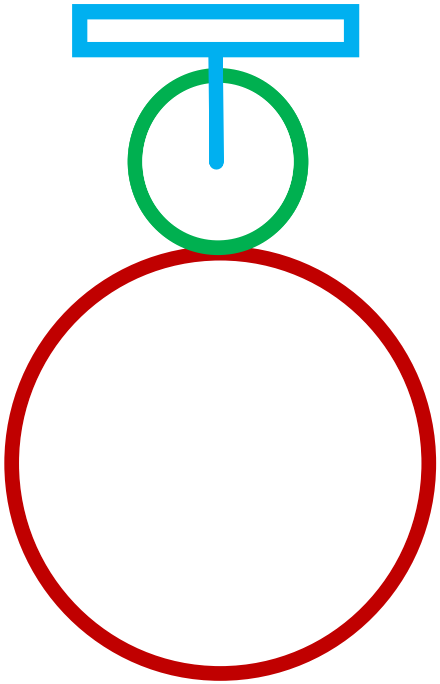
Mechanical Modeling of a Planar Ball-Bot

- Modeling the mechanics of our ball-bot to ensure the proper motor / transmission are selected
- We discussed forward and inverse models
  - Forward models solve the differential governing equations and requires an ODE solver to be used (e.g. Simulink)
    - These are causal, using only information from the present / past
    - They use integration and not derivation
  - Inverse models numerically differentiate to obtain a solution
    - Because there is differentiation, which requires knowledge of the slope, it uses information from the future
    - This is known as a non-causal analysis and is often a good approximation
- Our system was also two DOF (with constraints) and we needed to specify two motion profiles

Mechanical Modeling of a Planar Ball-Bot

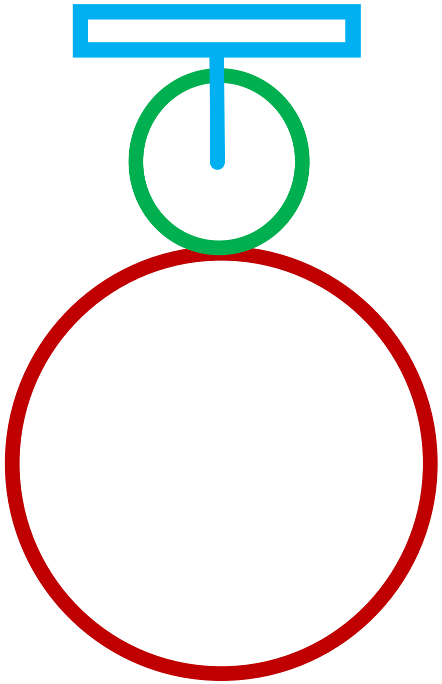
- We are learning a planar model of a ball-bot and discussed the simplification
- The required torques will be more than what would be needed in a design with three motors – use as conservative estimate





Mechanical Modeling of a Planar Ball-Bot

- We discussed their physical properties
- These can be found using many tools: estimating, direct measurements, online resources, datasheets, intuition
- We measured / looked up:



Body / chassis:

- Distance to CoM: $L = 0.23\ m$
- Mass: $m_A = 1.71\ kg$
- Moment of inertia in x and y planes: $I_{x,y} = 0.01\ kgm^2$
- Moment of inertia in z plane: $I_z = 0.017\ kgm^2$

Wheel:

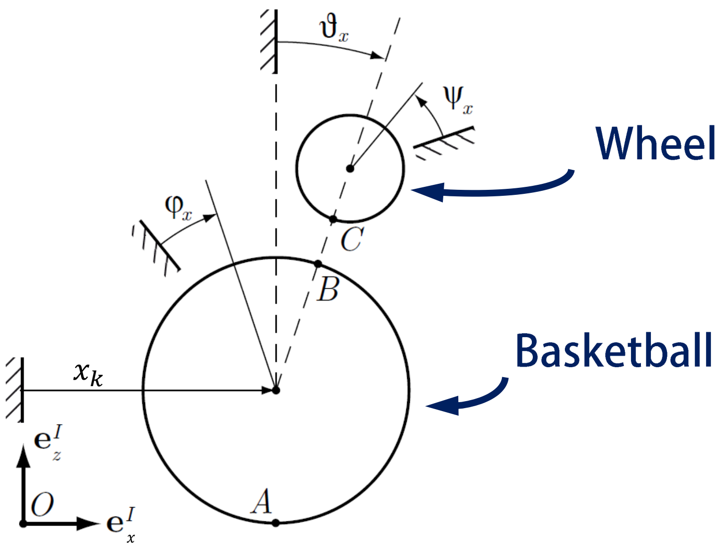
- Mass: $m_w = 0.29\ kg$
- Radius: $r_w = 0.1\ m$
- Moment of inertia: $I_w = 0.001\ kgm^2$

Ball:

- Mass: $m_k = 0.62\ kg$
- Radius: $r_k = 0.12\ m$
- Moment of inertia: $I_k = 0.004\ kgm^2$

Mechanical Modeling of a Planar Ball-Bot

- Now we want to think about the ball's motion
- We need to define parameters that describe this motion
- Both x and y planes are identical mechanically



Body / chassis motion:

- Lean rotation around the ball — ϑ_x or ϑ_y (rad)

Wheel motion:

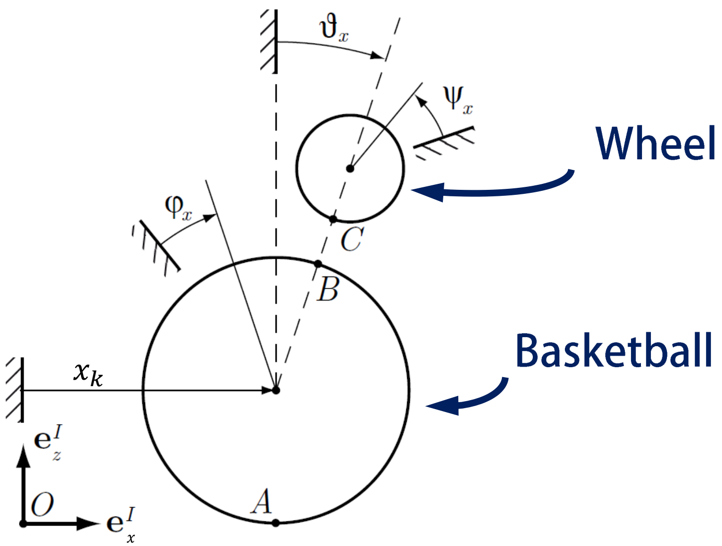
- Rotation — ψ_x or ψ_y (rad)
- Lean rotation around the ball — ϑ_x or ϑ_y (rad)

Ball motion:

- Translation in x or y planes: — x_k or y_k (m)
- Rotation — φ_x or φ_y (rad)

Mechanical Modeling of a Planar Ball-Bot

- Now we want to think about the ball's motion
- We need to define parameters that describe this motion
- Both x and y planes are identical mechanically



- Degrees of freedom: Four
  - x_k – Forward motion of ball
  - Theta - ϑ – Lean of body / chassis
  - Phi - φ – Rotation of ball
  - Psi - ψ – Rotation of wheel
- Constraints: Wheels do not slip

$$x_k = r_k \varphi$$

Basketball does not slip on the floor at A

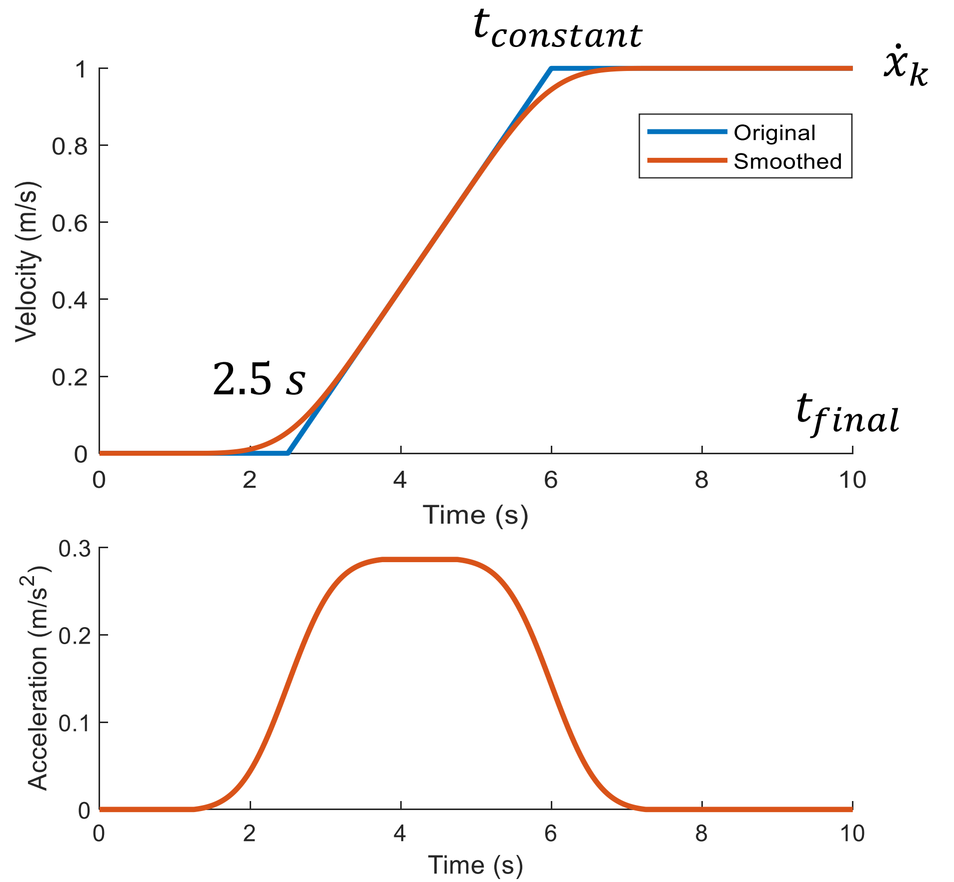
$$\psi = \frac{r_k}{r_w}(\varphi - \vartheta) - \vartheta$$

Tangential velocity constraint at B-C interface

- Two remaining DOFs:
  - ϑ and x_k for inverse modeling
  - ϑ and ψ for forward modeling

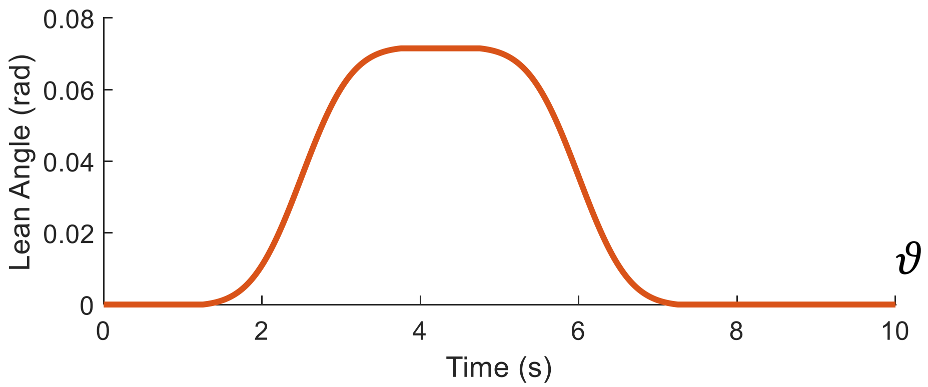
Mechanical Modeling of a Planar Ball-Bot

- We need to prescribe motion to the two remaining degrees of freedom
- Lets begin with x_k
- If we want to predict the torque required, we need to generate a motion profile.
- We choose this when spec'ing the task
  - How long do we want it to take?
  - Final velocity: $\dot{x}_k^{final} = 1\,\frac{m}{s}$
  - Ramp duration: $3.5\ s$
  - Resulting acceleration
  - What does this say about current and voltage needed?
    - Acceleration ~ Torque ~ Current
    - Velocity ~ Voltage



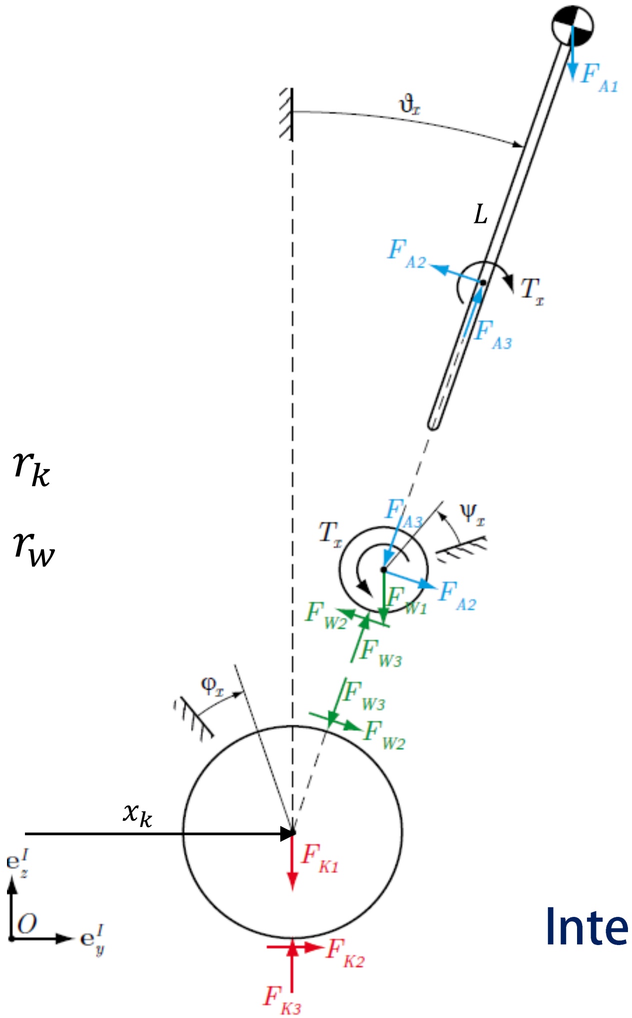
Mechanical Modeling of a Planar Ball-Bot

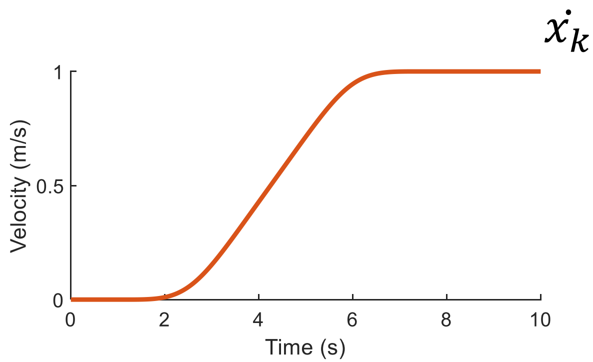
- We need to prescribe motion to the two remaining degrees of freedom
- We also need to provide information for the other DOF - ϑ – chassis lean angle
- Similarly, we choose this—how far do we want to be able to lean?
- We know it should lean with the applied torque
- Approach: scale the linear acceleration to an acceptable trajectory
- Lean angle is physically constrained – wheels will lose contact
- Lets use approx. 4 deg (0.07 rad) as an acceptable value



Mechanical Modeling of a Planar Ball-Bot

$$F_{K1} = g \cdot m_k$$
$$F_{W1} = g \cdot m_w$$
$$F_{A1} = g \cdot m_a$$





Ball radius r_k

Wheel radius r_w

Constraints:
$$x_k = r_k \varphi$$
$$\psi = \frac{r_k}{r_w}(\varphi - \vartheta) - \vartheta$$

Ball velocity: $\dot{\varphi} = \frac{\dot{x}_k}{r_k}$

Intermediate var: $\gamma = L \cdot m_a + (r_k + r_w) \cdot m_w$

Tangential contact force : $F_{W2} = (m_a + m_w) \cdot (g \cdot \sin(\vartheta) - r_k \ddot{\varphi} \cos(\vartheta)) - \gamma \ddot{\vartheta}$

Mechanical Modeling of a Planar Ball-Bot

$$\gamma = L \cdot m_a + (r_k + r_w) \cdot m_w$$

$$\underbrace{F_{W2} = (m_a + m_w) \cdot (g \cdot \sin(\vartheta) - r_k \ddot{\varphi} \cos(\vartheta)) - \gamma \ddot{\vartheta}}_{\text{From free body diagram}}$$

$$\left.\begin{aligned} \dot{\varphi} &= \frac{\dot{x}_k}{r_k} \\ \psi &= \frac{r_k}{r_w}(\varphi - \vartheta) - \vartheta \end{aligned}\right\} \text{Constraints}$$

Required current

Output torque to roll ball / mass

Torque to accelerate motor inertia

$$i_w = \frac{\frac{T_x}{N\eta} + JN\ddot{\psi} + bN\dot{\psi}}{k_t}$$

Torque to overcome viscous loss in motor

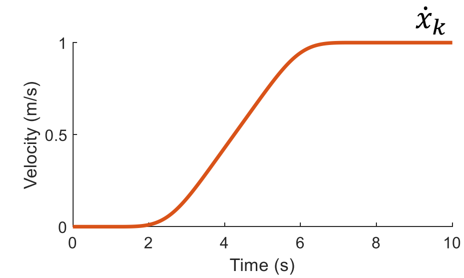
Required voltage

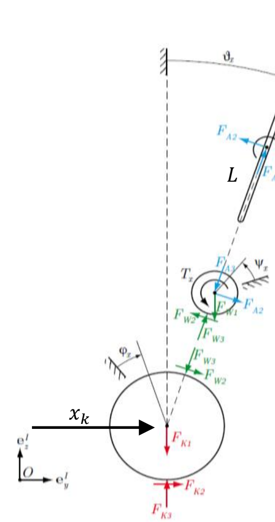
Voltage drop across resistance

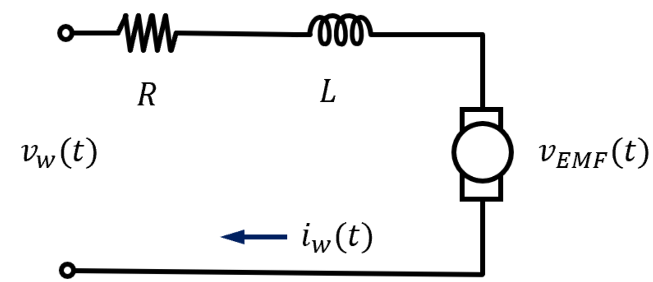
Back EMF voltage

$$v_w = i_w R + k_t N \dot{\psi} + L\frac{di_w}{dt}$$

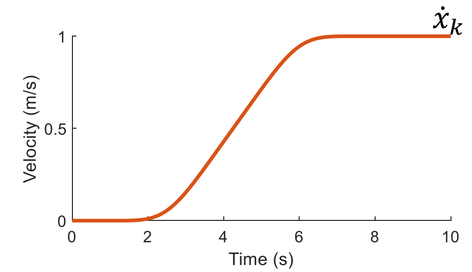
Voltage drop across inductance







Mechanical Modeling of a Planar Ball-Bot

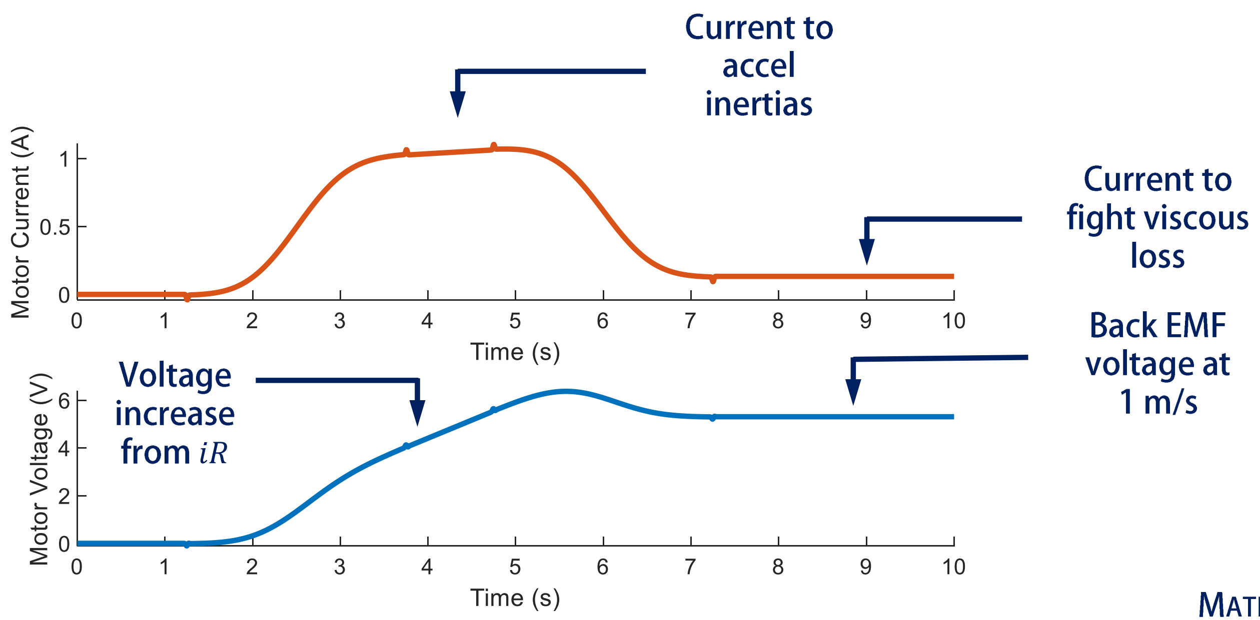


$$\gamma = L \cdot m_a + (r_k + r_w) \cdot m_w$$

$$F_{W2} = (m_a + m_w) \cdot (g \cdot \sin(\vartheta) - r_k \ddot{\varphi} \cos(\vartheta)) - \gamma \ddot{\vartheta}$$

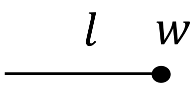
$$\dot{\varphi} = \frac{\dot{x}_k}{r_k} \qquad \psi = \frac{r_k}{r_w}(\varphi - \vartheta) - \vartheta$$

$$i_w = \frac{\frac{T_x}{N\eta} + JN\ddot{\psi} + bN\dot{\psi}}{k_t} \qquad v_w = i_w R + k_t N \dot{\psi} + L\frac{di_w}{dt}$$



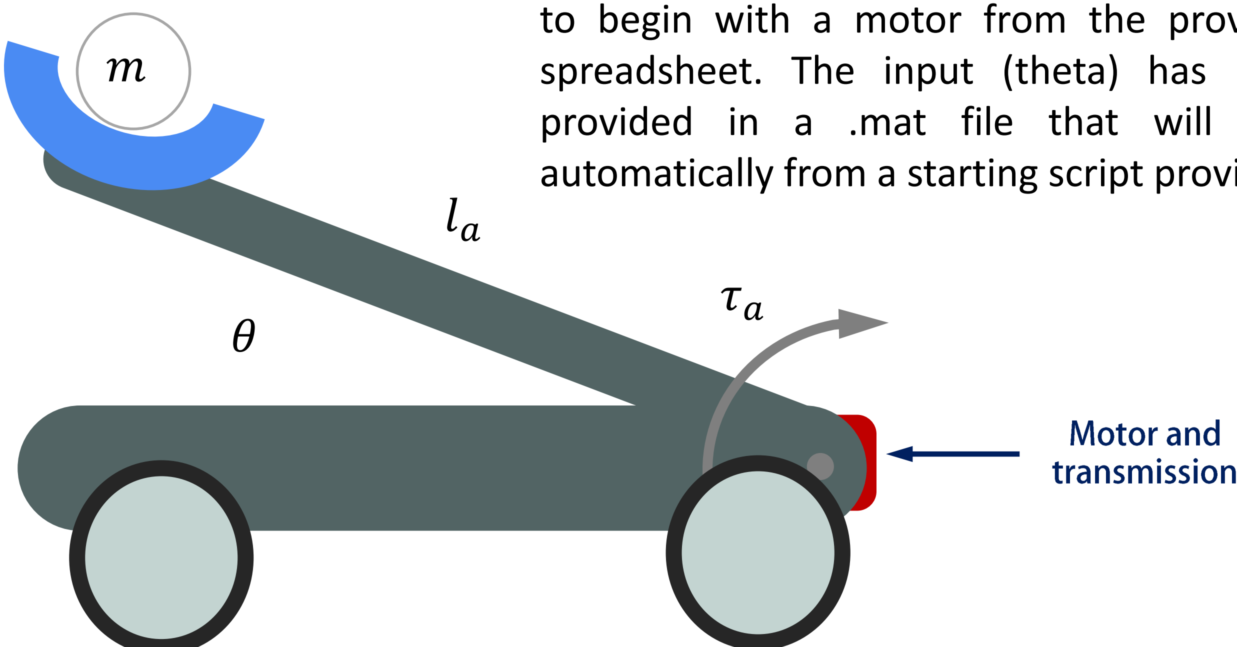
MATLAB walk through

Example – Robotic catapult



$J = wl^2$

Goal: Determine the motor current, voltage, torque, and speed for the task. You will need to begin with a motor from the provided spreadsheet. The input (theta) has been provided in a .mat file that will load automatically from a starting script provided.



$$m = 10\ kg$$
$$l_a = 1\ m$$
$$\eta = 0.8$$

$$k_t i_w = \tau = \left(J + \frac{m{l_a}^2}{N^2}\right)\ddot{\varphi} + b\dot{\varphi} + \frac{mgl\cos\theta}{\eta N}$$

$$i_w = \frac{\left(J + \frac{m{l_a}^2}{N^2}\right)N\ddot{\theta} + bN\dot{\theta} + \frac{mgl\cos\theta}{\eta N}}{k_t}$$

$$\varphi = N\theta$$

$$v = i_w R + k_t N\dot{\theta} + L\frac{di_w}{dt}$$