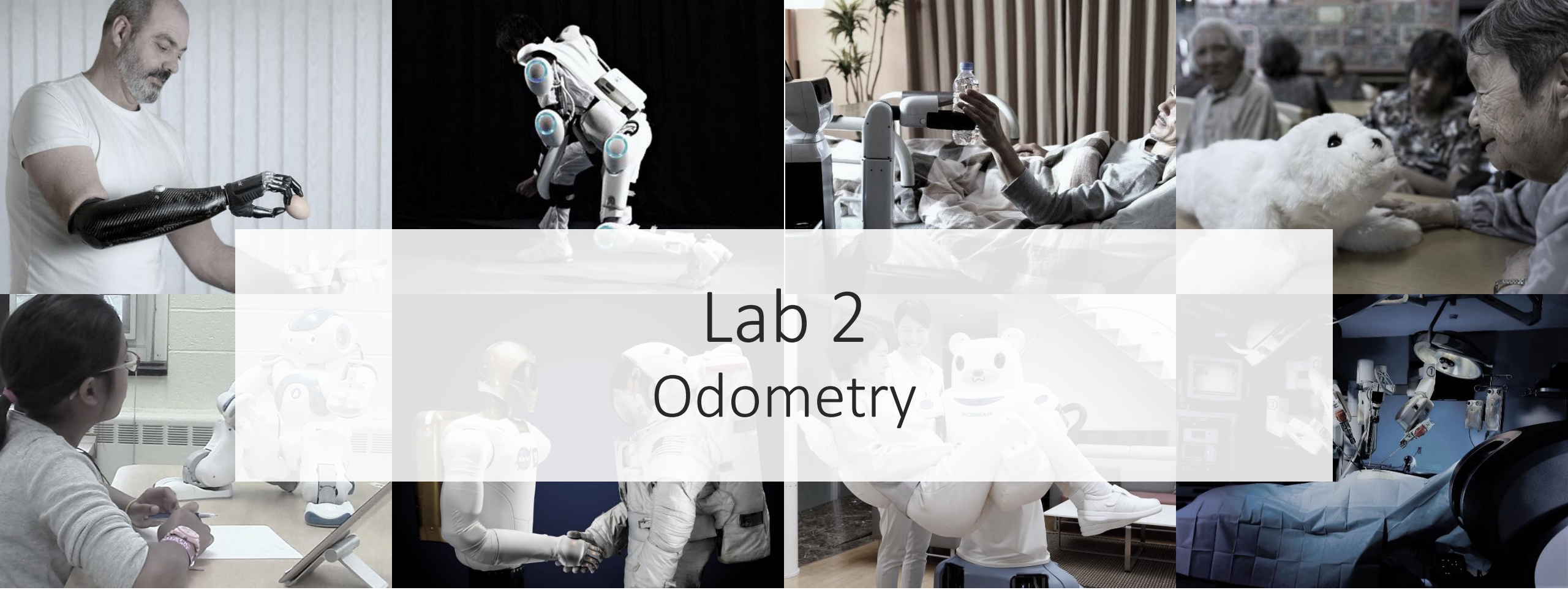
CSCI 3302
Introduction to Robotics

Alessandro Roncone

aroncone@colorado.edu

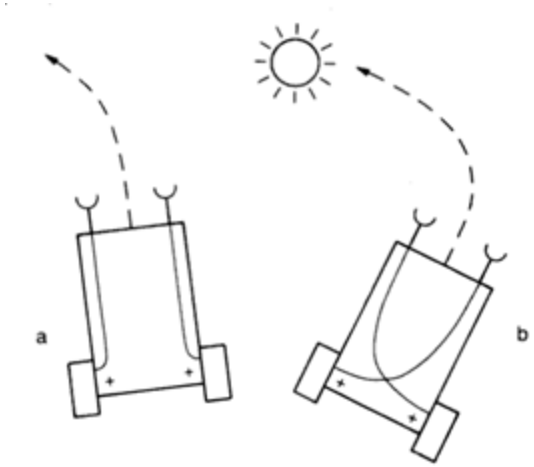


Lab 2
Odometry

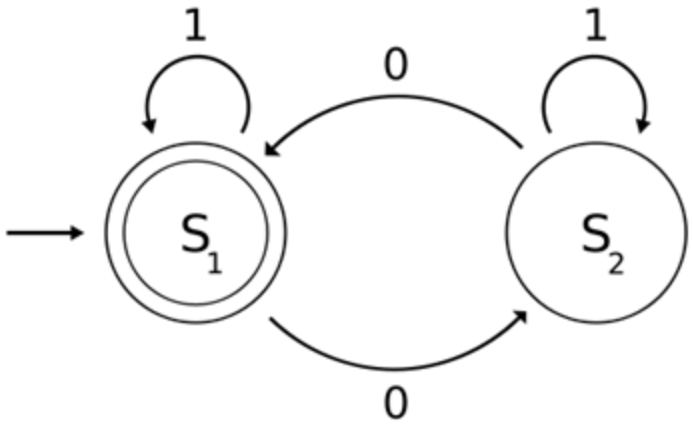
Agenda

- Lab Group Assignments
- Review from Last Lab
- Introduce Lab 2

Last Time

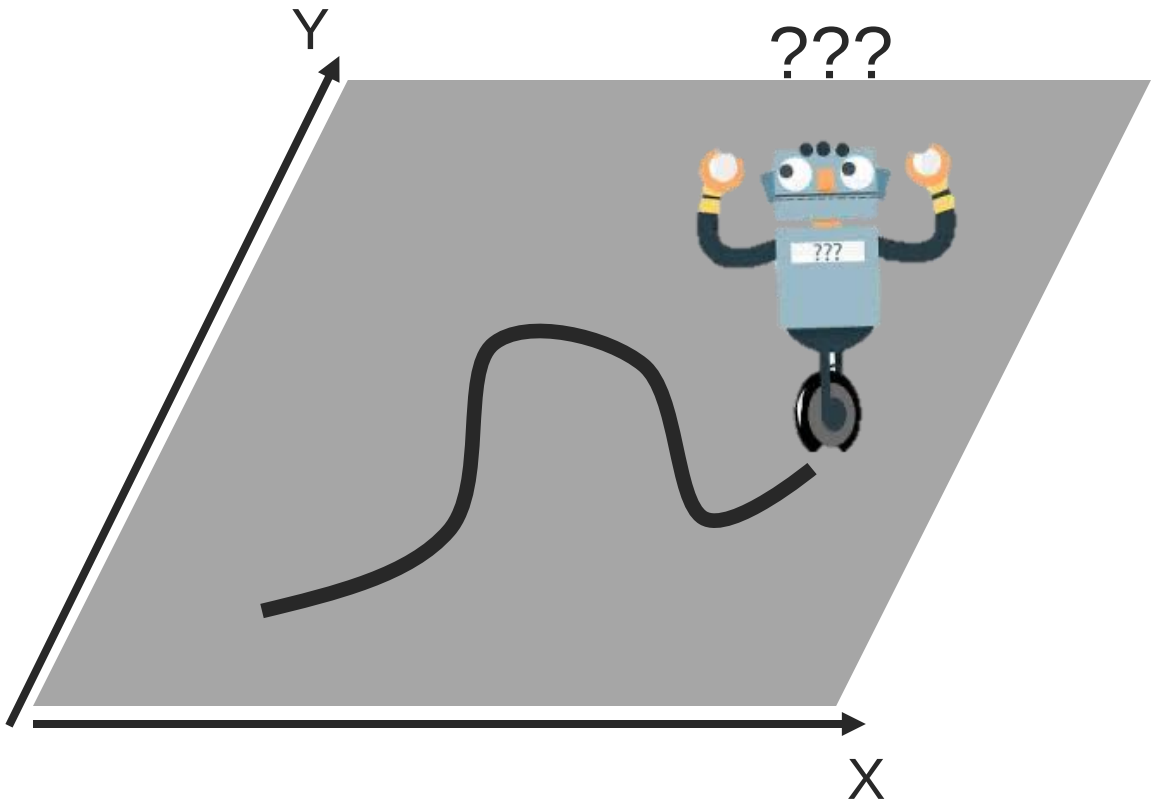


Reactive Behaviors



State Machines

Today: Odometry



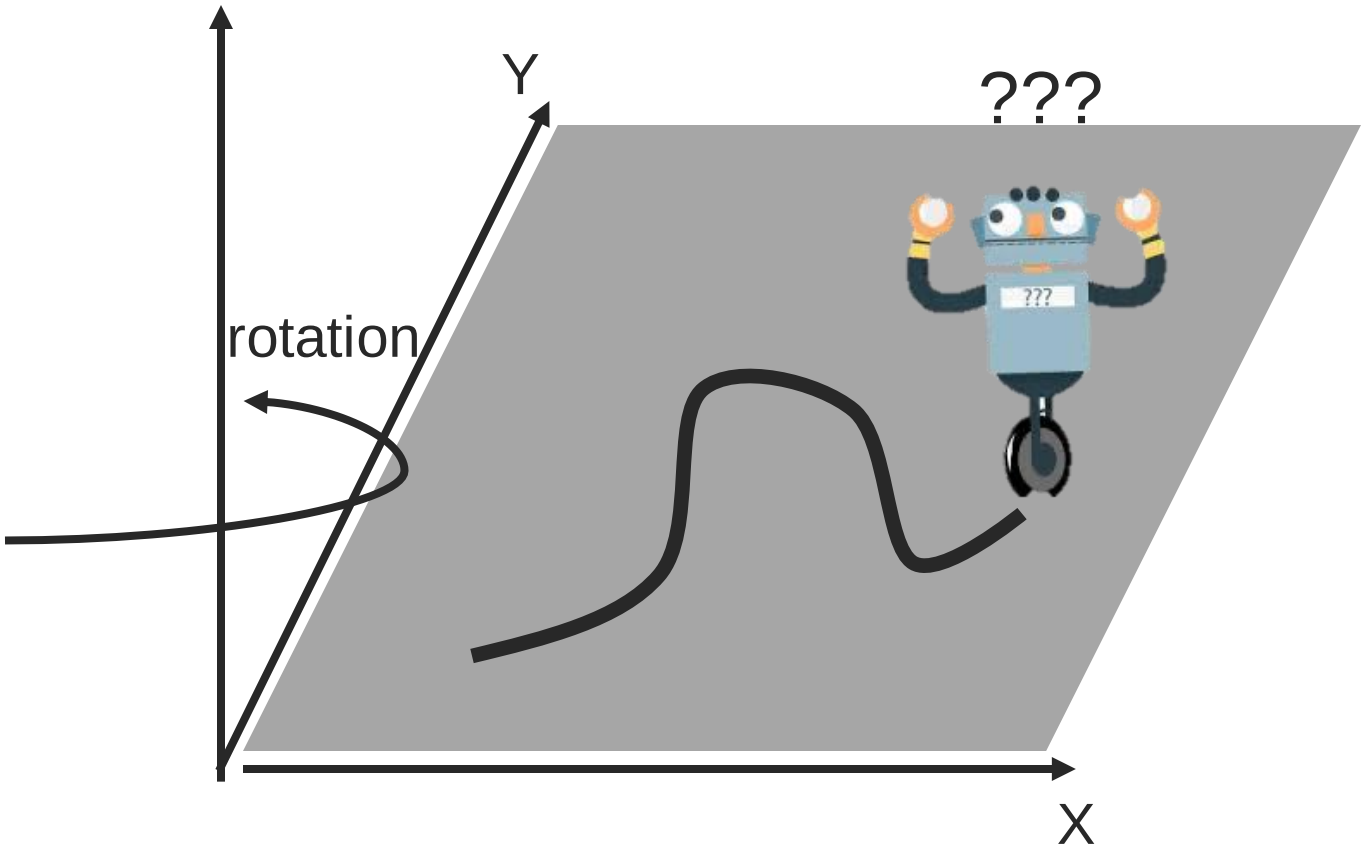
Odometer:

Integrates velocity

Example from last lab: when do I know I turned 180 degrees?

Question: What variable is missing to fully define the robot's pose on the plane?

Today: Odometry

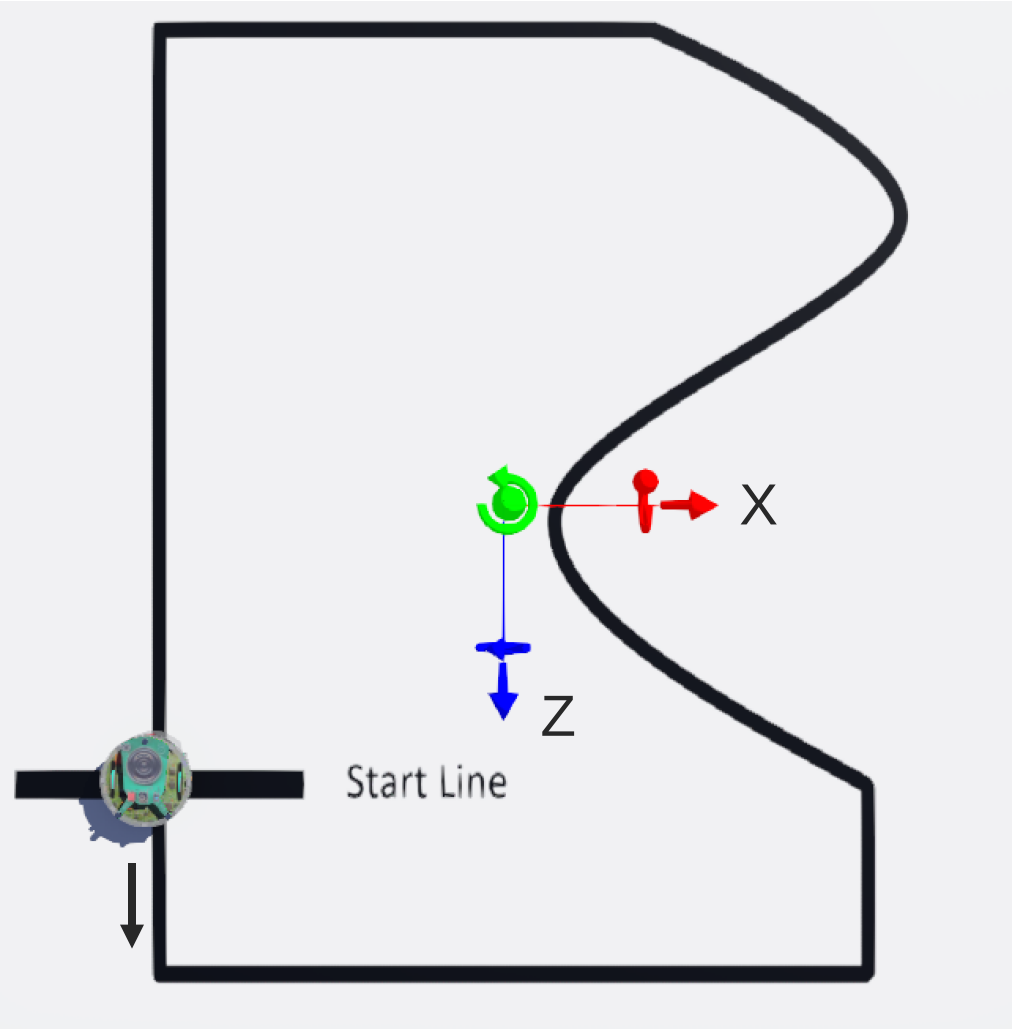


Odometer:

Integrates velocity

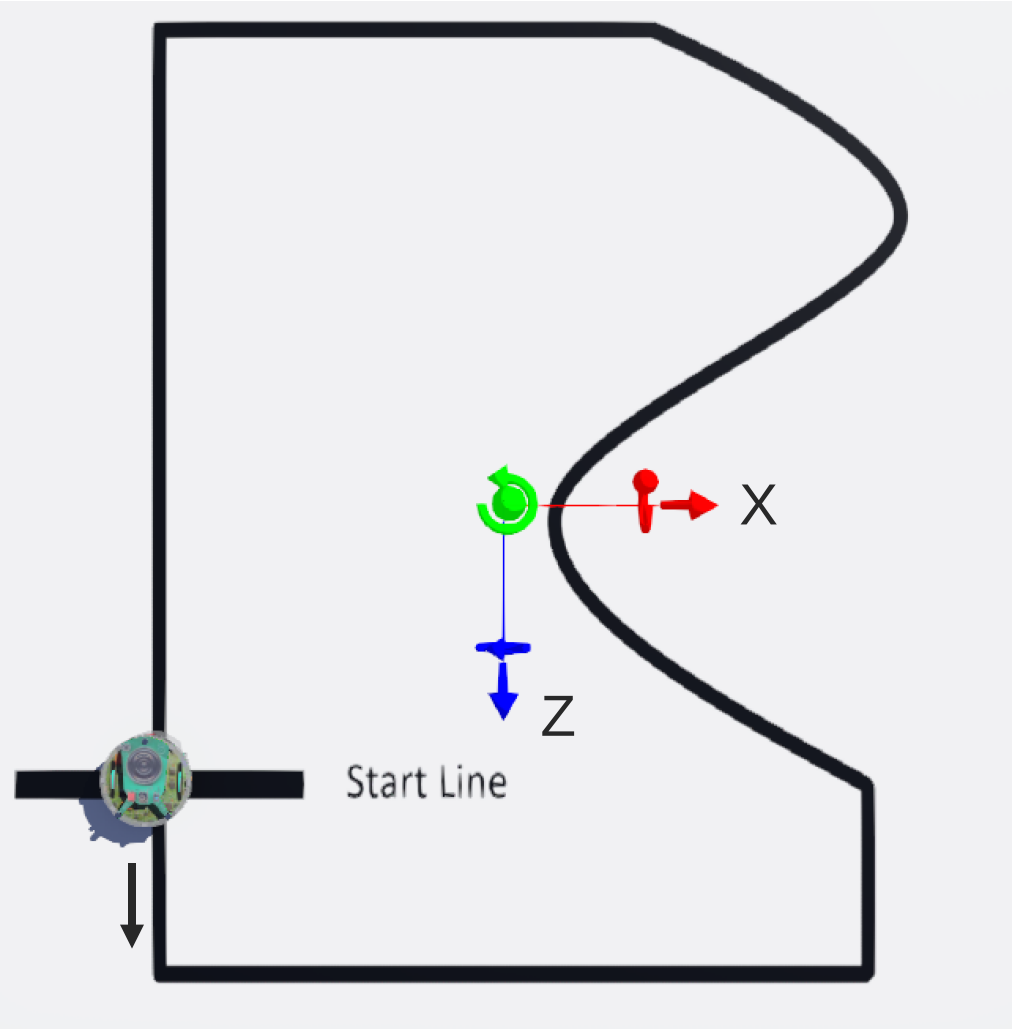
Example from last lab: when do I know I turned 180 degrees?

Lab Setup



- Which direction is the robot driving in the world coordinate frame?
- What unit is the robot's wheel-speed?
- How many meters is this per second?
- What is its pose after one complete tour?

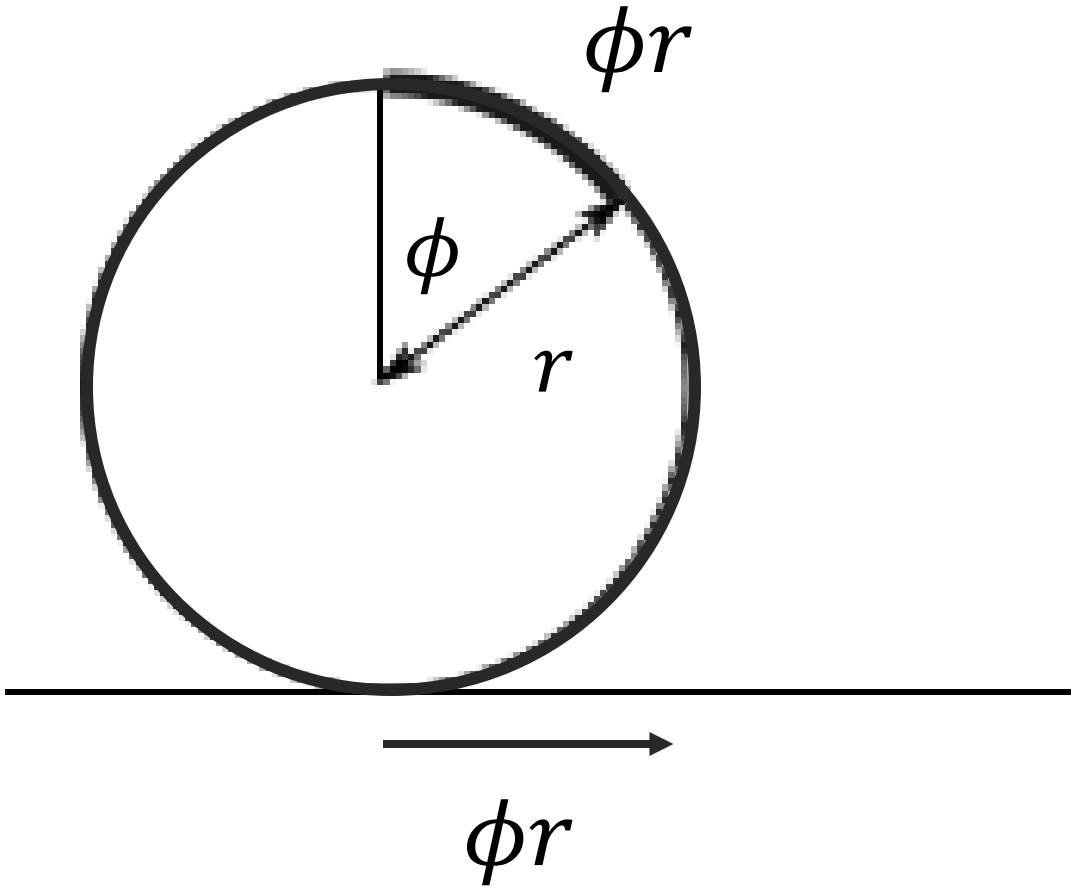
Lab Setup

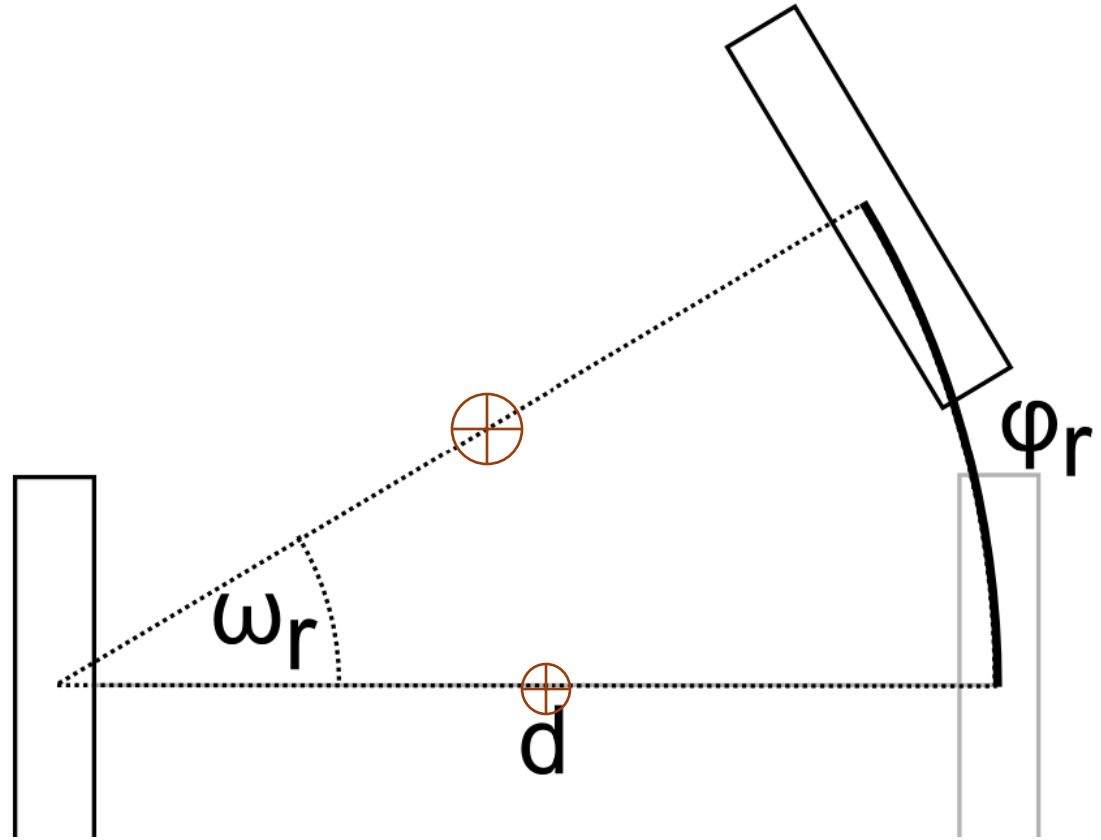


- Which direction is the robot driving in the world coordinate frame? *Along positive Z*
- What unit is the robot's wheel-speed? *Rad/s*
- How many meters is this per second? *Radius*Rad/s*
- What is its pose after one complete tour? *(0,0,0)*

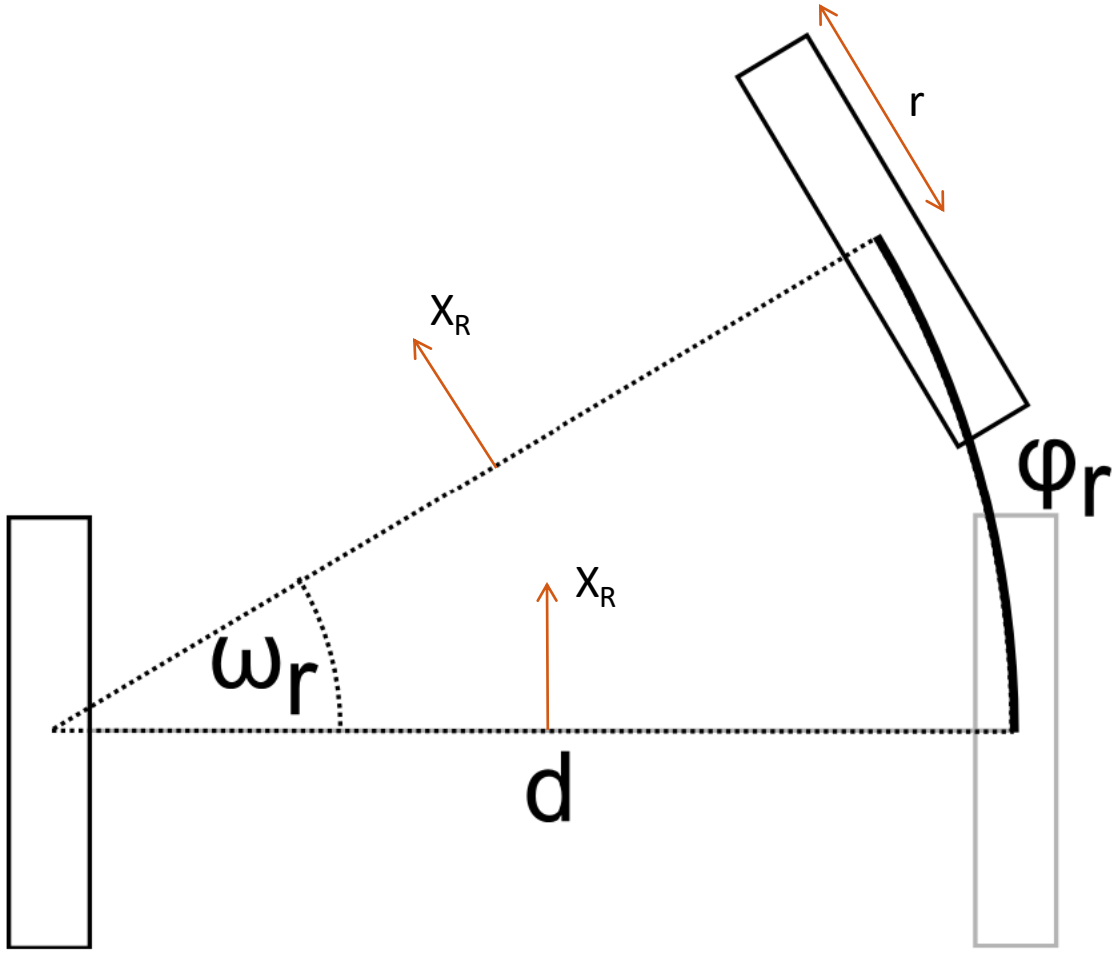
How can I find out how fast the robot is?

Angular and forward speed





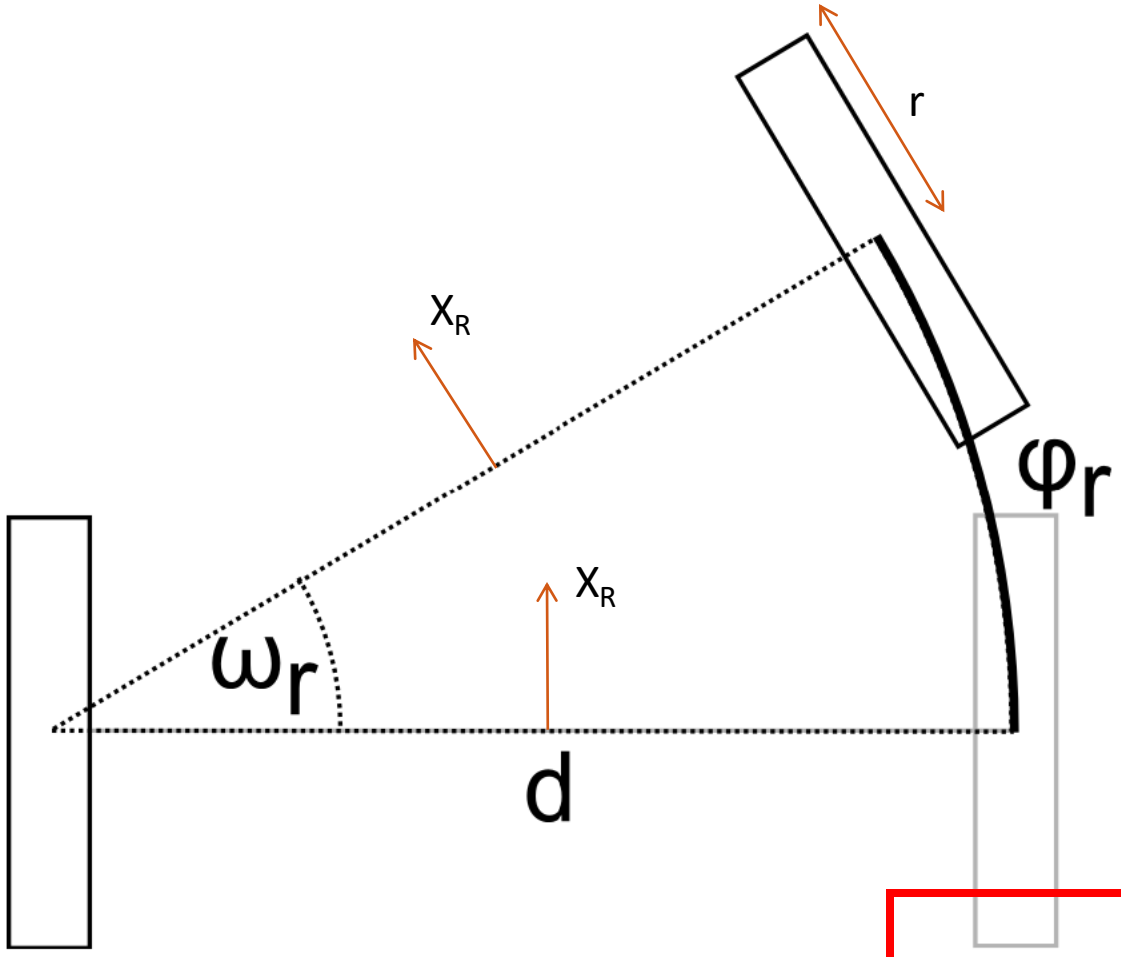
Forward Kinematics of a Mobile Robot



$$\dot{x}_r = \frac{r\dot{\phi}_r}{2}$$

$$\omega_r d = \phi_r r \rightarrow \dot{\omega}_r = \frac{\dot{\phi}_r r}{d}$$

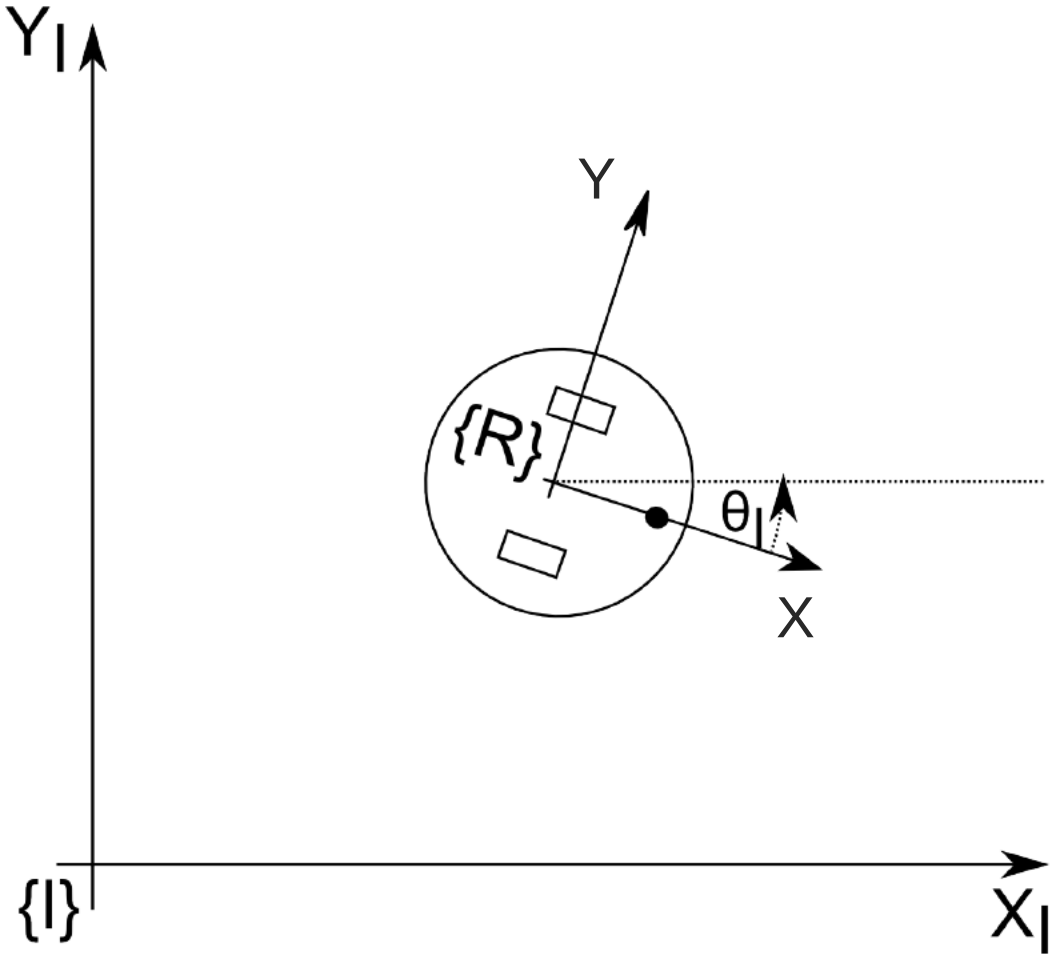
Superposition of velocities



$$\dot{x}_r = \frac{r\dot{\phi}_l}{2} + \frac{r\dot{\phi}_r}{2}$$

$$\omega_r d = \phi_r r \rightarrow \dot{\omega}_r = \frac{\dot{\phi}_r r}{d} - \frac{\dot{\phi}_l r}{d}$$

Forward Kinematics of a Mobile Robot



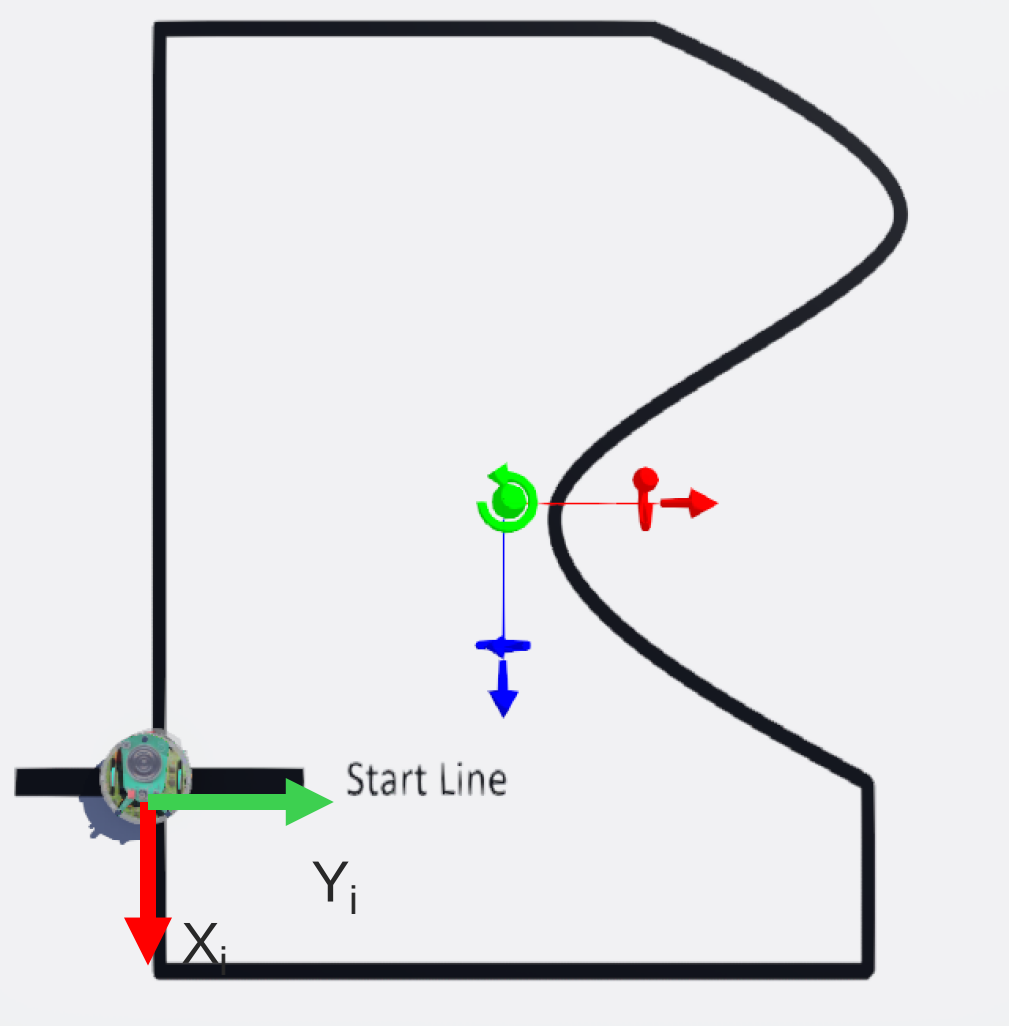
$$\dot{x_I} = cos(\theta)\dot{x_R} - sin(\theta)\dot{y_R}$$

$$\dot{y_I} = sin(\theta)\dot{x_R} + cos(\theta)\dot{y_R}$$

$$\dot{\theta_I} = \dot{\theta_R}$$

$$\begin{pmatrix} \dot{x_I} \\ \dot{y_I} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{r\dot{\phi_l}}{2} + \frac{r\dot{\phi_r}}{2} \\ 0 \\ \frac{\dot{\phi_r} r}{d} - \frac{\dot{\phi_l} r}{d} \end{pmatrix}$$

Your coordinate system

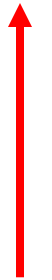


How can we compute position?

$$\begin{pmatrix} x_I(T) \\ y_I(T) \\ \theta(T) \end{pmatrix} = \boxed{} \begin{pmatrix} \dot{x}_I(t) \\ \dot{y}_I(t) \\ \dot{\theta}(t) \end{pmatrix} \boxed{}$$

How can we compute position?

$$\begin{pmatrix} x_I(T) \\ y_I(T) \\ \theta(T) \end{pmatrix} = \int_0^T \begin{pmatrix} \dot{x_I}(t) \\ \dot{y_I}(t) \\ \dot{\theta}(t) \end{pmatrix} dt \approx \sum_{k=0}^{k=T} \begin{pmatrix} \Delta x_I(k) \\ \Delta y_I(k) \\ \Delta \theta(k) \end{pmatrix} \Delta t$$



int(robot.getBasicTimeStep())

From linear algebra/diff equations to code

$$\begin{pmatrix} \dot{x_I} \\ \dot{y_I} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{r\dot{\phi_l}}{2} + \frac{r\dot{\phi_r}}{2} \\ 0 \\ \frac{\dot{\phi_r} r}{d} - \frac{\dot{\phi_l} r}{d} \end{pmatrix}$$

1. Robot never moves sideways

$$\begin{pmatrix} \dot{x_I} \\ \dot{y_I} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{r\dot{\phi_l}}{2} + \frac{r\dot{\phi_r}}{2} \\ \frac{\dot{\phi_r} r}{d} - \frac{\dot{\phi_l} r}{d} \end{pmatrix}$$

2. Multiply matrices

$$\begin{aligned} \dot{x_I} &= cos(\theta) \left(\frac{r\dot{\phi_l}}{2} + \frac{r\dot{\phi_r}}{2} \right) \\ \dot{y_I} &= sin(\theta) \left(\frac{r\dot{\phi_l}}{2} + \frac{r\dot{\phi_r}}{2} \right) \\ \dot{\theta} &= \frac{\dot{\phi_r} r}{d} - \frac{\dot{\phi_l} r}{d} \end{aligned}$$

3. Integrate

$$\begin{pmatrix} x_I(T) \\ y_I(T) \\ \theta(T) \end{pmatrix} \mathrel{+}= \begin{pmatrix} \Delta x_I(k) \\ \Delta y_I(k) \\ \Delta\theta(k) \end{pmatrix} \Delta t$$

Sources of Error

Deterministic

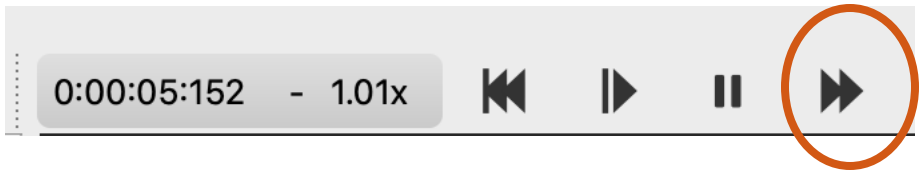
- Unequal wheel diameter
- Wheel contact point variation

Addressed with calibration

Non-deterministic

- Variable friction
- Measurement noise

Code Suggestions



- Start implementing line following. Do NOT move forward before line following is not robust.
- Use an if/elif chain to check which state your system is in (unless you want to do it the Pythonic way)
- Don’t ever use time.sleep() or estimate the elapsed time, instead skip steps in the loop and call robot_step()

```
current_state = ‘STATE_1’
while robot.step(TIME_STEP) != -1:
    if current_state == 'STATE_1’:
                        pass
    elif current_state = ‘STATE_2’:
                        pass
```

Need Help? Send a chat message!

Group programming IDEs: www.codeshare.io | http://collabedit.com/

FAQs:

- What's due at the end of today?
- Nothing, this is a 2-week lab. Due Tuesday 2/21 at 11:59pm.
- What do I turn in?
  - ONE person needs to turn in the lab report and code per group.
- We're done! Can we leave?
  - Yep. Lab is meant to provide an interactive problem-solving time. If you complete the work early, you are free to go!

Questions?