Prof. Kirstin Hagelskjær Petersen <u>kirstin@cornell.edu</u>

...with help from Vivek Thangavelu Alex Coy Jonathan Jaramillo Cameron Urban

Fast Robots Lab 10 (Flipped classroom)



ECE 4160/5160 MAE 4910/5910

Simulation Software

- Multiple processes
 - Simulator
 - Robot
 - Motion
 - Ground truth
 - YAML (map and other parameters)
 - Plotter
 - Controller
 - Get odometry pose, get and plot sensor data, move the robot, etc.
- Why do we bother with the simulator?
 - Helpful for debugging the Bayes Filter
 - Helpful for evaluating implication of accuracy
 - Can be used to also debug the real robot

Jupyter Notebook

Spawns the Simulator
and Plotter processes
Communicates with
them using pipes







Simulator (today)

Commander

Functions to interact with the simulator and the plotter



Task 1: Control and plotting

- Keyboard control of robot
 - H brings up a key-map in the plotter
- Programmatically control your robot and visualize the trajectory in the plotter

4

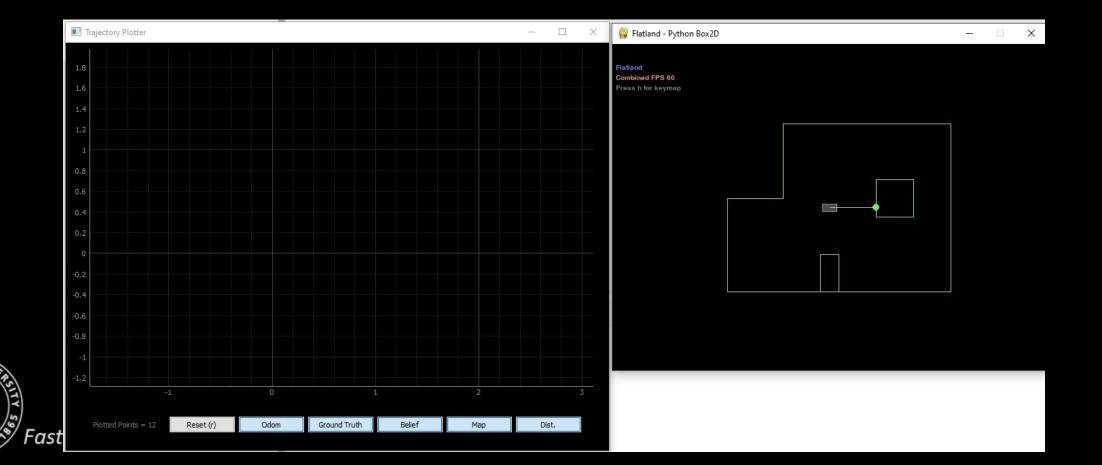
- How to add delays
 - time.sleep(1)
 - await asyncio.sleep(1)



Task 2: Open loop control

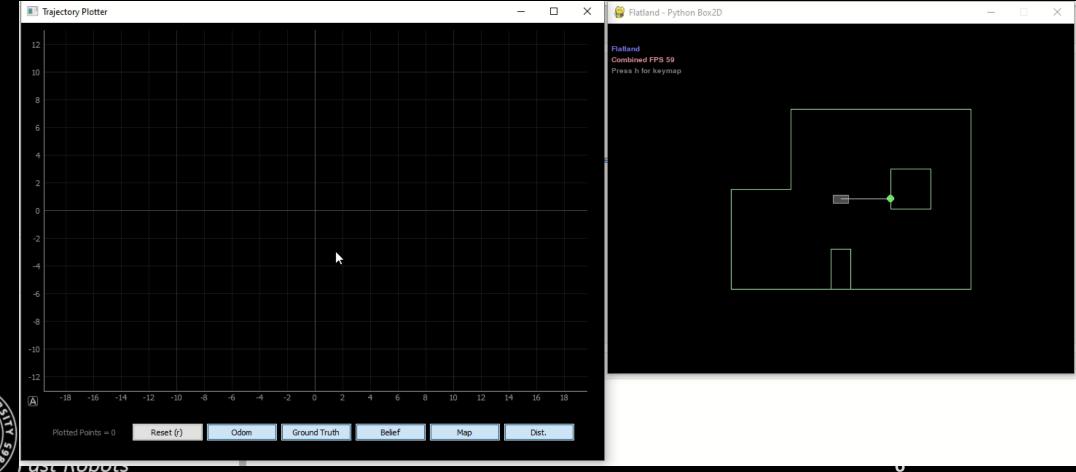
- Make your robot drive in a square loop
 - Compare ground truth and odometry
 - Compare across computers

- Why do think solutions differ across computers?
 - System's load, specs, rounding, etc.



Task 3: Closed loop control and obstacle avoidance

- Design a simple controller in Jupyter to avoid obstacles
 - Suggestions?





Task 3: Closed loop control and obstacle avoidance

- Design a simple controller in Jupyter to avoid obstacles
 - Suggestions?

- Consider...
 - Turn size and forward speed
 - How close can the virtual robot get to an obstacle without colliding?
 - Does your obstacle avoidance code always work? If not, what can you do to minimize crashes or (maybe) prevent them completely?

7



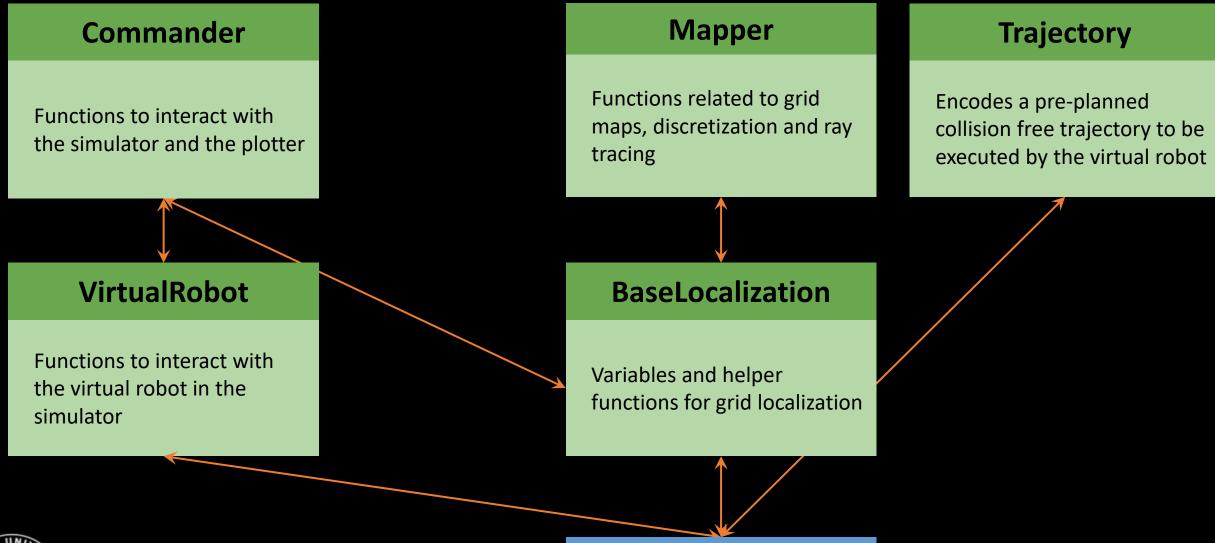
Task 3: Closed loop control and obstacle avoidance

- Design a simple controller in Jupyter to avoid obstacles
 - Suggestions?

Trajectory Plotter	>	Search Statland - Python Box2D - X
4		Flatland Combined FPS 60 Press h for keymap
2		
1 0		
-1 -2	▶	
-3		
-4 -5		
▲ -6 -4 -2	0 2 4 6	
Plotted Points = 0 Reset (r) Odom	Ground Truth Belief Map Dist.	
Fast novots		



Lab 10 - Localization on the virtual robot





Grid Localization Code

Lab 11 - Pure localization on the virtual robot

Commander	ArtemisBLE	Mapper	Trajectory
Functions to interact with the simulator and the plotter	Controller Functions to interact with the real robot	Functions related to grid maps, discretization and ray tracing	Encodes a pre-planned collision free trajectory to be executed by the virtual robot
			(Prediction class
RealRobot		BaseLocalization	is not used)
Functions to interact with the real robot.		Variables and helper functions for grid localization	
D D		Localization	
Fast Robots		Grid Localization Code	

Lab 12 - Localization and planning on the virtual robot

Commander	ArtemisBLE	Mapper	Trajectory
Functions to interact with the simulator and the plotter	Controller Functions to interact with the real robot	Functions related to grid maps, discretization and ray tracing	Encodes a pre-planned collision free trajectory to be executed by the real robot
RealRobot		BaseLocalization	
Functions to interact with the real robot.		Variables and helper functions for grid localization	
UNIVERSION DE DE		Localization	
Fast Robots		Grid Localization Code	

Logistics

• Lab 8 – Stunts

- Voting can start today Friday
- Please submit your votes by Friday April 22nd
 - <u>https://tinyurl.com/vp5wrten</u>
 - 10 points for best stunt
 - 1 point for best blooper
- Lab 9 Catch up now!
- Lab 10 Localization
 - Get a head start on the Lab 10 documentation!

