AuE-8230
Autonomy: Science and Systems

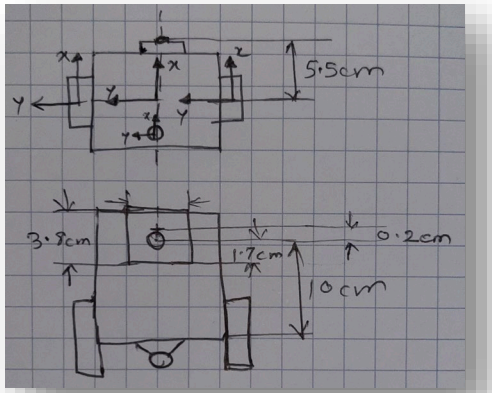
Capstone Project

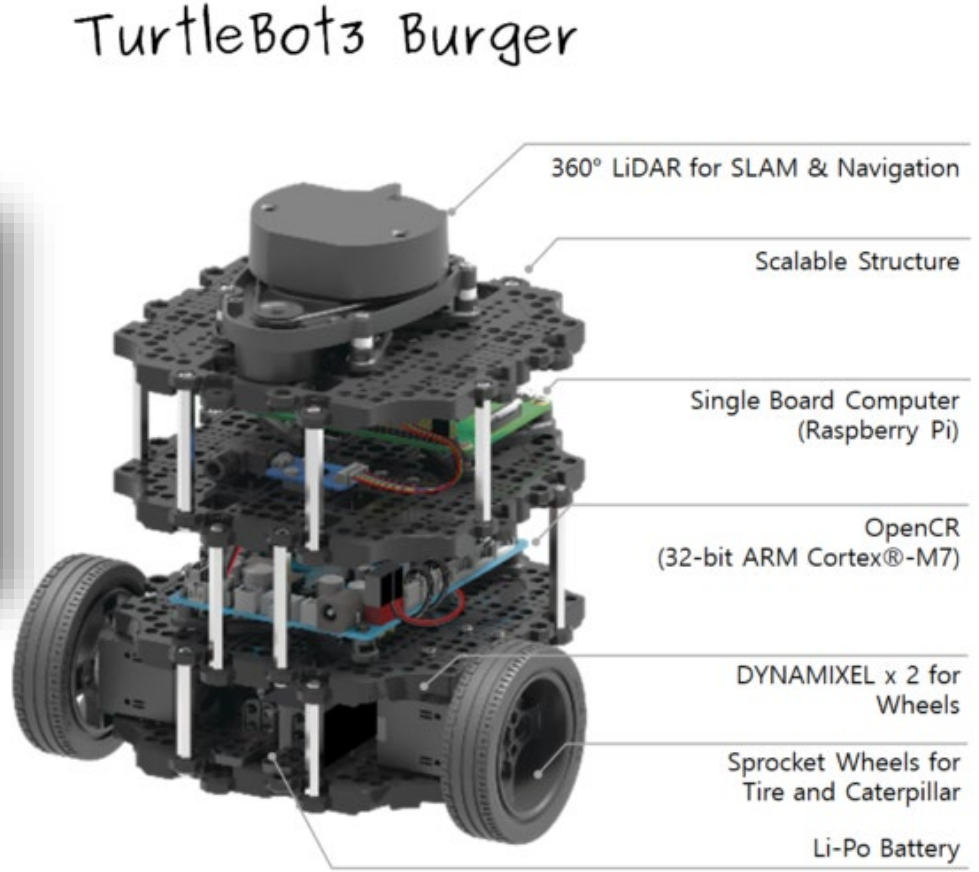
Group 1: Chinmay Samak, Tanmay Samak

Codebase

Project Tools

- **Robot:** TurtleBot3 Burger (with camera)
- **Simulation:** Gazebo
- **Framework:** ROS 2 Foxy
- **Programming:** Python, C++

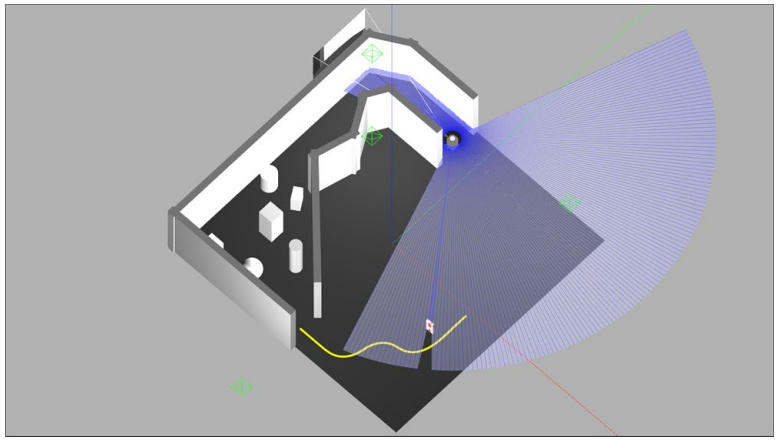




Source: Robotis Inc.

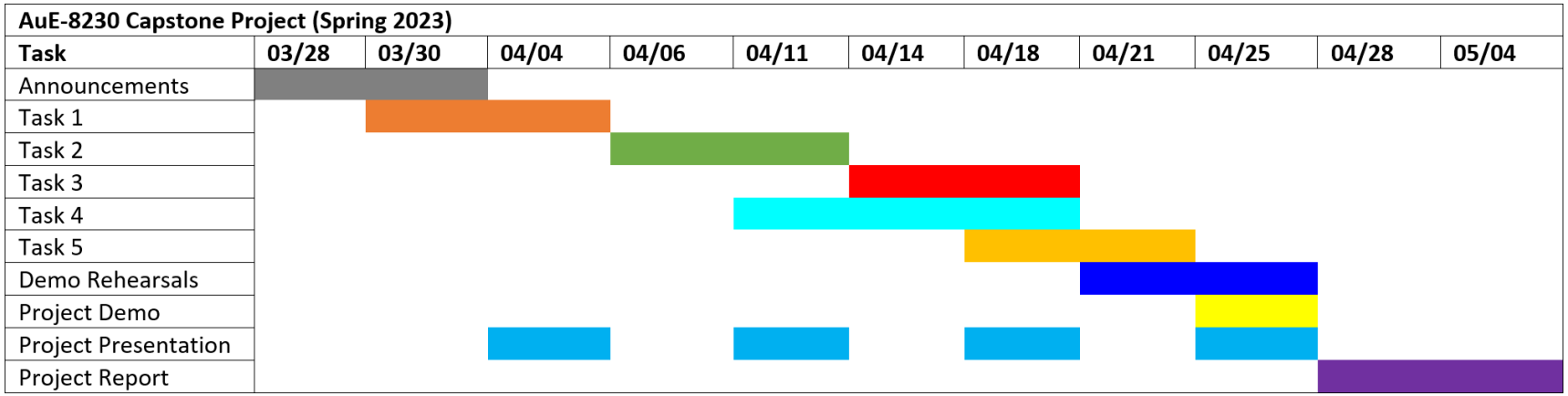
Project Tasks

- **Task 1:** Wall Following
- **Task 2:** Obstacle Avoidance
- **Task 3:** Line Following
- **Task 4:** Stop Sign Detection
- **Task 5:** AprilTag Tracking



Project Timeline

- **Task 1:** Wall Following
- **Task 2:** Obstacle Avoidance
- **Task 3:** Line Following
- **Task 4:** Stop Sign Detection
- **Task 5:** AprilTag Tracking



Responsibility Assignment Matrix

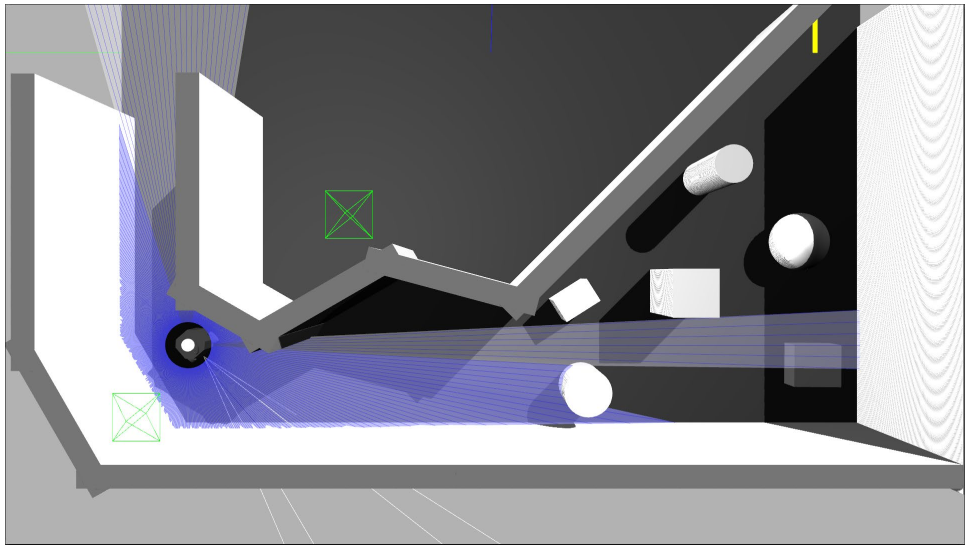
- **Task 1:** Wall Following
- **Task 2:** Obstacle Avoidance
- **Task 3:** Line Following
- **Task 4:** Stop Sign Detection
- **Task 5:** AprilTag Tracking

RESPONSIBILITY	Task 1	Task 2	Task 3	Task 4	Task 5	Integration
Algorithm development	Tanmay	Chinmay	Tanmay	Chinmay	Chinmay	Chinmay
Simulation setup	Chinmay	Tanmay	Chinmay	Tanmay	Chinmay	Chinmay
Simulation deployment	Chinmay	Tanmay	Chinmay	Tanmay	Tanmay	Chinmay
Real-world deployment	Tanmay	Chinmay	Tanmay	Chinmay	Tanmay	Tanmay
Git repository management	Chinmay	Chinmay	Chinmay	Tanmay	Chinmay	Tanmay
Documentation	Tanmay	Tanmay	Tanmay	Chinmay	Tanmay	Tanmay

***Note**: Responsibility does not directly indicate contribution. Both members contributed equally to this project and have no conflict of interest to declare.*

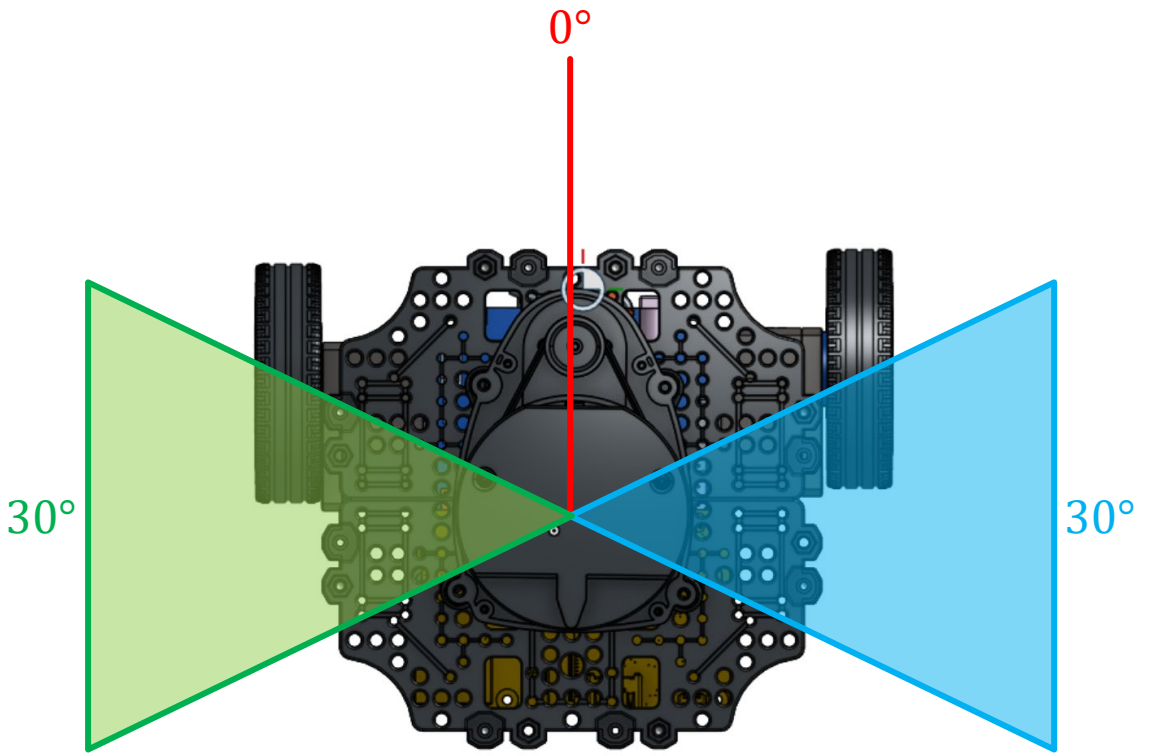
Task 1

Wall Following



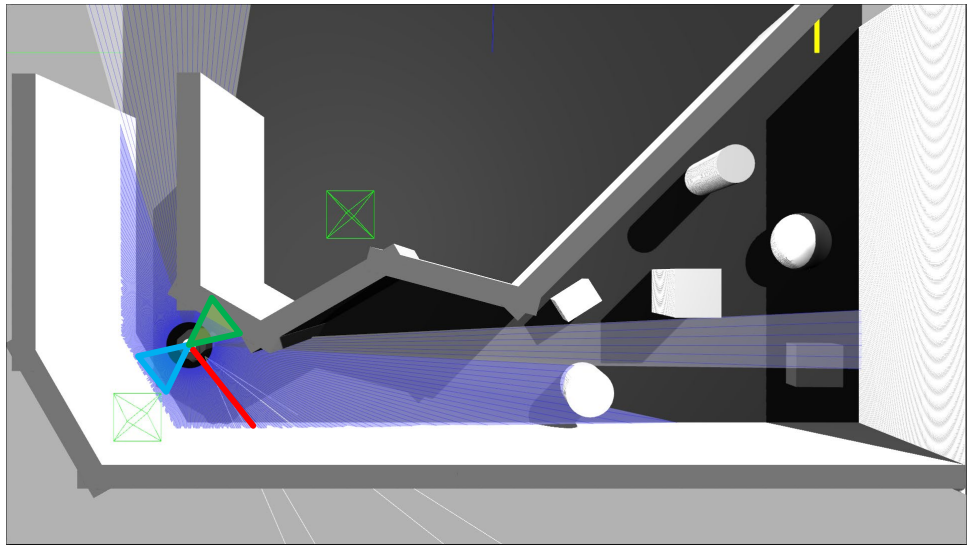
Approach

- **Sensors:** LIDAR
- **Algorithm:**
  - Front laser scan ranging measurement [0]
  - Left [75:105] and right [255:285] laser scan sectors
  - De-coupled lat-lon PID controller architecture
  - Adaptive cruise control based on frontal range
  - Cross-track error based on mean of left and right sectors
  - Safety (inf range, actuation limits)
- **Alternate approaches:**
  - Different FOV and orientation of laser scan sectors
  - CTE based on min, max, mean, wt. mean of sectors



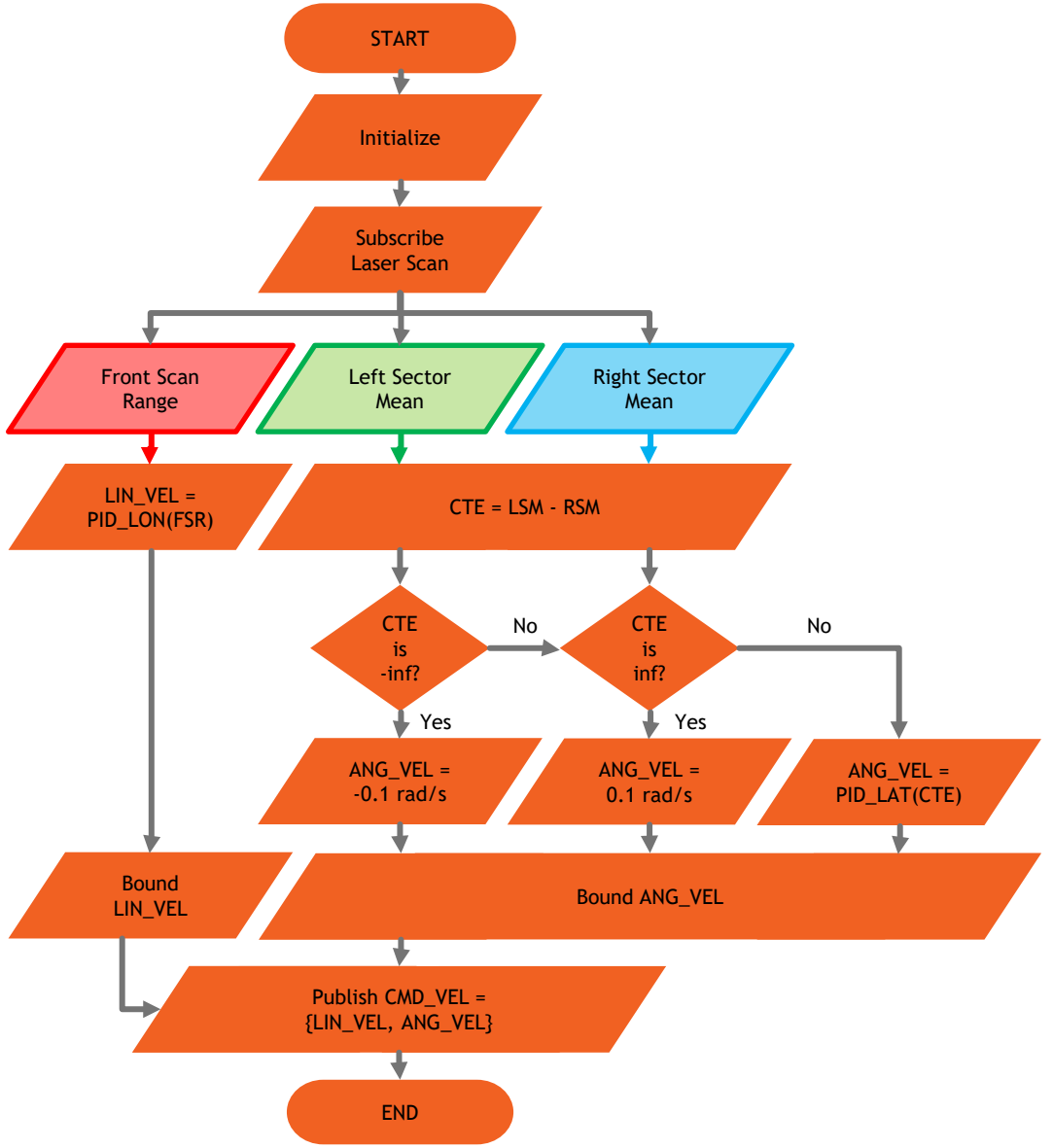
Approach

- **Sensors:** LIDAR
- **Algorithm:**
  - Front laser scan ranging measurement [0]
  - Left [75:105] and right [255:285] laser scan sectors
  - De-coupled lat-lon PID controller architecture
  - Adaptive cruise control based on frontal range
  - Cross-track error based on mean of left and right sectors
  - Safety (inf range, actuation limits)
- **Alternate approaches:**
  - Different FOV and orientation of laser scan sectors
  - CTE based on min, max, mean, wt. mean of sectors

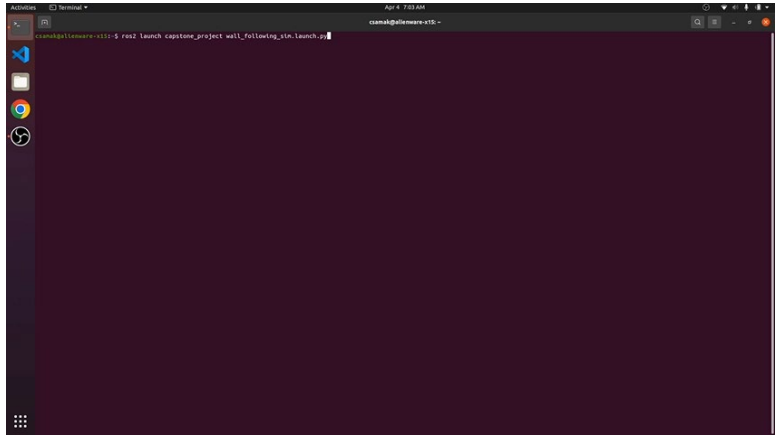


Approach

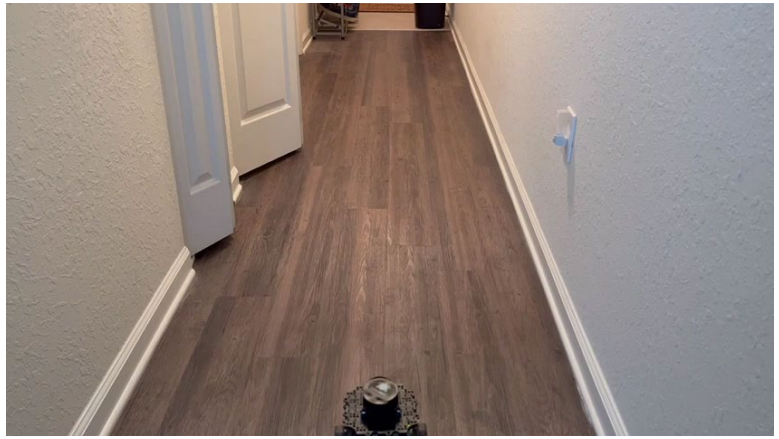
- **Sensors:** LIDAR
- **Algorithm:**
  - Front laser scan ranging measurement [0]
  - Left [75:105] and right [255:285] laser scan sectors
  - De-coupled lat-lon PID controller architecture
  - Adaptive cruise control based on frontal range
  - Cross-track error based on mean of left and right sectors
  - Safety (inf range, actuation limits)
- **Alternate approaches:**
  - Different FOV and orientation of laser scan sectors
  - CTE based on min, max, mean, wt. mean of sectors



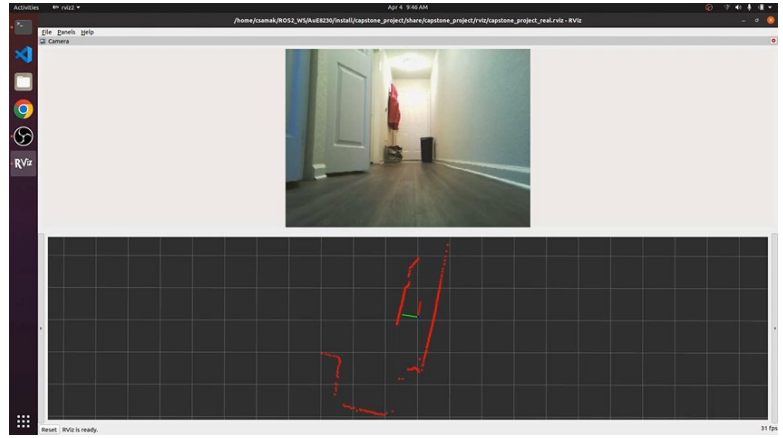
Results



Simulation



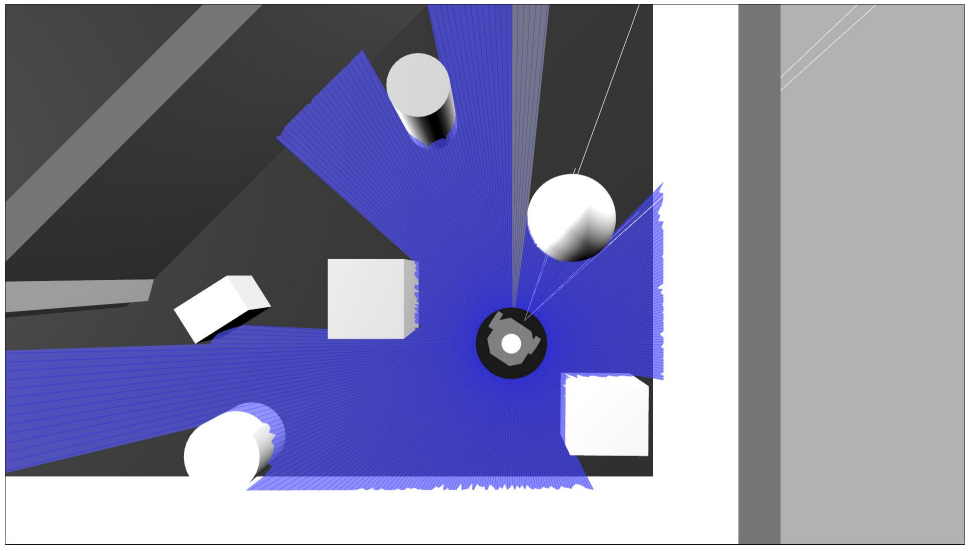
TurtleBot3



Remote PC

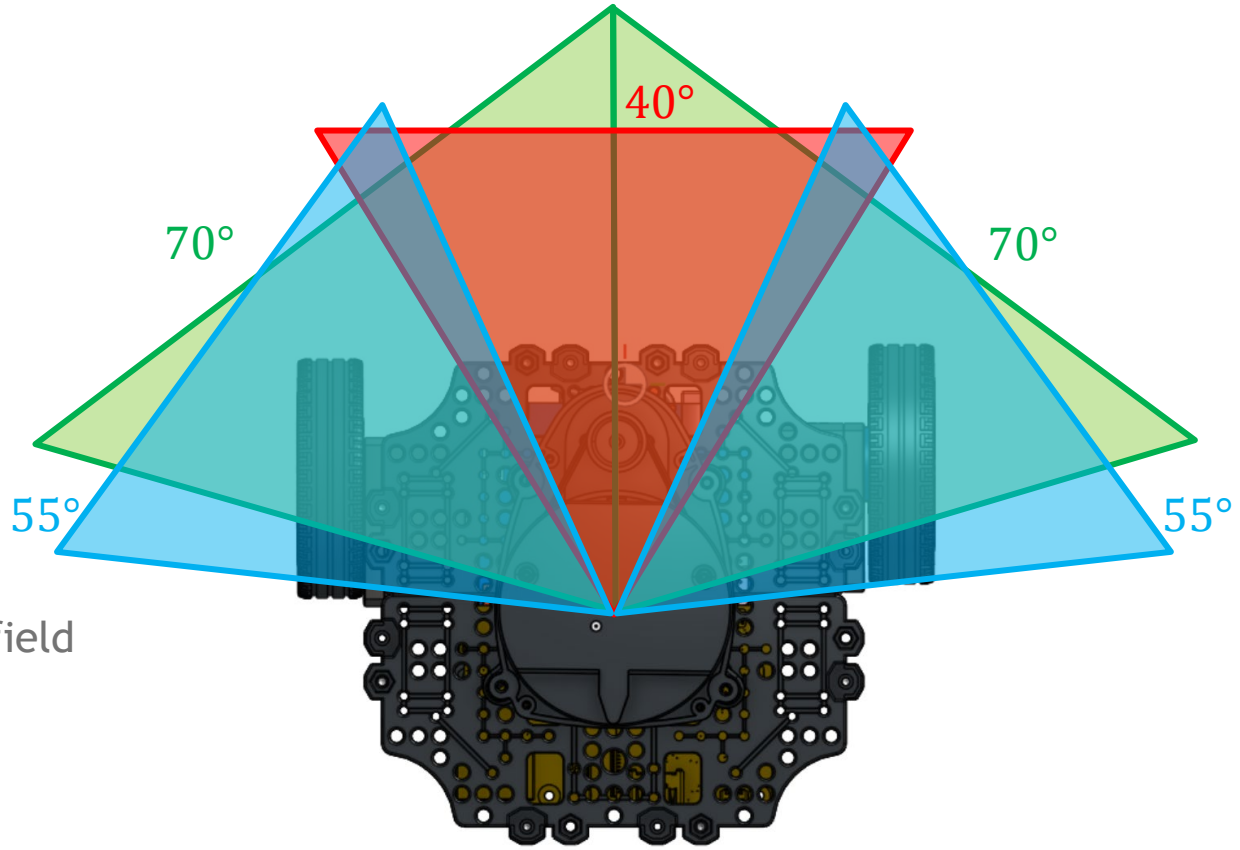
Task 2

Obstacle Avoidance



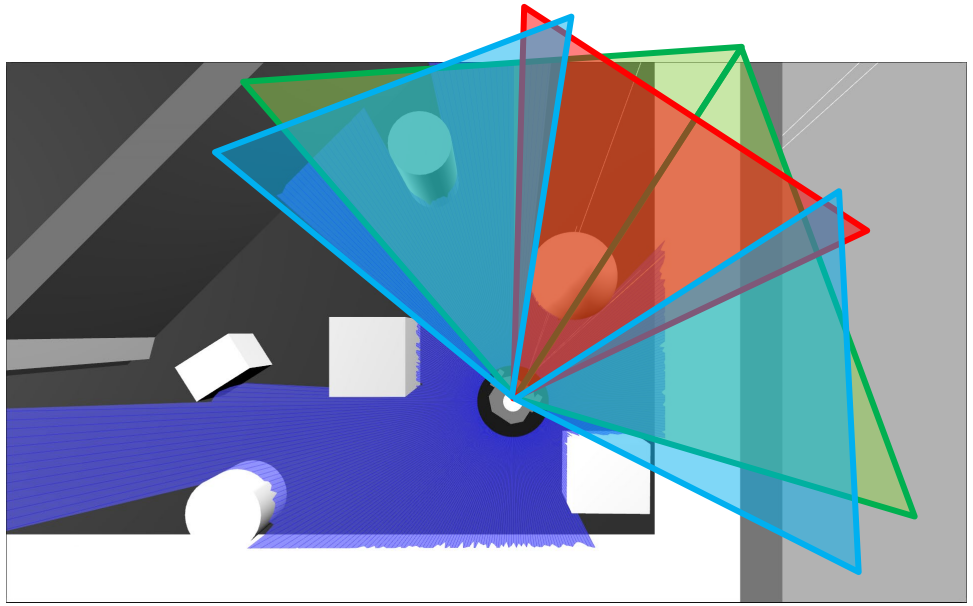
Approach

- **Sensors:** LIDAR
- **Algorithm:**
  - Front laser scan sector [0:20]+[340:360]
  - Oblique laser scan sectors [0:70] and [290:360]
  - Side laser scan sectors [30:85] and [275:330]
  - 3-stage attention mechanism + pseudo potential field
  - De-coupled lat-lon PID controller architecture
  - Safety (inf range, on-spot turn, actuation limits)
- **Alternate approaches:**
  - Different FOV and orientation of laser scan sectors
  - CTE based on min, max, mean, wt. mean of sectors
  - Safety vs. performance



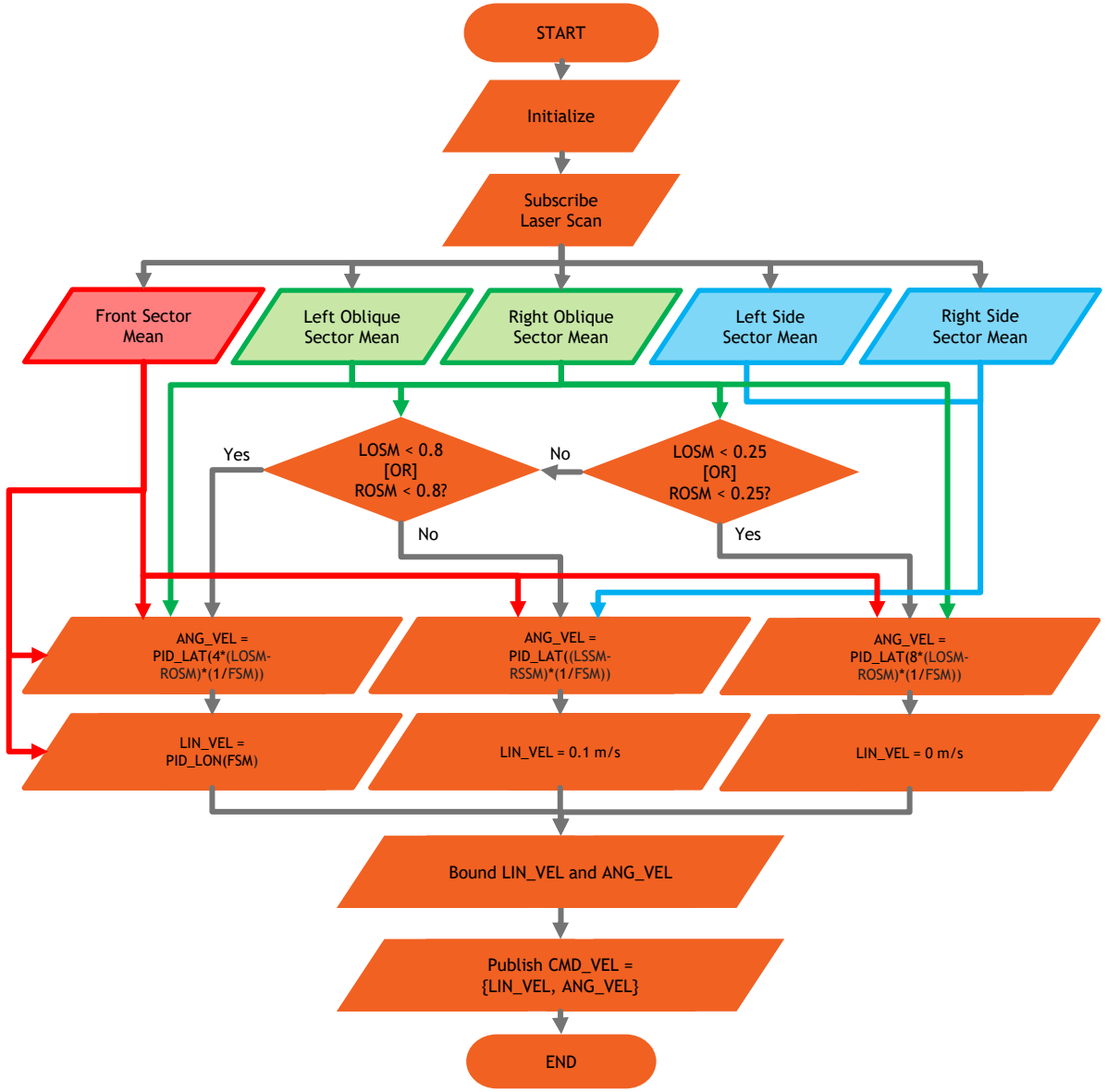
Approach

- **Sensors:** LIDAR
- **Algorithm:**
  - Front laser scan sector [0:20]+[340:360]
  - Oblique laser scan sectors [0:70] and [290:360]
  - Side laser scan sectors [30:85] and [275:330]
  - 3-stage attention mechanism + pseudo potential field
  - De-coupled lat-lon PID controller architecture
  - Safety (inf range, on-spot turn, actuation limits)
- **Alternate approaches:**
  - Different FOV and orientation of laser scan sectors
  - CTE based on min, max, mean, wt. mean of sectors
  - Safety vs. performance

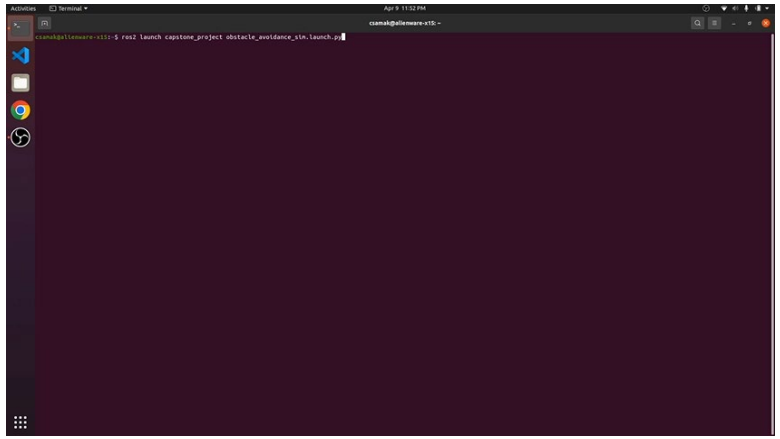


Approach

- **Sensors:** LIDAR
- **Algorithm:**
  - Front laser scan sector [0:20]+[340:360]
  - Oblique laser scan sectors [0:70] and [290:360]
  - Side laser scan sectors [30:85] and [275:330]
  - 3-stage attention mechanism + pseudo potential field
  - De-coupled lat-lon PID controller architecture
  - Safety (inf range, on-spot turn, actuation limits)
- **Alternate approaches:**
  - Different FOV and orientation of laser scan sectors
  - CTE based on min, max, mean, wt. mean of sectors
  - Safety vs. performance

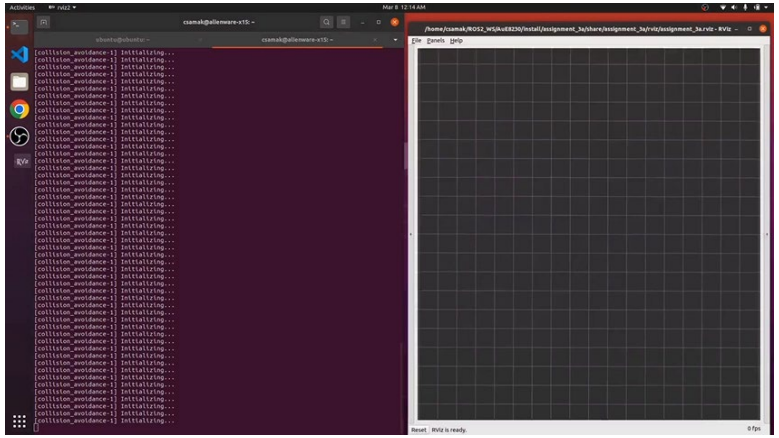


Results



Simulation

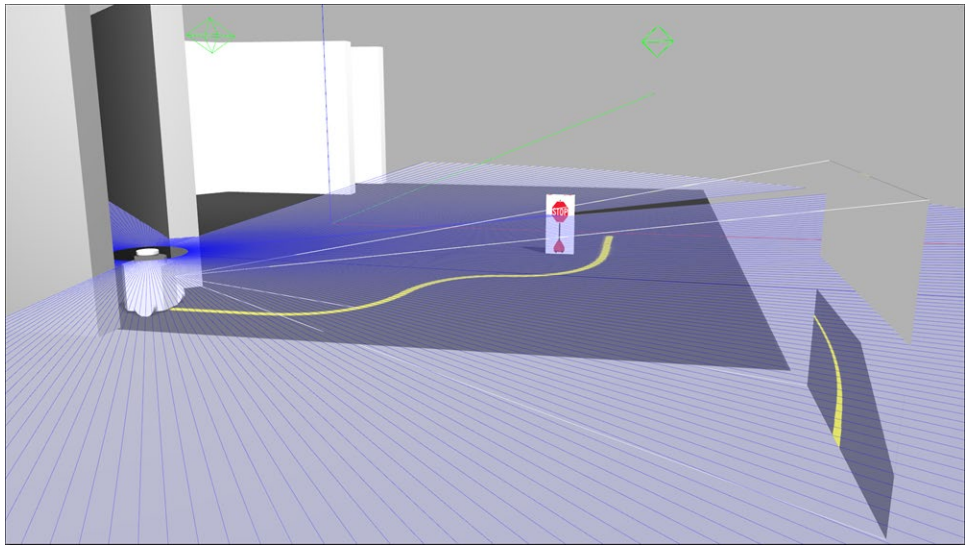
TurtleBot3

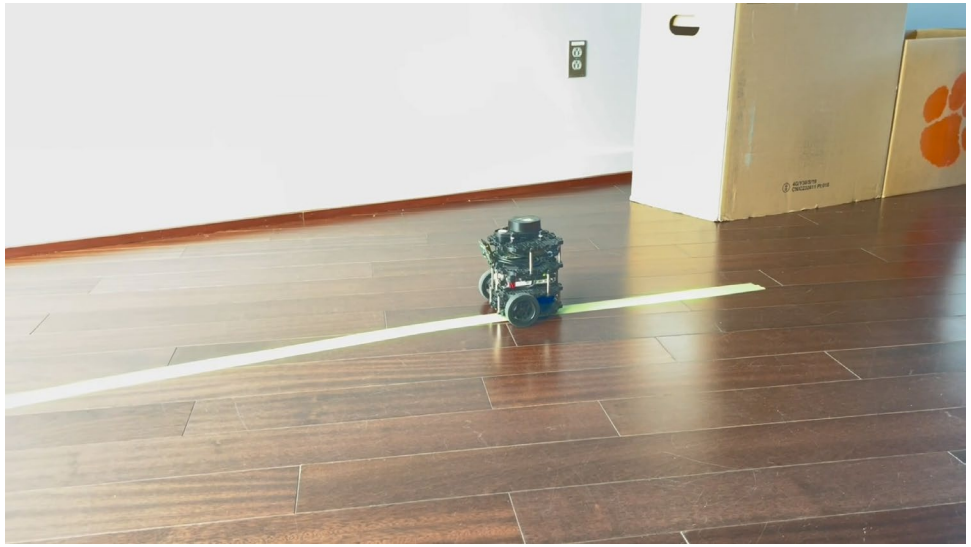


Remote PC

Task 3

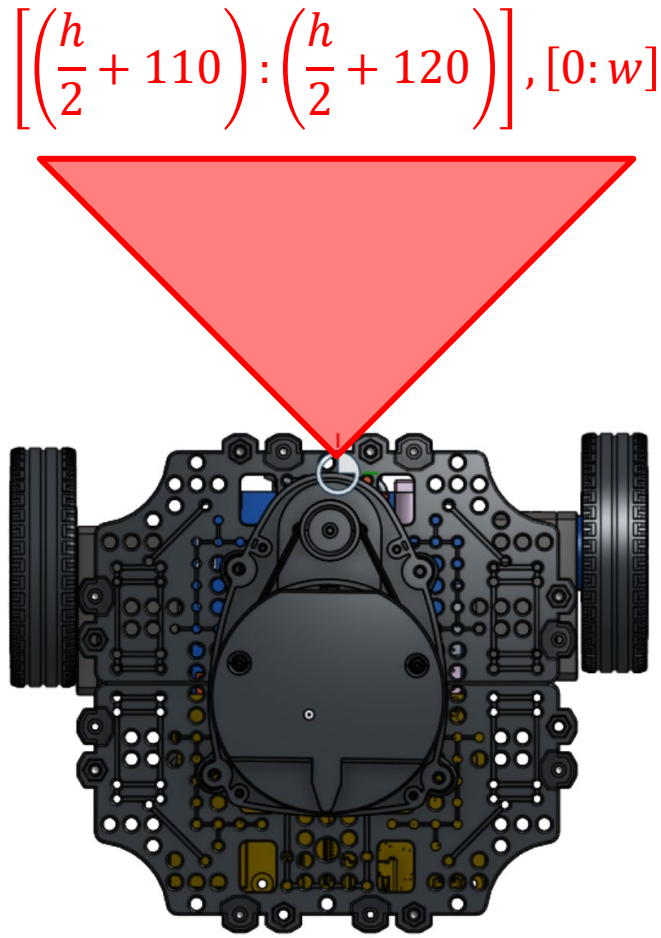
Line Following





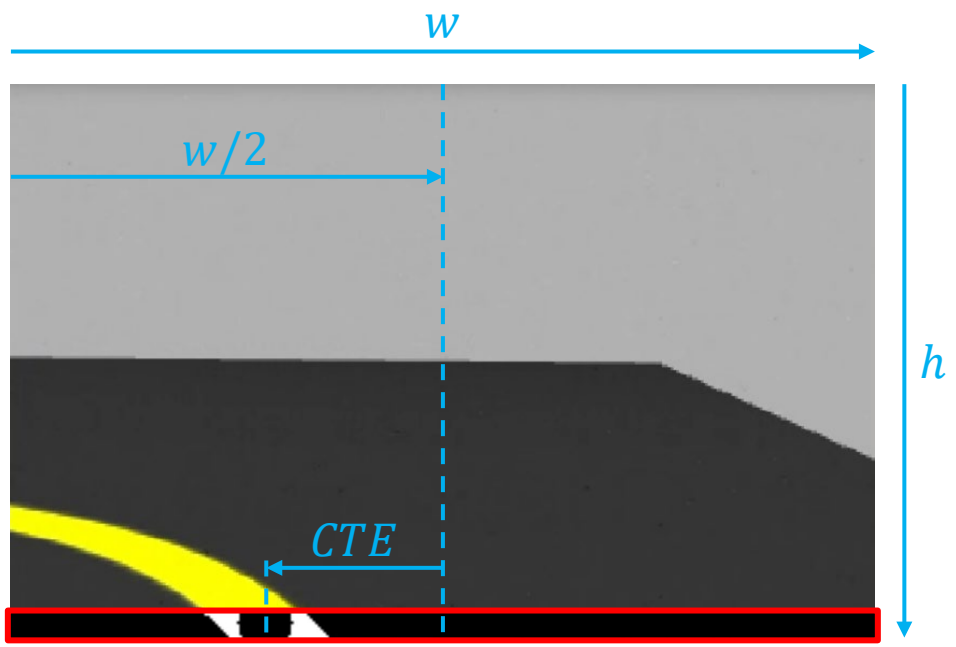
Approach

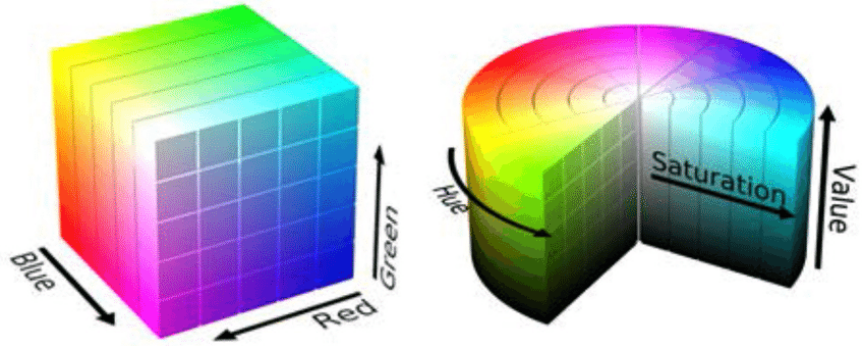
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Crop incoming image: $ROI = \left[\left(\frac{h}{2} + 110\right) : \left(\frac{h}{2} + 120\right)\right], [0 : w]$
  - Convert from RGB to HSV color space and mask specific color
  - Calculate weighted average of pixel intensities (detect clusters)
  - Calculate centroid of cluster and thereby deviation from line
  - Constant `lin_vel` with PID controller for `ang_vel`
  - Safety mechanisms and bounds (actuation limits)
- **Alternate approaches:**
  - Different ROI and HSV thresholds
  - Tilting camera – tight curves



Approach

- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Crop incoming image: $ROI = \left[\left(\frac{h}{2} + 110\right) : \left(\frac{h}{2} + 120\right)\right], [0 : w]$
  - Convert from RGB to HSV color space and mask specific color
  - Calculate weighted average of pixel intensities (detect clusters)
  - Calculate centroid of cluster and thereby deviation from line
  - Constant `lin_vel` with PID controller for `ang_vel`
  - Safety mechanisms and bounds (actuation limits)
- **Alternate approaches:**
  - Different ROI and HSV thresholds
  - Tilting camera – tight curves

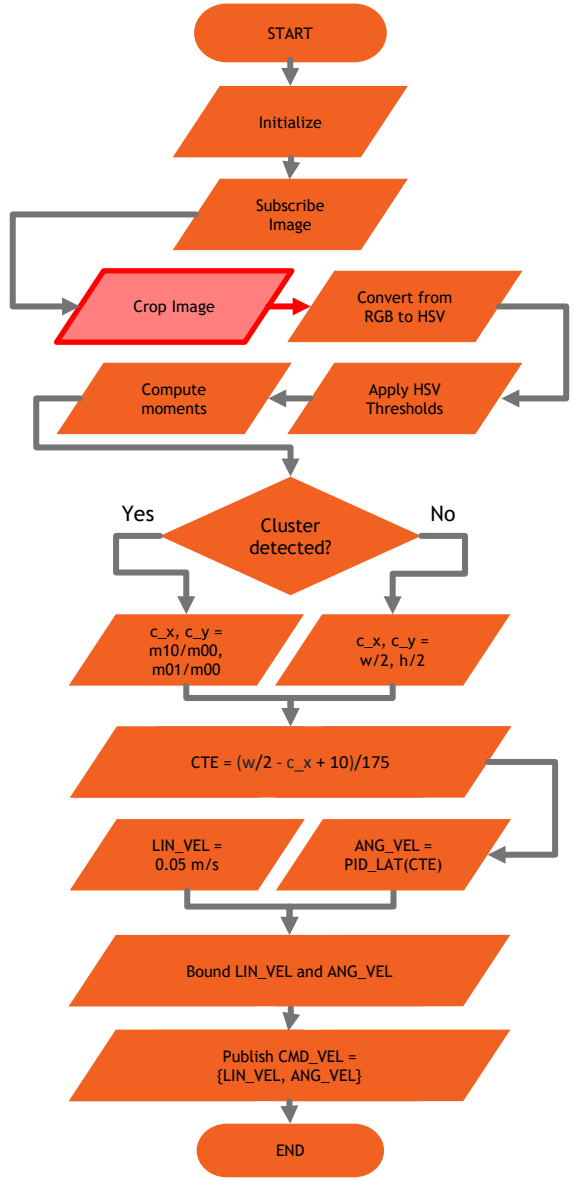




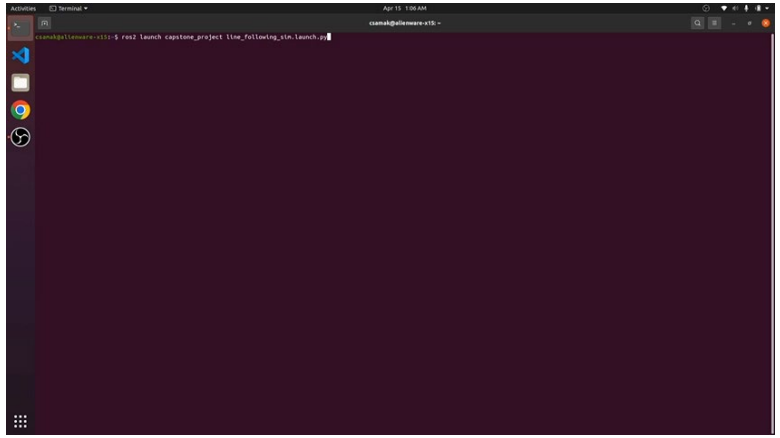
Source: V. Popov, et al.

Approach

- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Crop incoming image: $ROI = \left[\left(\frac{h}{2} + 110\right) : \left(\frac{h}{2} + 120\right)\right], [0 : w]$
  - Convert from RGB to HSV color space and mask specific color
  - Calculate weighted average of pixel intensities (detect clusters)
  - Calculate centroid of cluster and thereby deviation from line
  - Constant `lin_vel` with PID controller for `ang_vel`
  - Safety mechanisms and bounds (actuation limits)
- **Alternate approaches:**
  - Different ROI and HSV thresholds
  - Tilting camera – tight curves

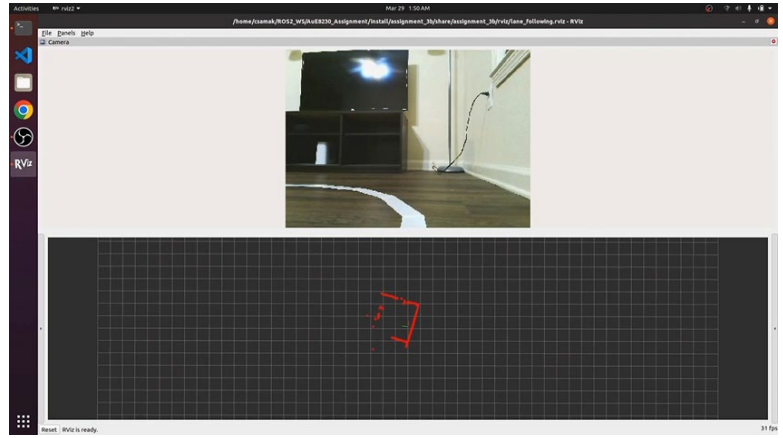


Results



Simulation

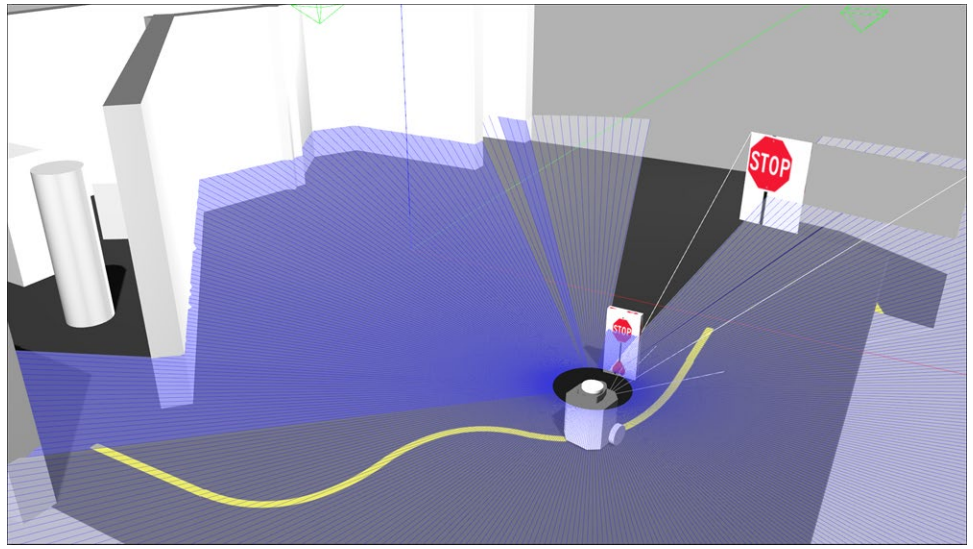
TurtleBot3

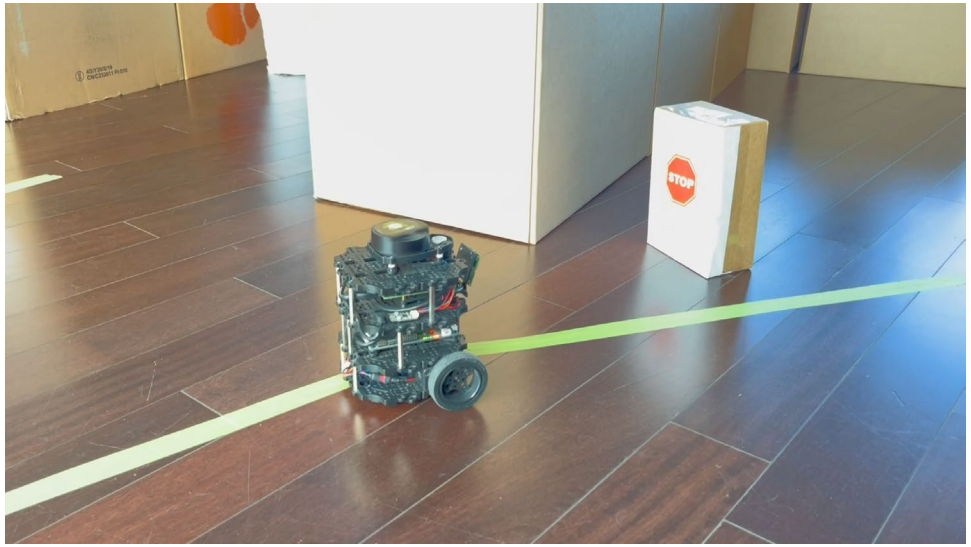


Remote PC

Task 4

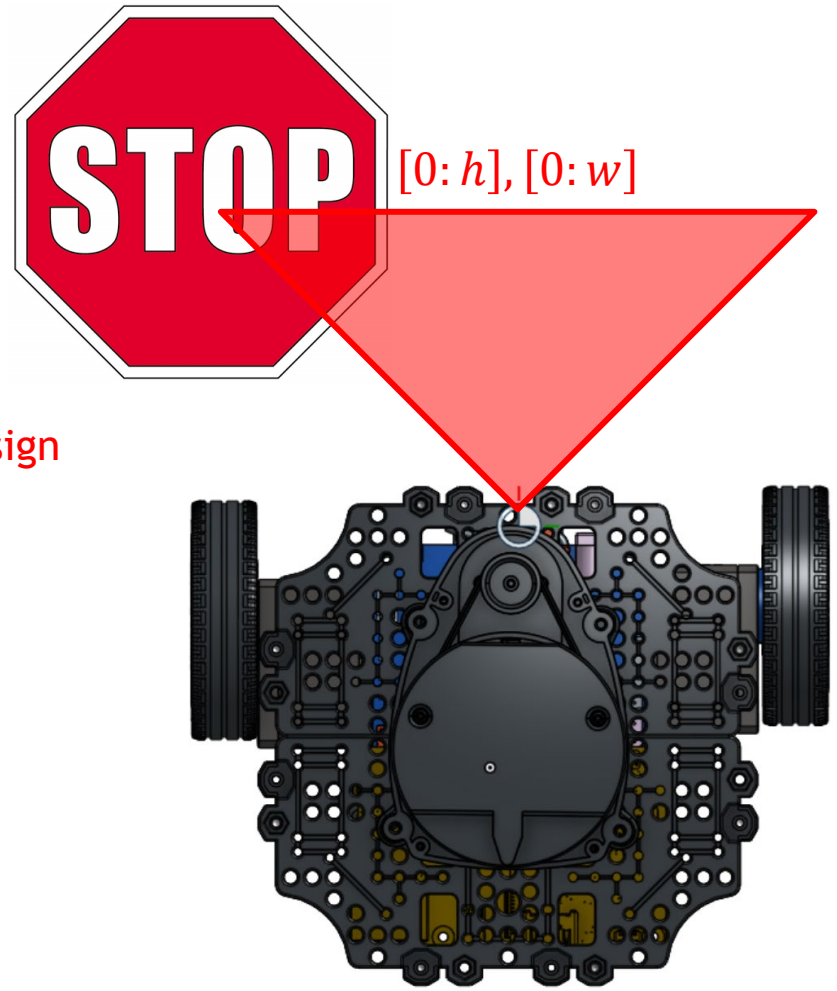
Stop Sign Detection





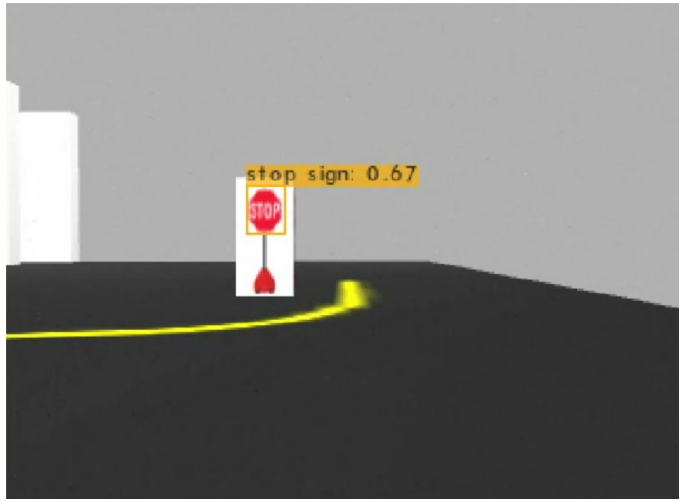
Approach

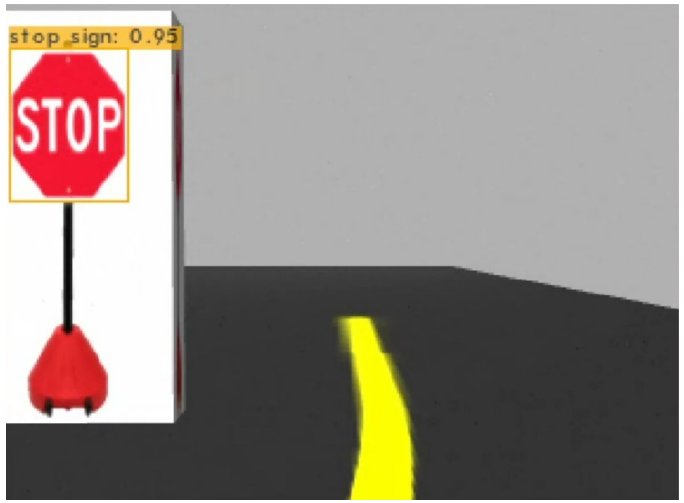
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run Tiny-YOLO V7 inference on incoming image
  - Verify detection class, confidence & bounding box area of stop sign
  - Come to a complete stop for predefined time (4 seconds)
  - Constant `lin_vel` $\begin{cases} v; & \text{stop sign absent} \\ 0; & \text{stop sign present} \end{cases}$ and 0 `ang_vel`
  - Safety mechanisms and bounds (actuation limits)
- **Alternate approaches:**
  - Different versions of YOLO
  - Different detection class(es), confidence & bounding box area
  - **CV vs. DL:** shape/color thresholding



Approach

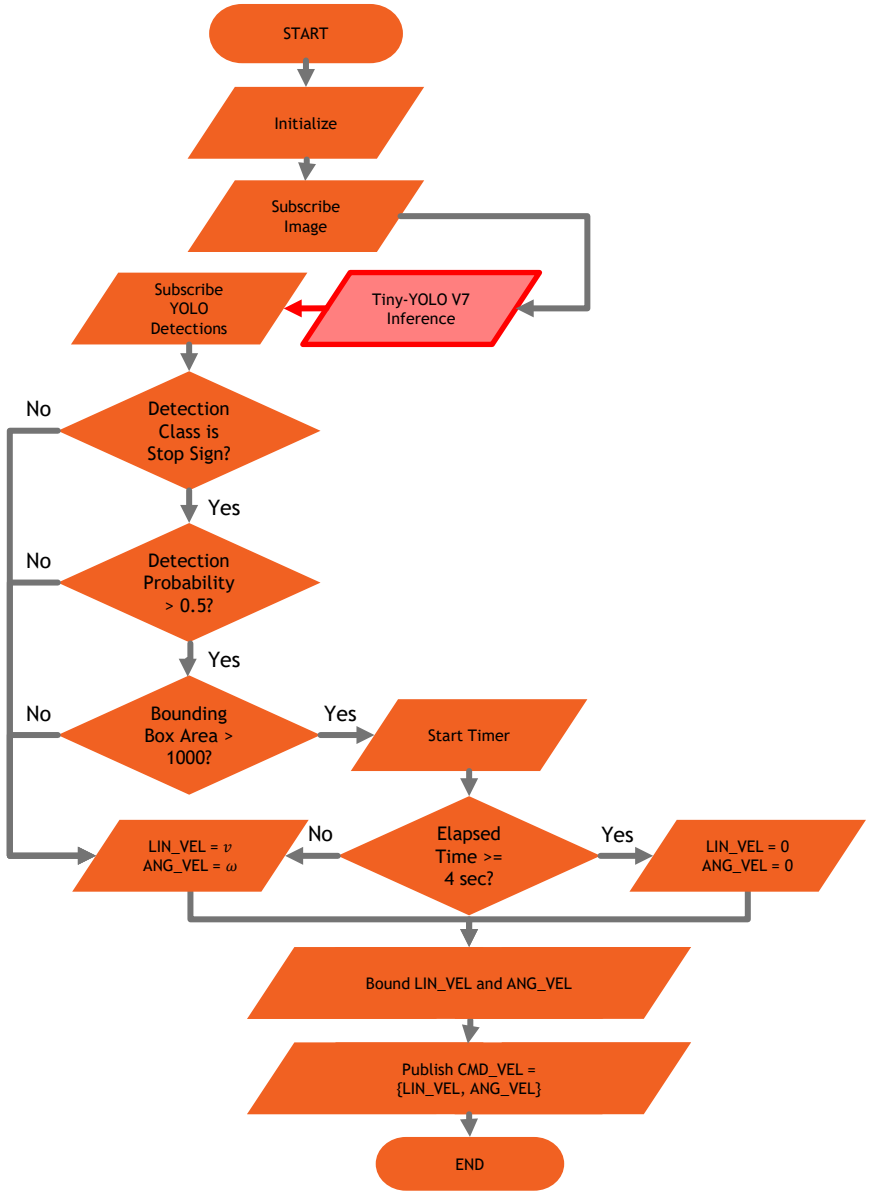
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run Tiny-YOLO V7 inference on incoming image
  - Verify detection class, confidence & bounding box area of stop sign
  - Come to a complete stop for predefined time (4 seconds)
  - Constant `lin_vel` $\begin{cases} v; & \text{stop sign absent} \\ 0; & \text{stop sign present} \end{cases}$ and 0 `ang_vel`
  - Safety mechanisms and bounds (actuation limits)
- **Alternate approaches:**
  - Different versions of YOLO
  - Different detection class(es), confidence & bounding box area
  - **CV vs. DL:** shape/color thresholding



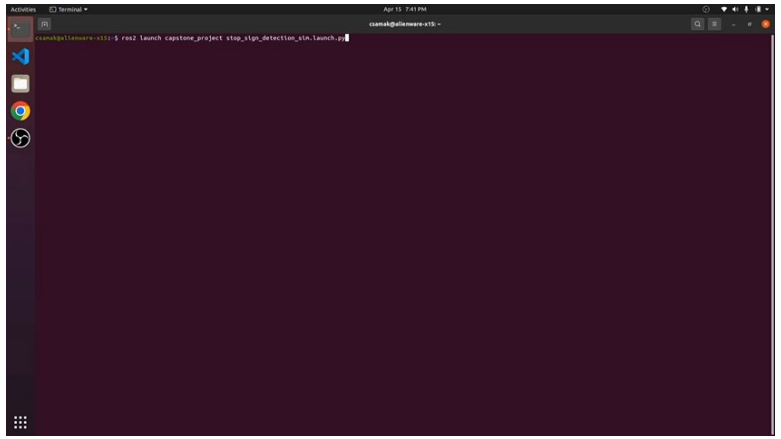


Approach

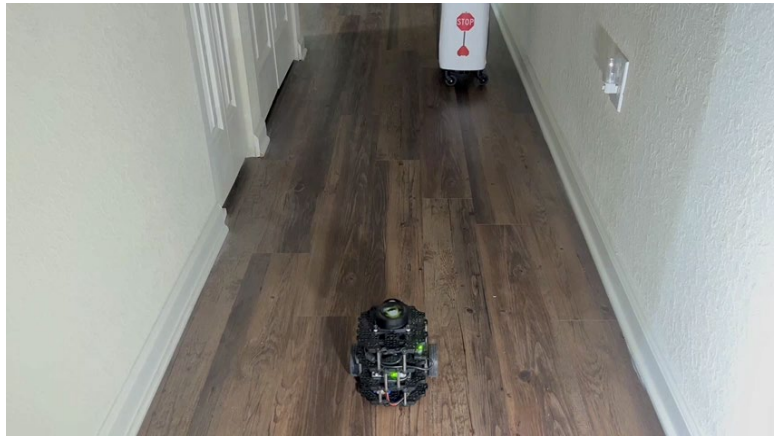
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run Tiny-YOLO V7 inference on incoming image
  - Verify detection class, confidence & bounding box area of stop sign
  - Come to a complete stop for predefined time (4 seconds)
  - Constant `lin_vel` $\begin{cases} v; & \text{stop sign absent} \\ 0; & \text{stop sign present} \end{cases}$ and 0 `ang_vel`
  - Safety mechanisms and bounds (actuation limits)
- **Alternate approaches:**
  - Different versions of YOLO
  - Different detection class(es), confidence & bounding box area
  - **CV vs. DL:** shape/color thresholding



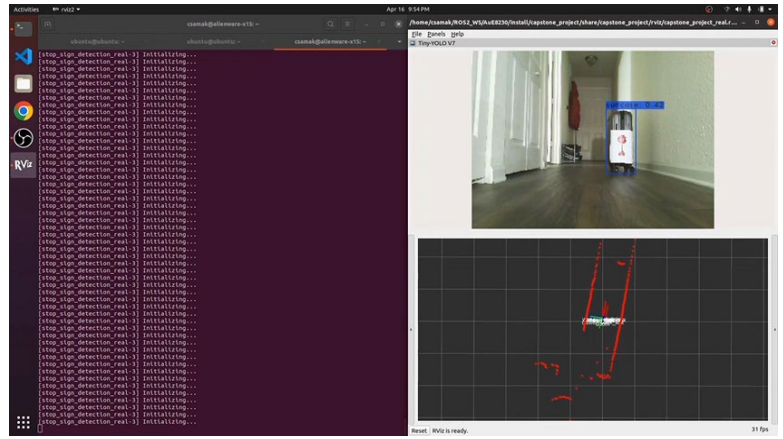
Results



Simulation



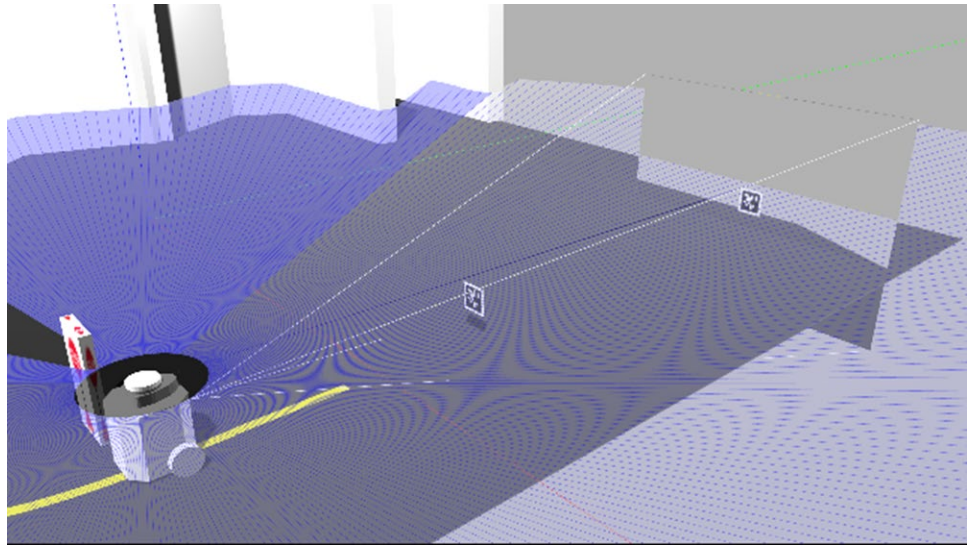
TurtleBot3

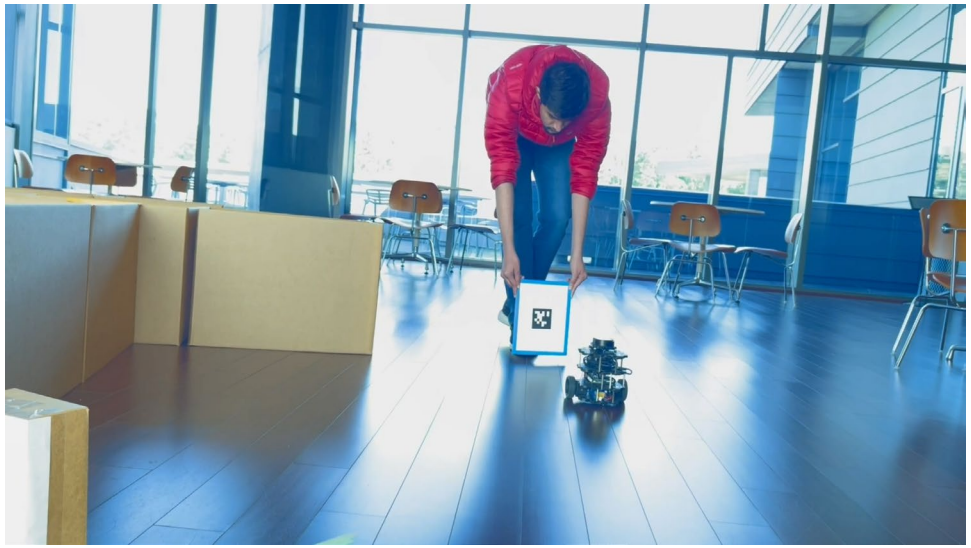


Remote PC

Task 5

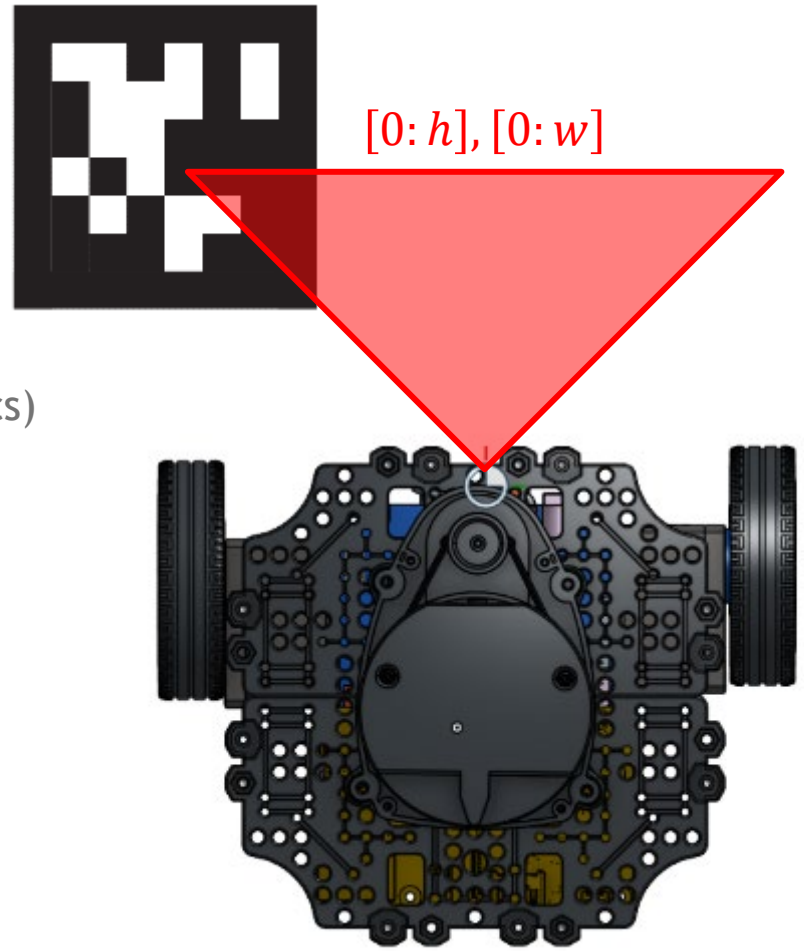
AprilTag Tracking





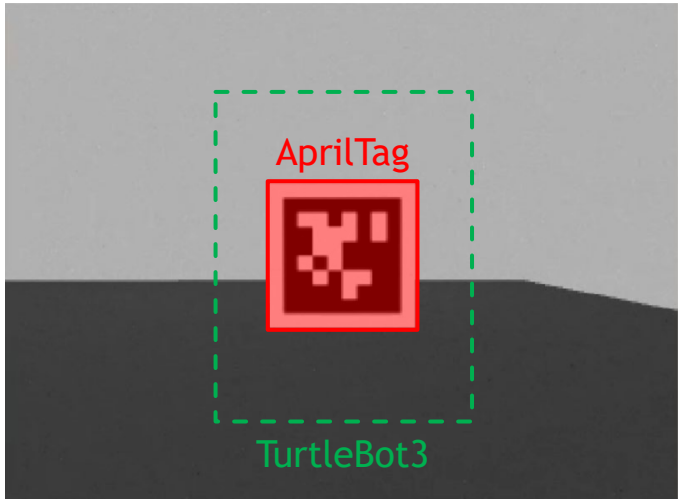
Approach

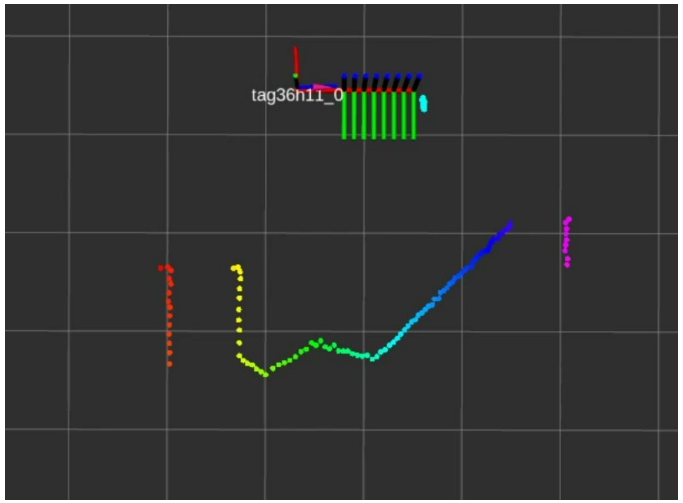
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run AprilTag detection on incoming image
  - Estimate 6D metric pose of marker w.r.t. robot's camera (intrinsics)
  - Apply static transform from `base_link` to `camera`
  - Operate on lateral (-x) & longitudinal (z) deviation from marker
  - Stop if too close
  - De-coupled lat-lon PID controller architecture
  - Safety mechanisms and bounds (actuation limits)
- **Alternate approaches:**
  - Print quality
  - Tag size



Approach

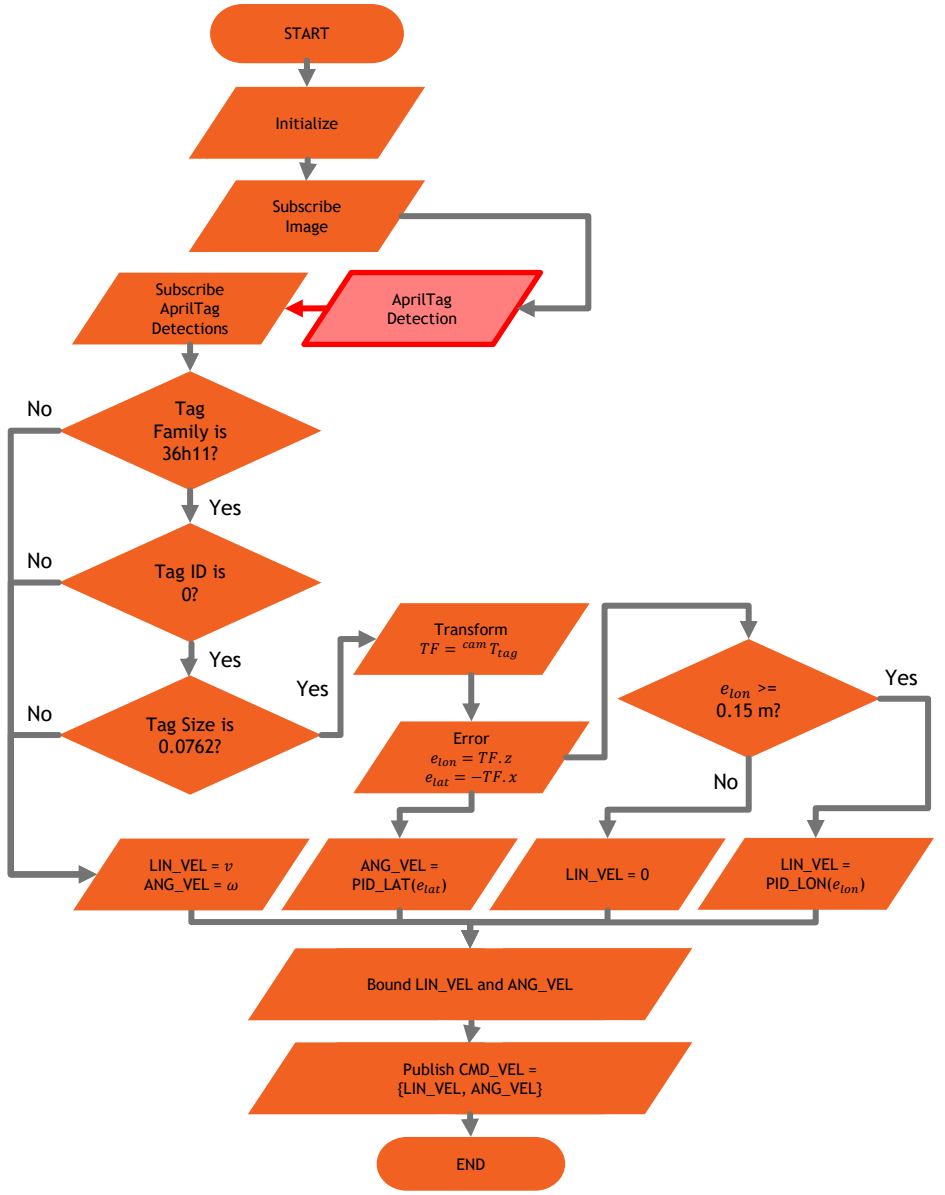
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run AprilTag detection on incoming image
  - Estimate 6D metric pose of marker w.r.t. robot's camera (intrinsics)
  - Apply static transform from `base_link` to `camera`
  - Operate on lateral (-x) & longitudinal (z) deviation from marker
  - Stop if too close
  - De-coupled lat-lon PID controller architecture
  - Safety mechanisms and bounds (actuation limits)
- **Alternate approaches:**
  - Print quality
  - Tag size



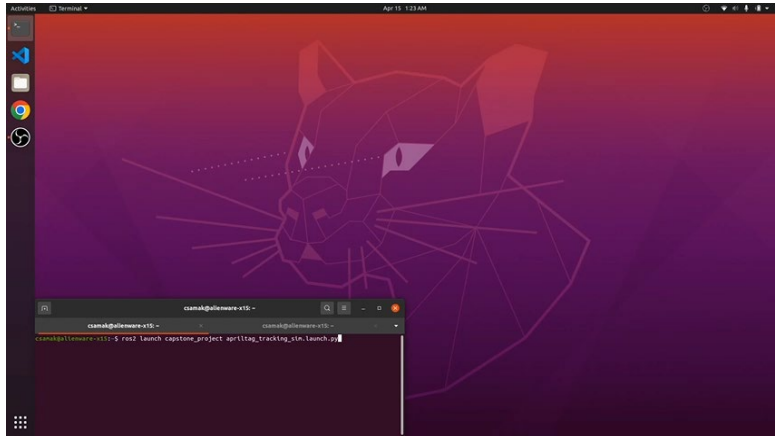


Approach

- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run AprilTag detection on incoming image
  - Estimate 6D metric pose of marker w.r.t. robot's camera (intrinsics)
  - Apply static transform from `base_link` to `camera`
  - Operate on lateral (-x) & longitudinal (z) deviation from marker
  - Stop if too close
  - De-coupled lat-lon PID controller architecture
  - Safety mechanisms and bounds (actuation limits)
- **Alternate approaches:**
  - Print quality
  - Tag size

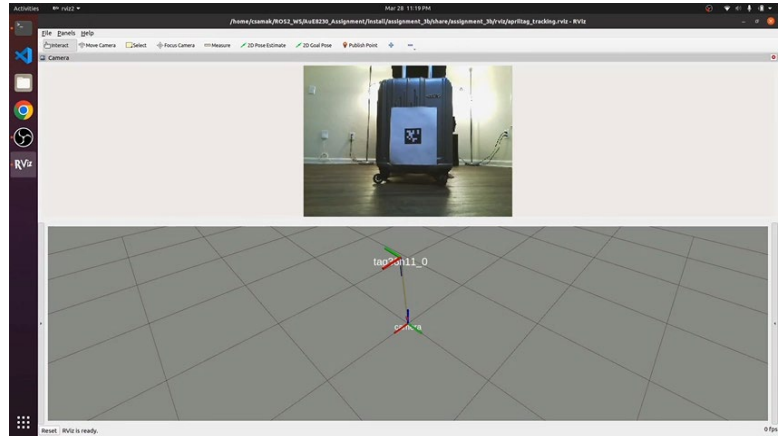


Results



Simulation

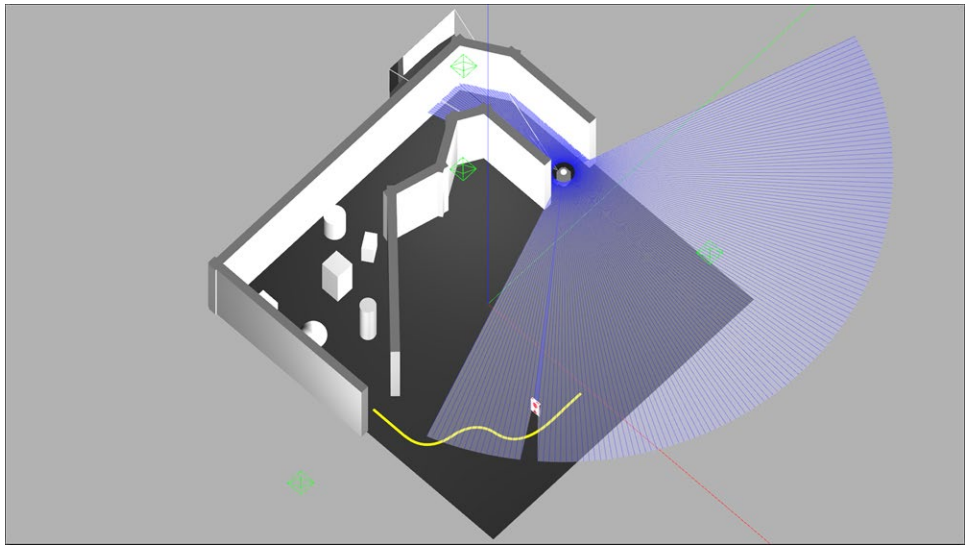
TurtleBot3



Remote PC

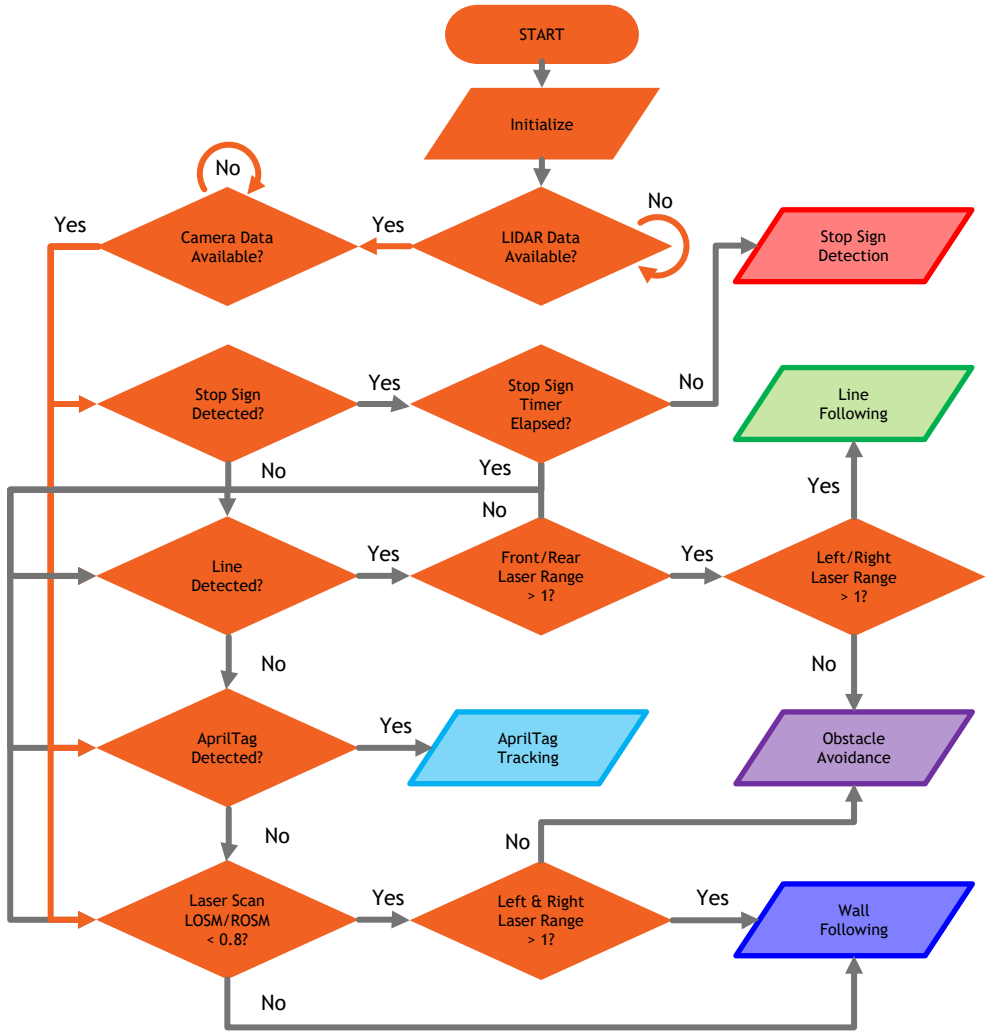
Integration

Finite State Automaton



Approach

- **Sensors:** Camera + LIDAR
- **Priority:**
  - Priority 1: Stop Sign Detection
  - Priority 2: Line Following
  - Priority 3: AprilTag Tracking
  - Priority 4: Obstacle Avoidance
  - Priority 5: Wall Following
- **Alternate approaches:**
  - Keyboard switching
  - Fiducial markers



Approach

- Initialize and check if LIDAR and camera data is available
- **Priority 1:** Stop Sign Detection

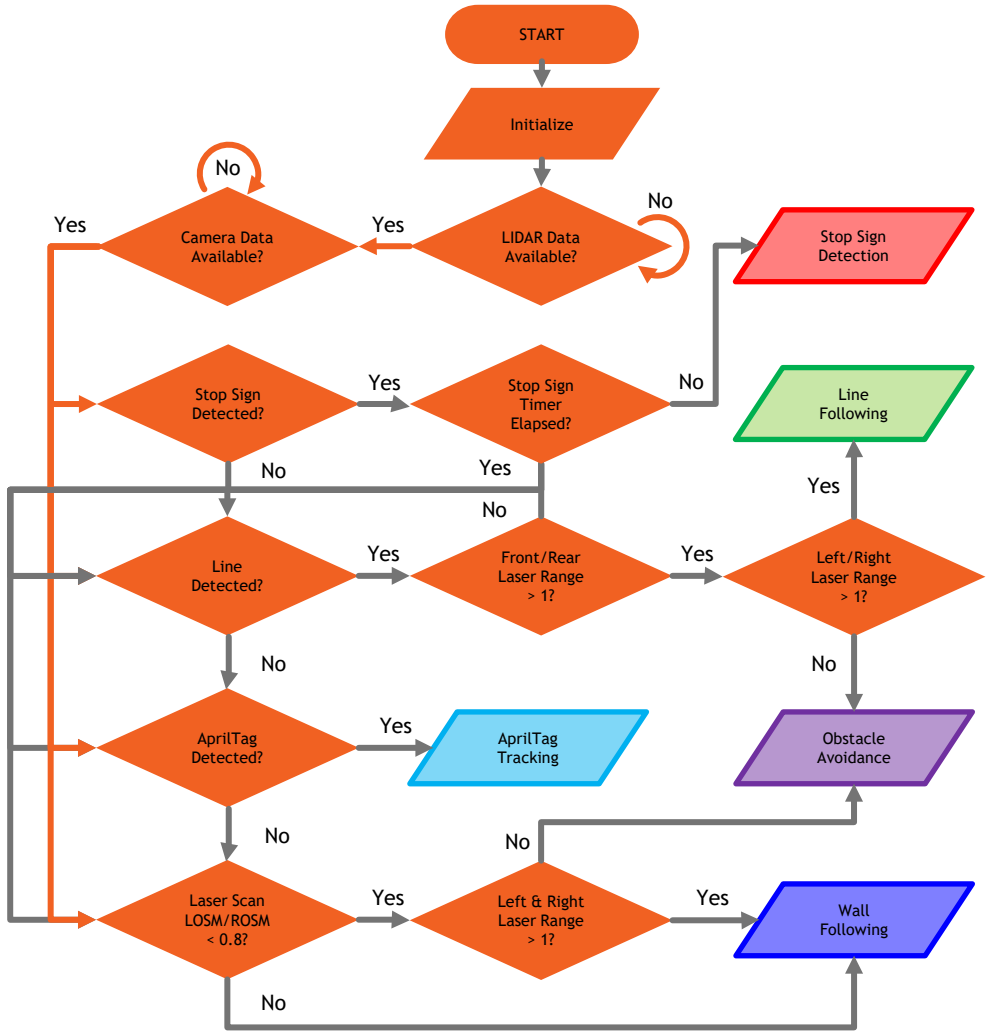
  True if stop sign detected and timer not elapsed
- **Priority 2:** Line Following

  True if line detected, not stopped at stop sign and front/rear and left/right laser range > 1 m
- **Priority 3:** AprilTag Tracking

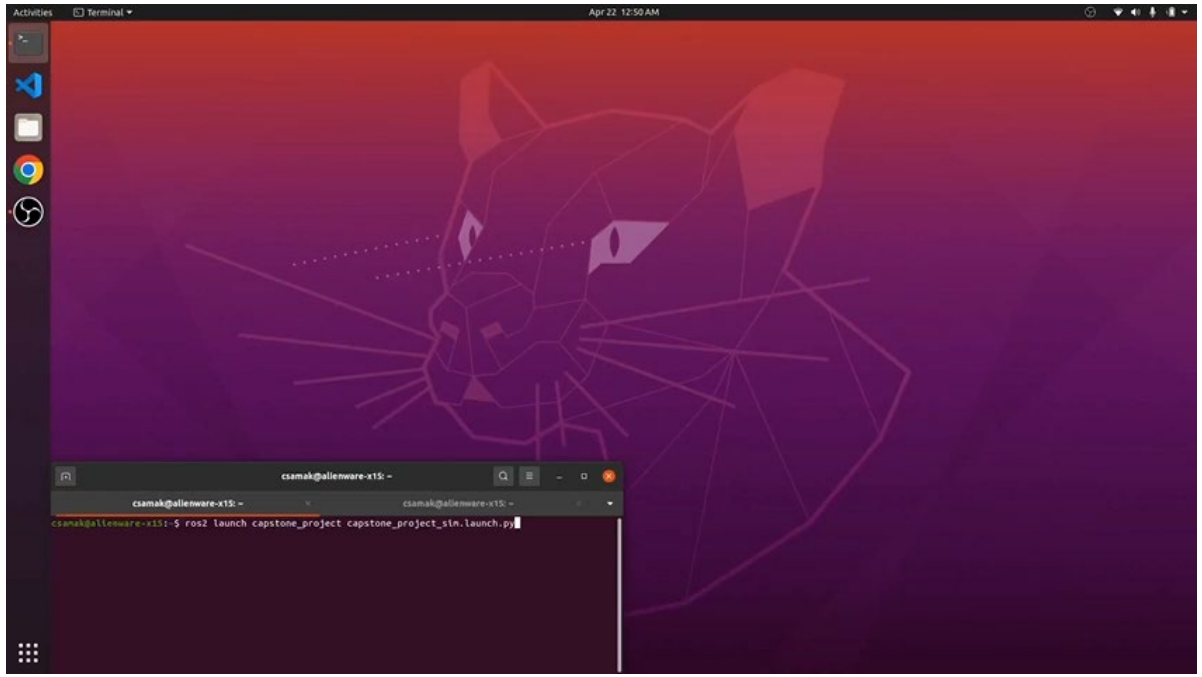
  True if tag detected, line not detected and not stopped at stop sign
- **Priority 4:** Obstacle Avoidance

  True if tag not detected, line not detected, not stopped at stop sign and laser scan LOSM or ROSM < 0.8
- **Priority 5:** Wall Following

  True if tag not detected, line not detected, not stopped at stop sign and laser scan LSOM or ROSM >= 0.8 or left and right laser range > 1 m



Results

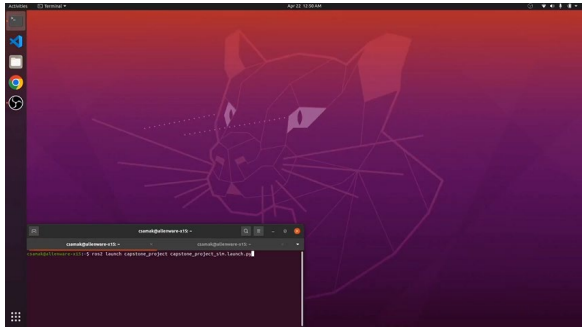


Simulation

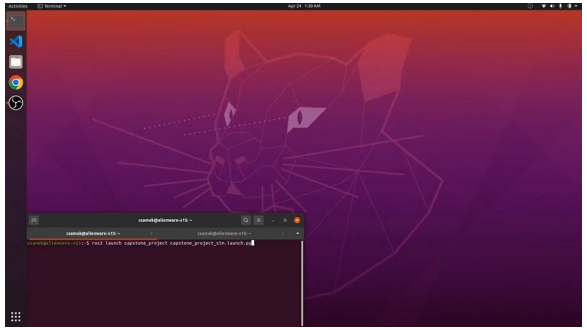
Reality

Robustness Testing

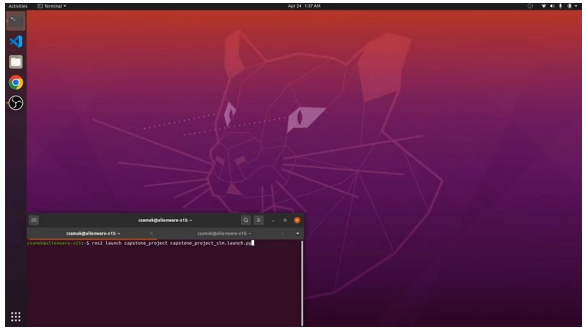
Simulation



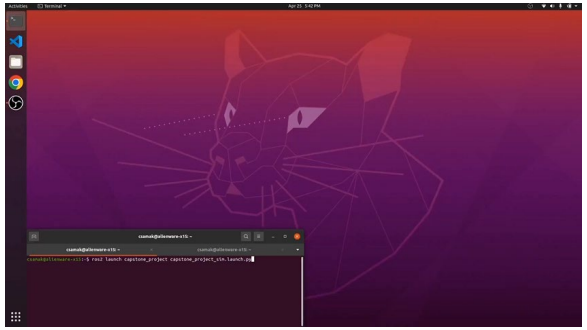
Forward Direction;
Start with Wall Following



Forward Direction;
Start with Line Following



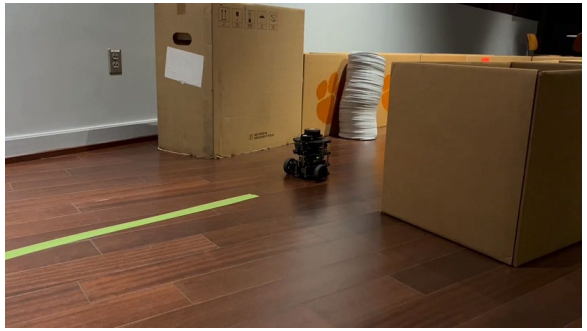
Forward Direction;
Start with AprilTag Tracking



Inverse Direction;
Start with Line Following

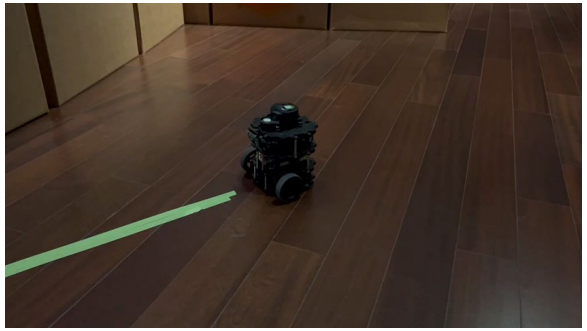
Reality

Day; Forward Direction;
Start at Wall Following



Night; Forward Direction;
Start with Line Following

Night; Forward Direction;
Start with AprilTag Tracking



Night; Inverse Direction;
Start with Line Following

Challenges Faced

- ROS 2 DDS communication framework
  - Domain ID not robust to network traffic fluctuations
  - Topics/nodes not discoverable over network (Ethernet/WiFi)
  - Wireless overlap with limited available channels
  - Conflicting QoS profiles for sensors and actuators
  - Specific issues: Raspberry Pi + Remote PC + ROS 2 Foxy (reason unknown)
  - Open issues on community forums (ROS 2 Foxy EOL: May 2023)
  - Fast DDS to Cyclone DDS migration in future ROS 2 distros

Challenges Faced

- Unavailable ROS 2 drivers, packages and lack of resources
- ROS 2 TurtleBot3 URDF -- larger in size as compared to ROS 1 URDF and real-world robot
- Gazebo world and texture setup for ROS 2 -- fixed the “wobbly obstacle” bug
- Gazebo residual errors and crashes -- added troubleshooting tips to README.md
- Practical considerations for sim2real transition -- tips added to README.md
- Automatic mode switching leaves very less room for fail-safe mechanisms
- Avoid access to privileged information (e.g. map) or equipment (e.g. flashlights) -- personal choice

Thank you!

…open to questions and suggestions