AuE-8230
Autonomy: Science and Systems

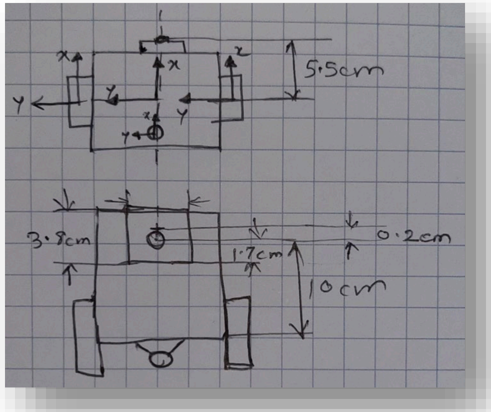
Capstone Project

Group 1: Chinmay Samak, Tanmay Samak

Codebase

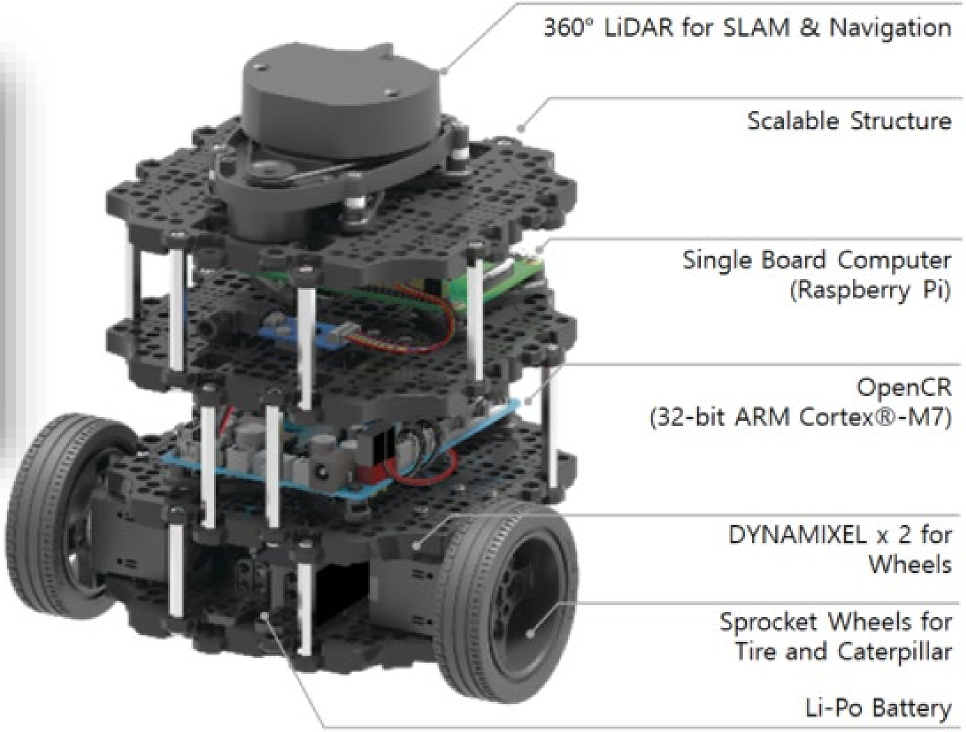
Project Tools

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



TURTLEBOT3 GAZEBO 2

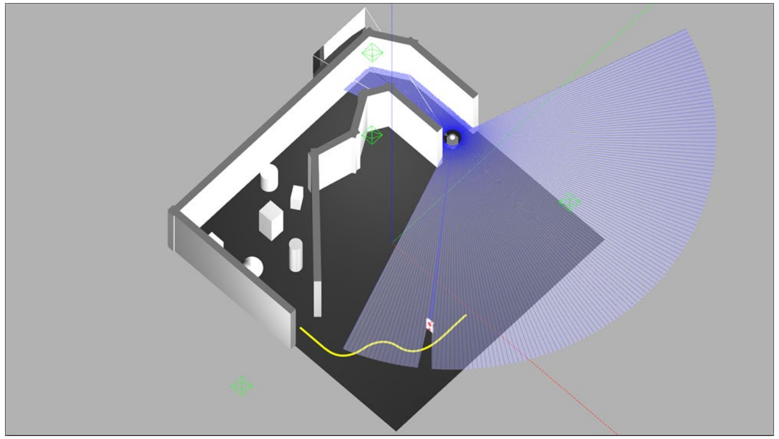
TurtleBot3 Burger


360° LiDAR for SLAM & Navigation
Scalable Structure
Single Board Computer (Raspberry Pi)
OpenCR (32-bit ARM Cortex®-M7)
DYNAMIXEL x 2 for Wheels
Sprocket Wheels for Tire and Caterpillar
Li-Po Battery

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

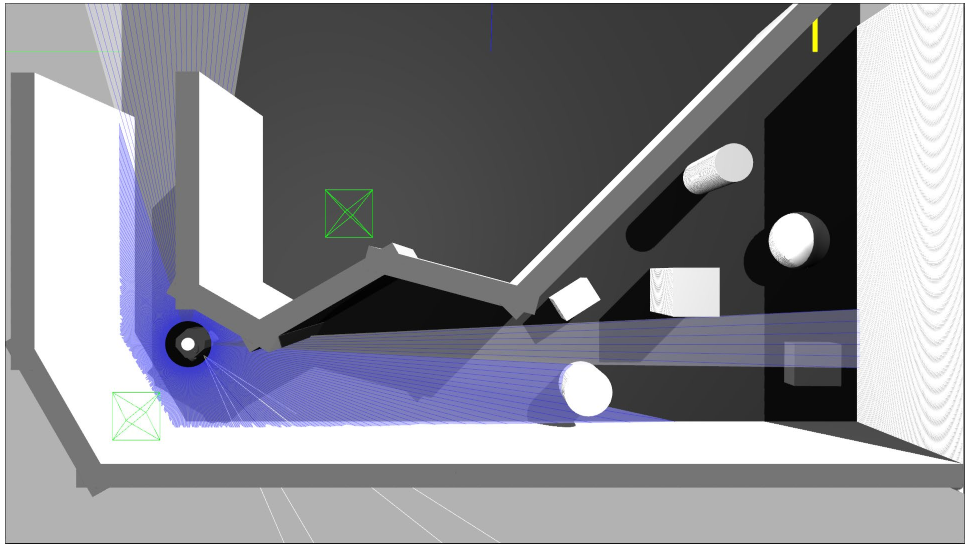
AuE-8230 Capstone Project (Spring 2023)											
Task	**03/28**	**03/30**	**04/04**	**04/06**	**04/11**	**04/14**	**04/18**	**04/21**	**04/25**	**04/28**	**05/04**
Announcements	⬛	⬛									
Task 1		🟧	🟧								
Task 2				🟩	🟩						
Task 3						🟥	🟥				
Task 4					🟦	🟦	🟦				
Task 5							🟨	🟨			
Demo Rehearsals								🟦	🟦		
Project Demo									🟨		
Project Presentation			🟦		🟦		🟦		🟦		
Project Report										🟪	🟪

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.*

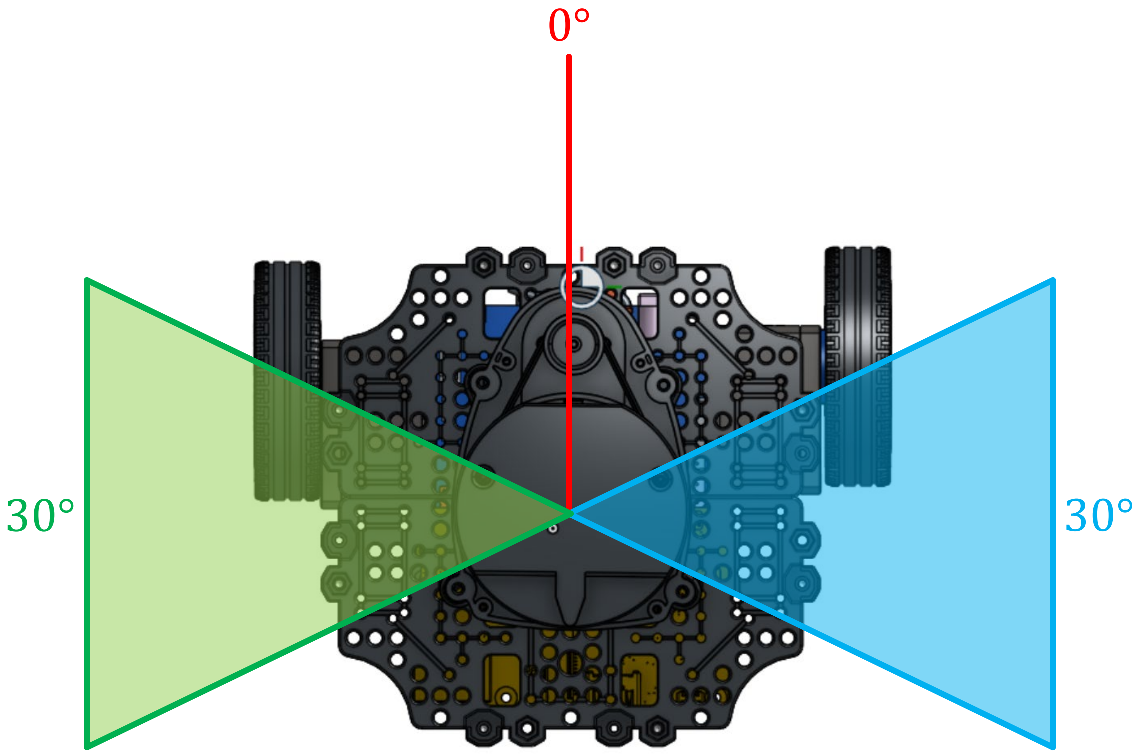


Task 1

Wall Following

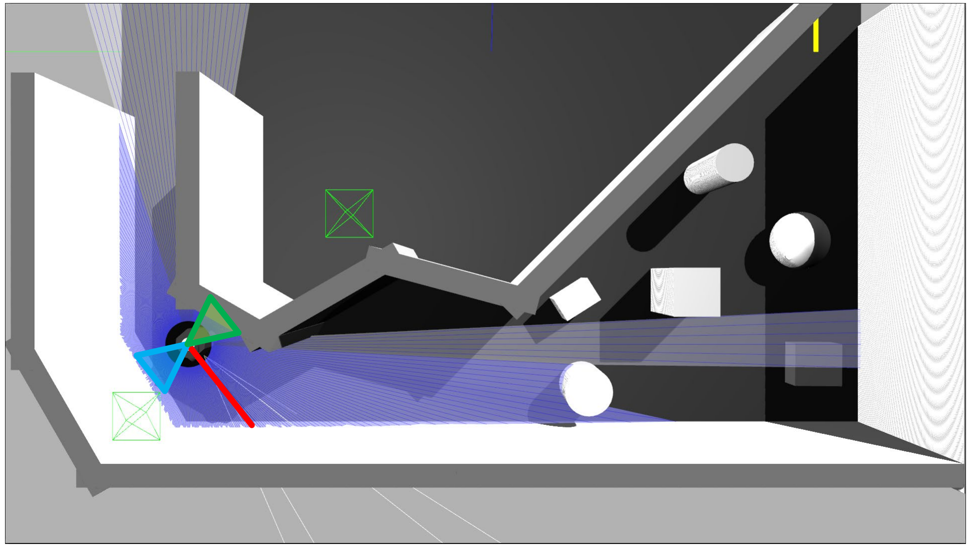
Approach

- **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



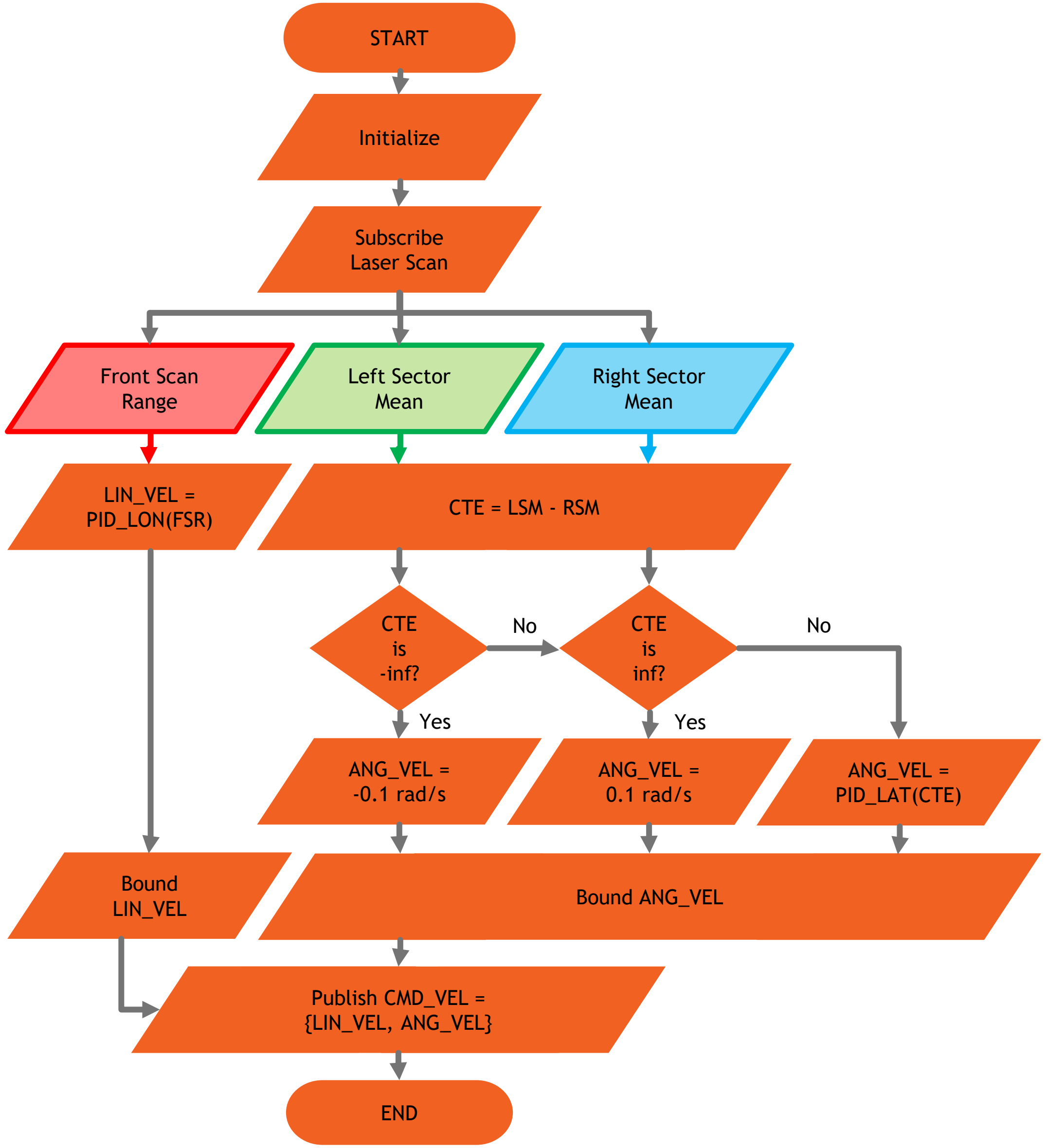
Approach

- **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

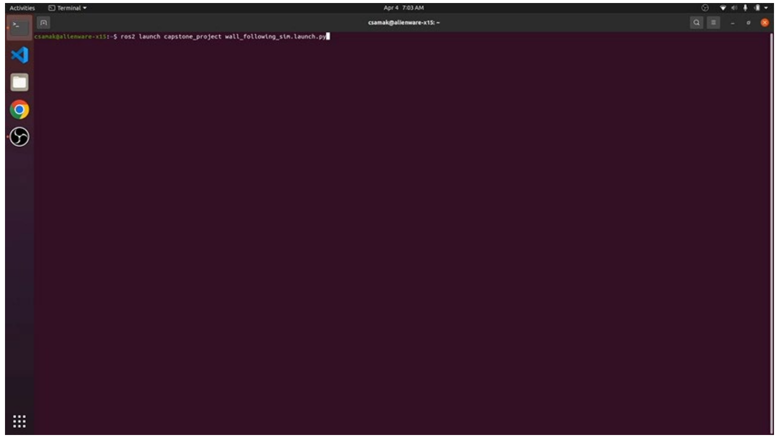


Approach

- **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

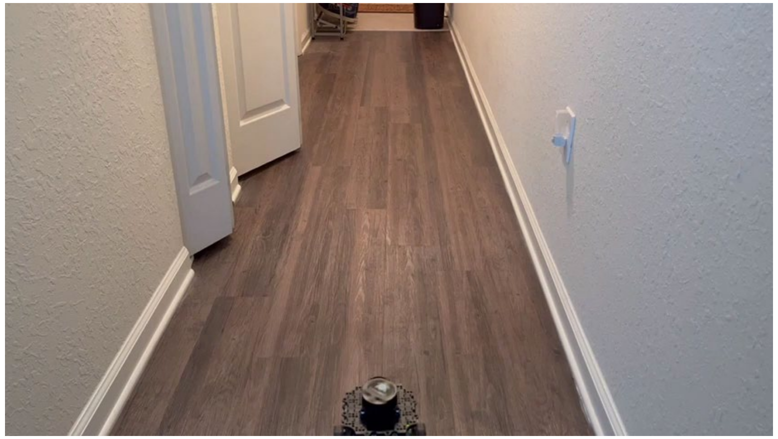


Results

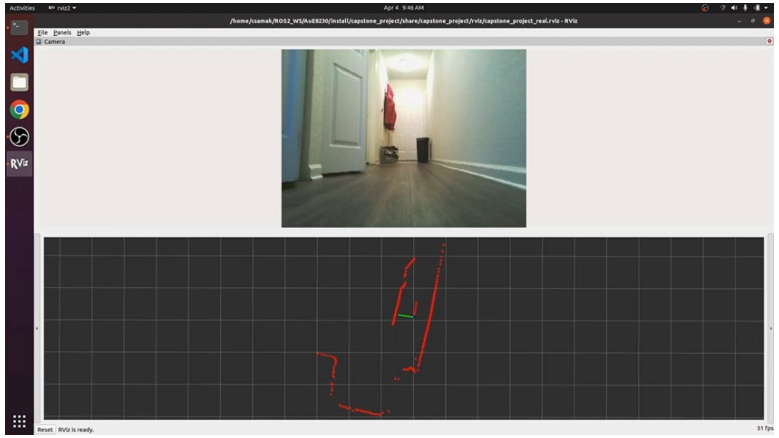


```
csamak@alienware-x15:~$ ros2 launch capstone_project wall_following_sim.launch.py
```

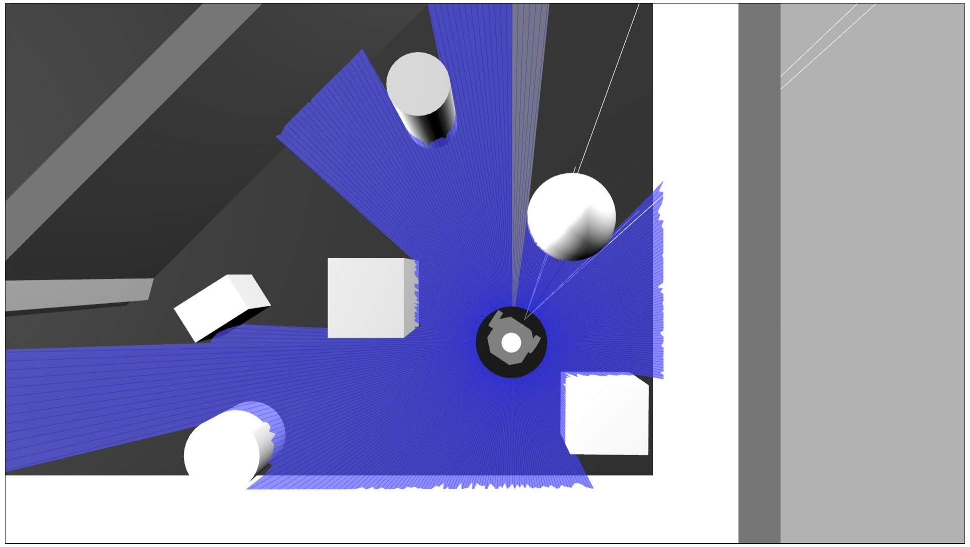
Simulation



TurtleBot3


Activities
rviz2
Apr 4 9:46 AM
/home/csamak/ROS2_WS/AuE8230/install/capstone_project/share/capstone_project/rviz/capstone_project_real.rviz - RViz
File
Panels
Help
Camera
Reset
RViz is ready.
31 fps

Remote PC

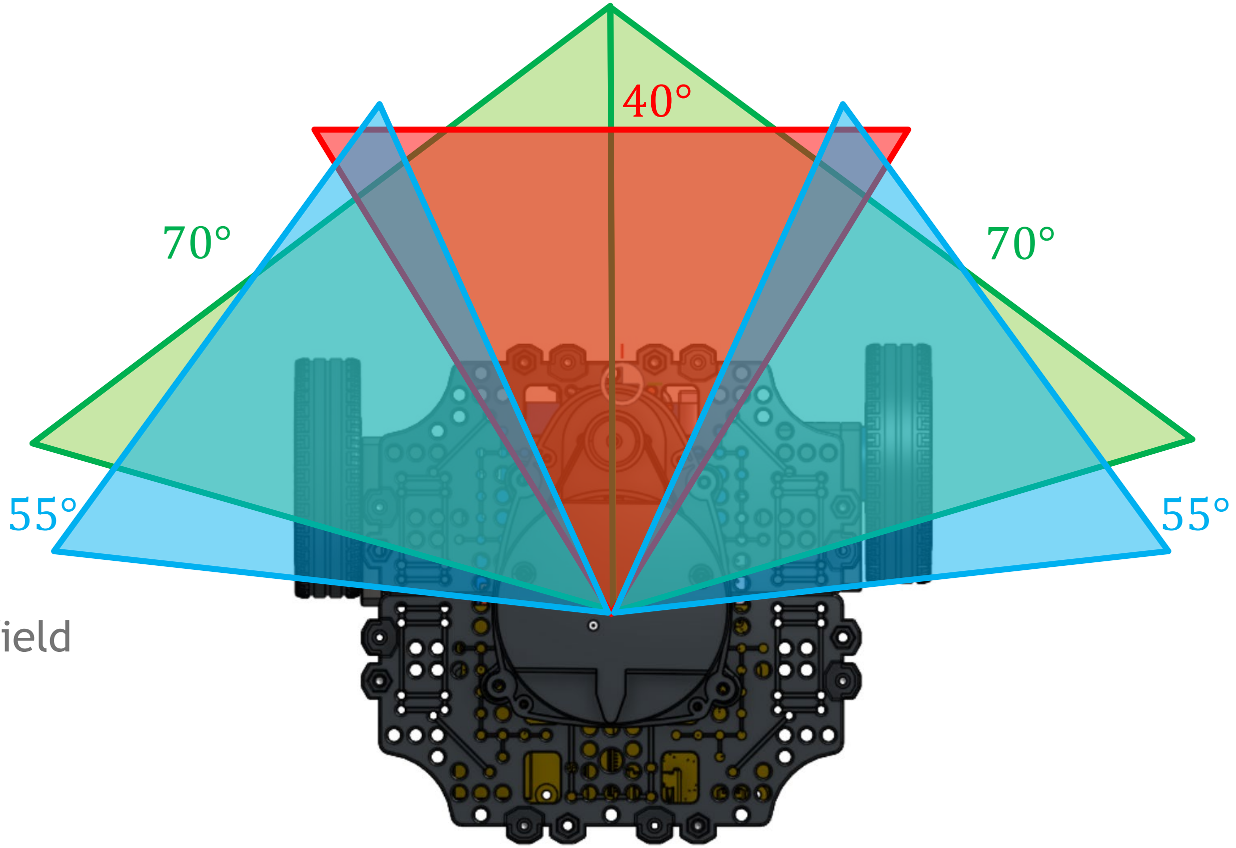


Task 2

Obstacle Avoidance

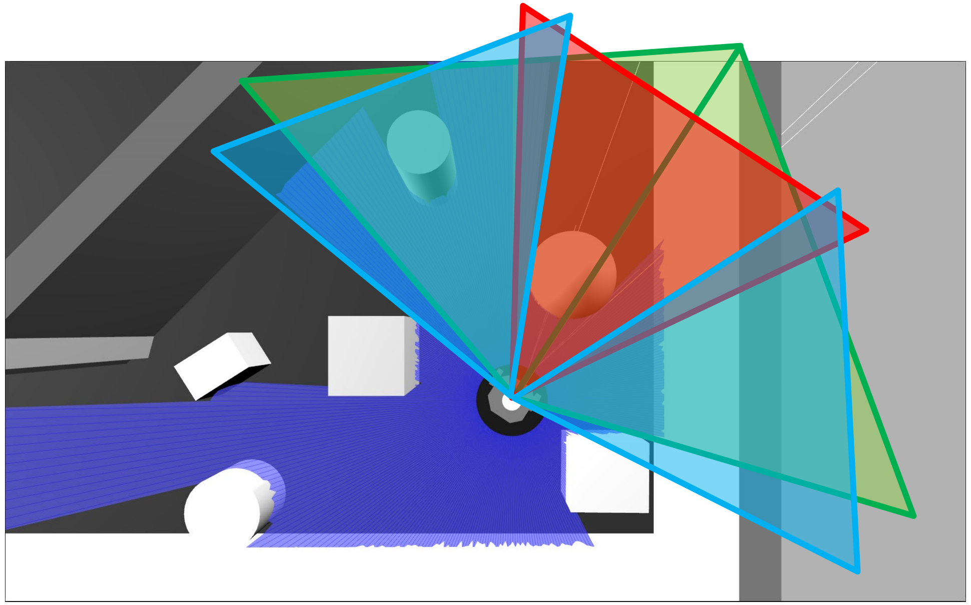
Approach

- **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



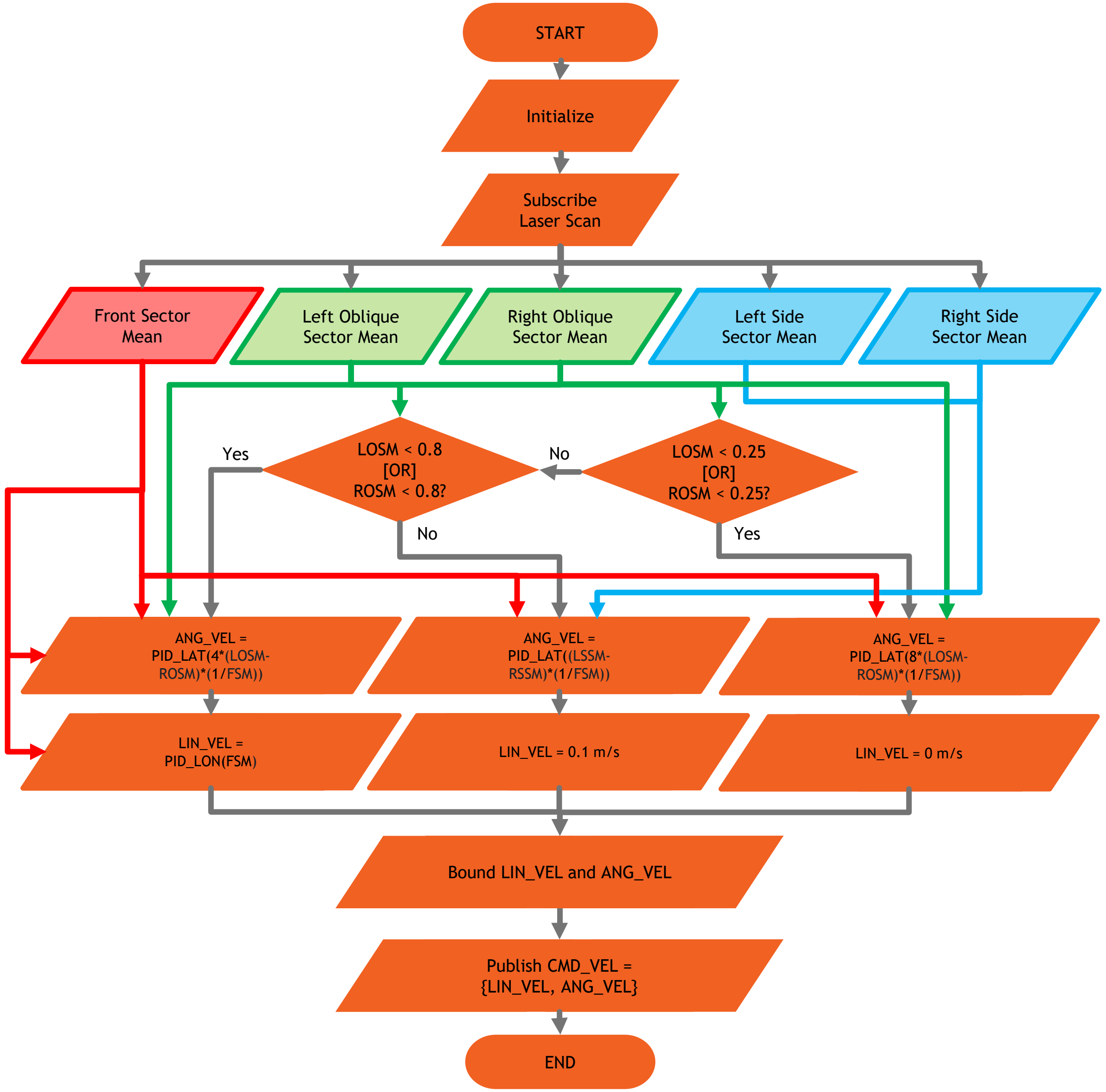
Approach

- **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

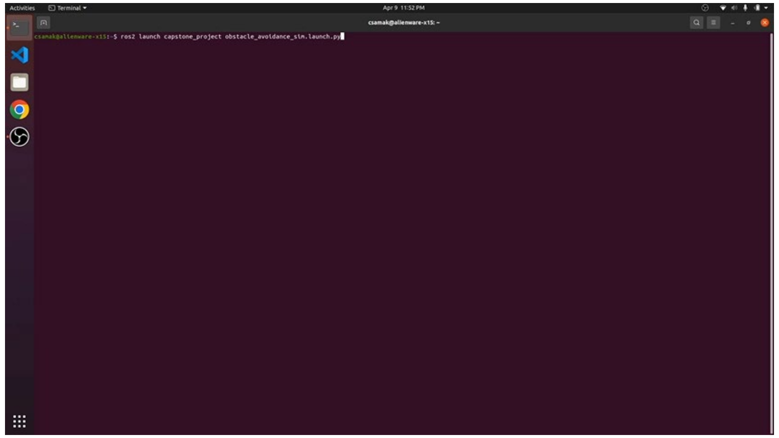


Approach

- **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

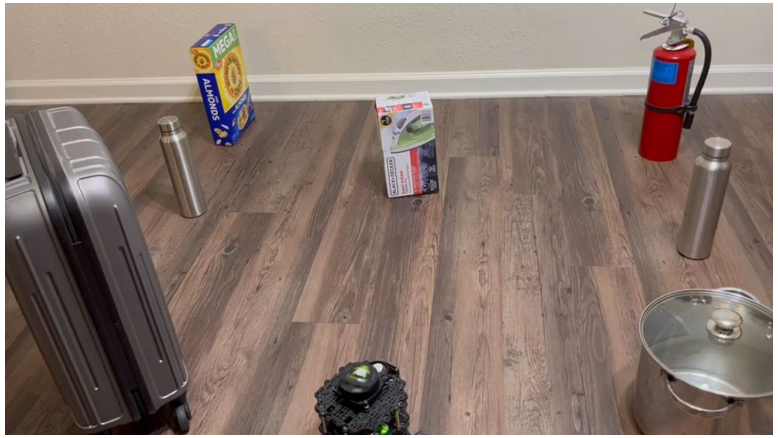

START
Initialize
Subscribe Laser Scan
Front Sector Mean
Left Oblique Sector Mean
Right Oblique Sector Mean
Left Side Sector Mean
Right Side Sector Mean
Yes
LOSM < 0.8 [OR] ROSM < 0.8?
No
LOSM < 0.25 [OR] ROSM < 0.25?
No
Yes
ANG_VEL = PID_LAT(4*(LOSM-ROSM)*(1/FSM))
ANG_VEL = PID_LAT((LSSM-RSSM)*(1/FSM))
ANG_VEL = PID_LAT(8*(LOSM-ROSM)*(1/FSM))
LIN_VEL = PID_LON(FSM)
LIN_VEL = 0.1 m/s
LIN_VEL = 0 m/s
Bound LIN_VEL and ANG_VEL
Publish CMD_VEL = {LIN_VEL, ANG_VEL}
END

Results

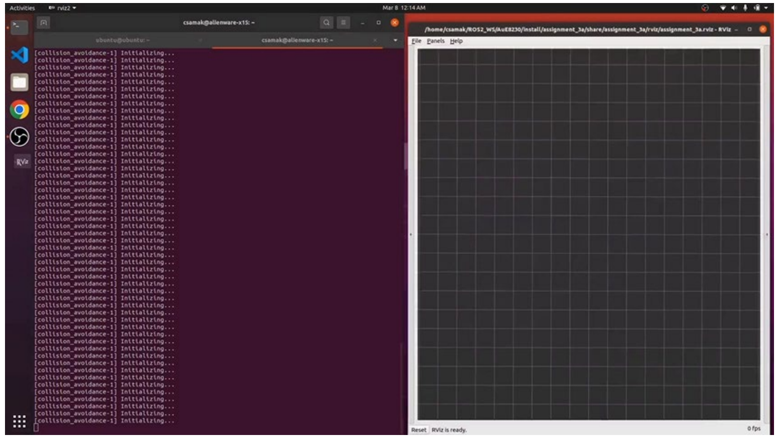


```
csamak@alienware-x15:~$ ros2 launch capstone_project obstacle_avoidance_sim.launch.py
```

Simulation



TurtleBot3

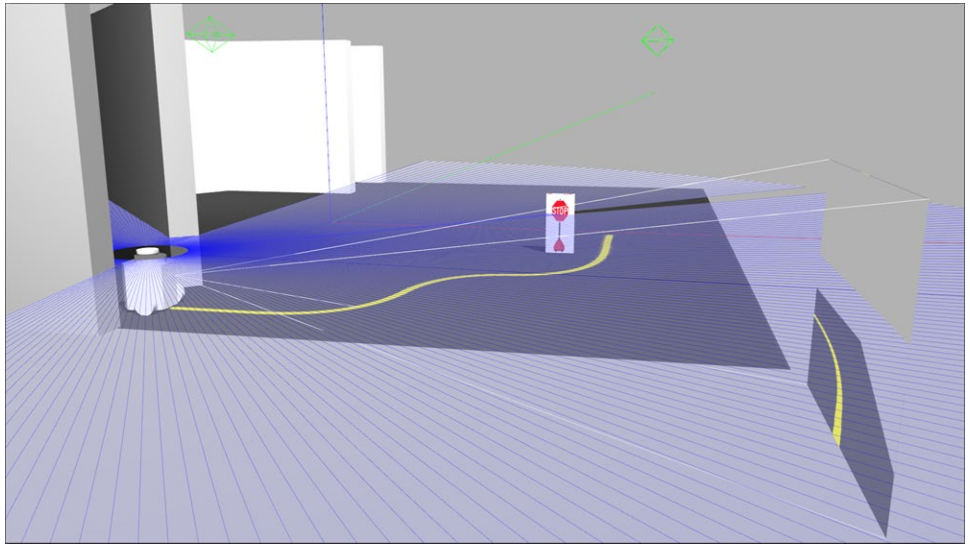


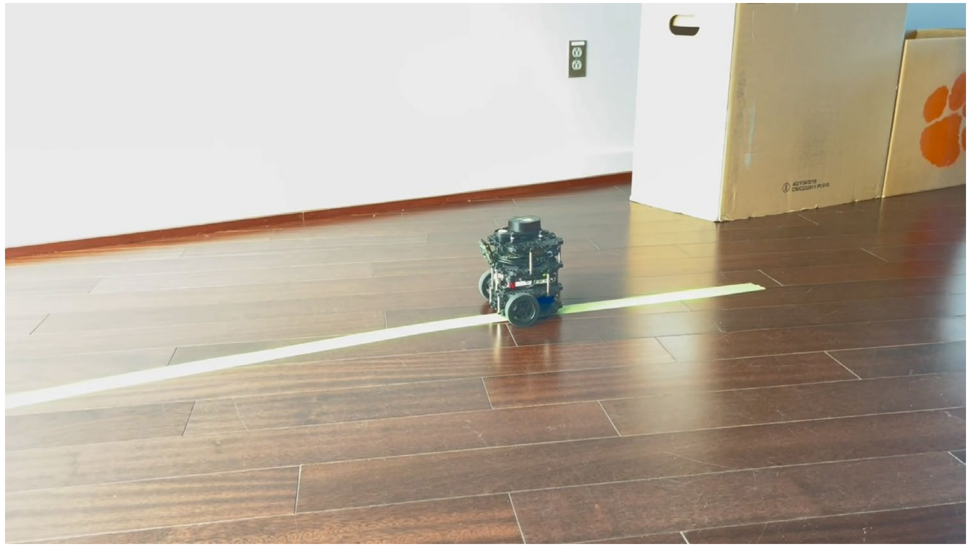
```
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
[collision_avoidance-1] Initializing...
```

Remote PC

Task 3

Line Following

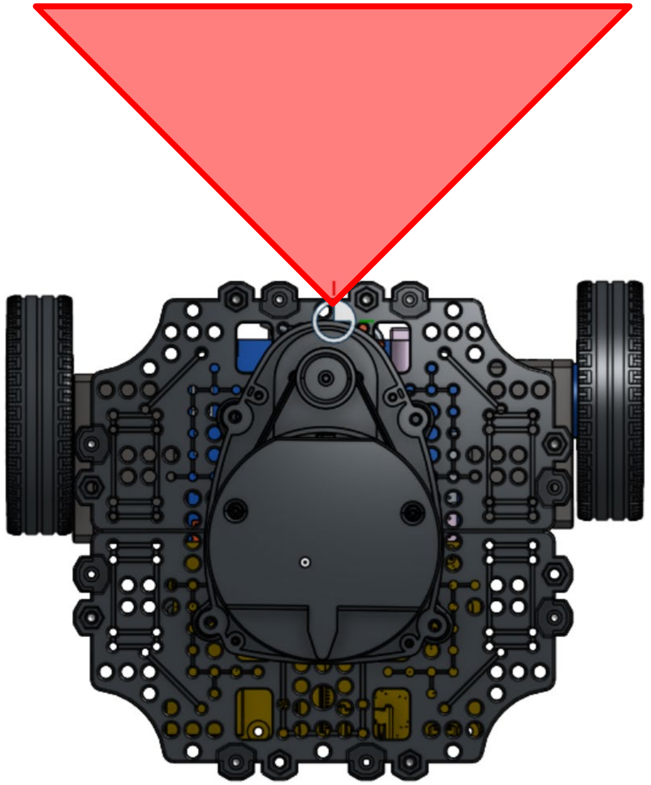




Approach

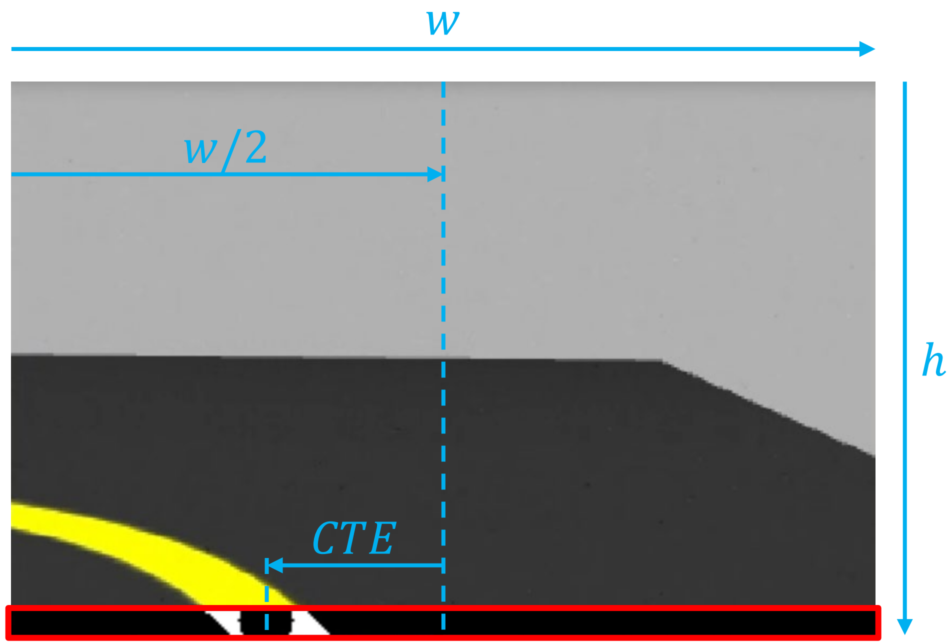
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Crop incoming image: $ROI = \left[\left(\frac{h}{2} + 110\right) : \left(\frac{h}{2} + 120\right)\right], [0{:}\, w]$
  - Convert from RGB to HSV color space and mask specific color
  - Calculate weighted average of pixel 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

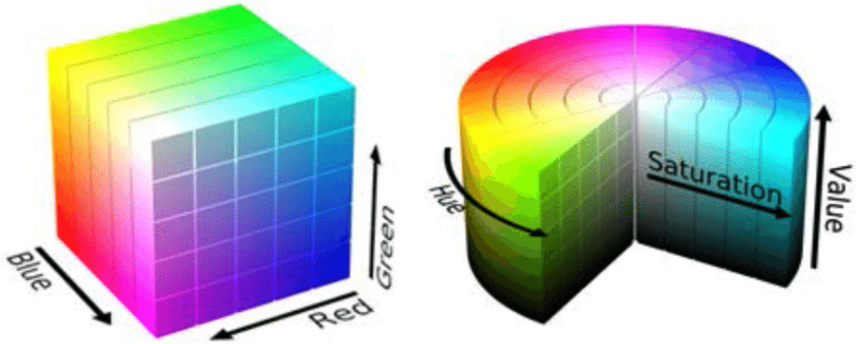
$$\left[\left(\frac{h}{2} + 110\right) : \left(\frac{h}{2} + 120\right)\right], [0{:}\, w]$$



Approach

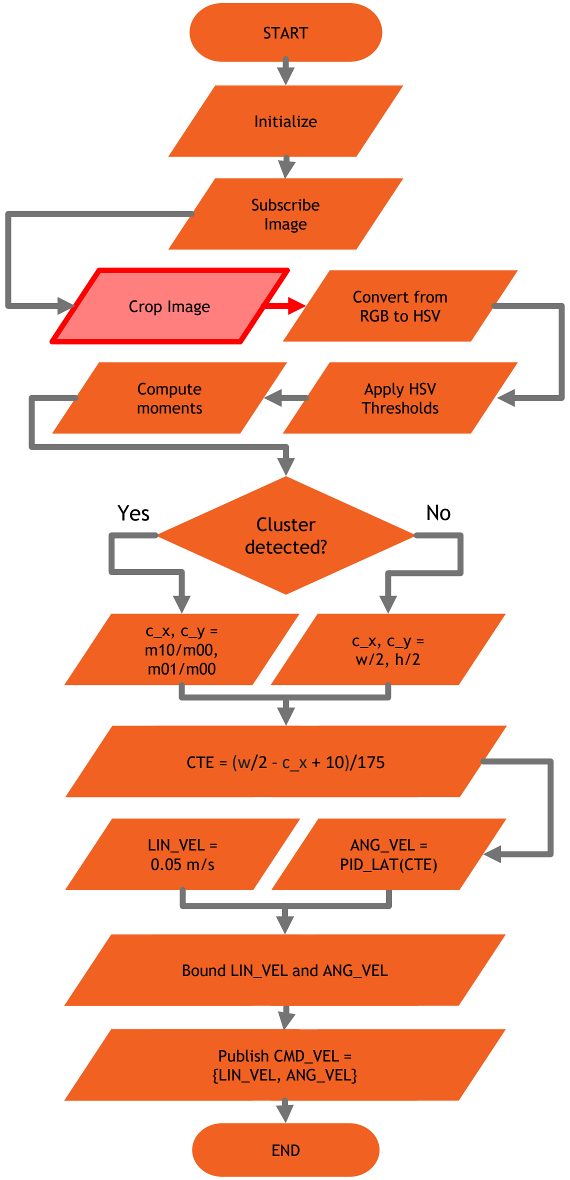
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Crop incoming image: $ROI = \left[\left(\frac{h}{2}+110\right):\left(\frac{h}{2}+120\right)\right], [0:w]$
  - Convert from RGB to HSV color space and mask specific color
  - Calculate weighted average of pixel intensities (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



START

Initialize

Subscribe Image

Crop Image

Convert from RGB to HSV

Apply HSV Thresholds

Compute moments

Cluster detected?

Yes

No

c_x, c_y = m10/m00, m01/m00

c_x, c_y = w/2, h/2

CTE = (w/2 - c_x + 10)/175

LIN_VEL = 0.05 m/s

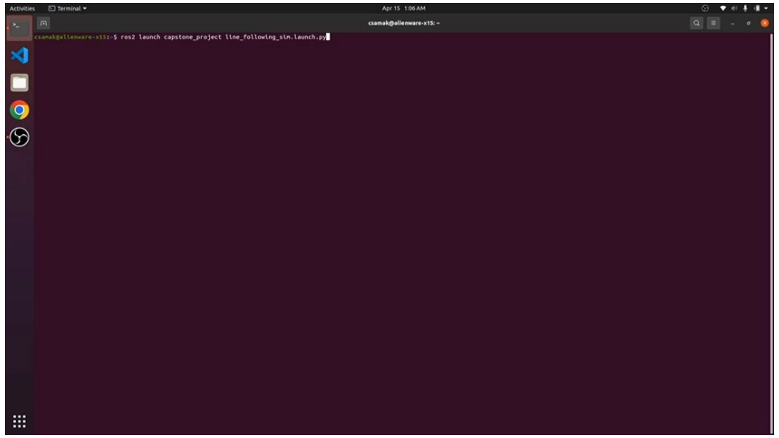
ANG_VEL = PID_LAT(CTE)

Bound LIN_VEL and ANG_VEL

Publish CMD_VEL = {LIN_VEL, ANG_VEL}

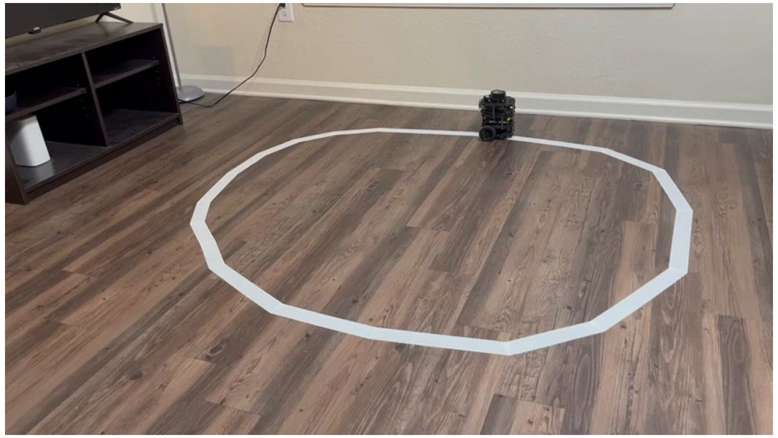
END

Results

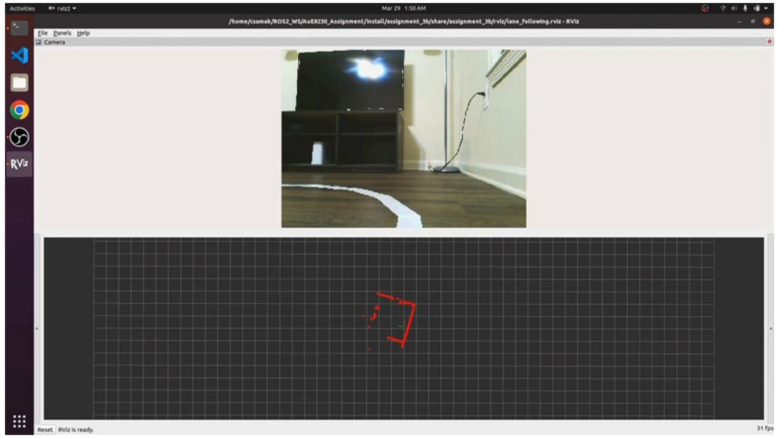


```
csamak@alienware-x15:~$ ros2 launch capstone_project line_following_sim.launch.py
```

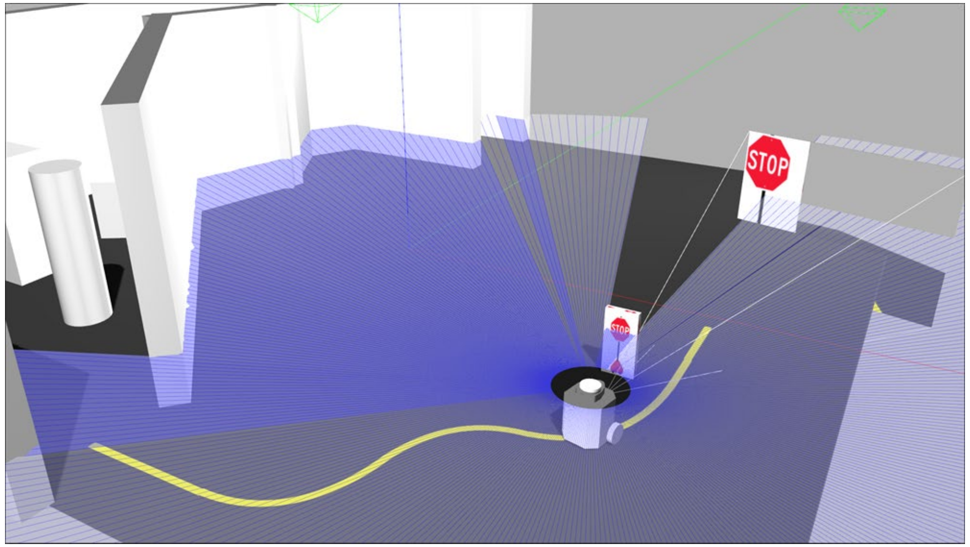
Simulation



TurtleBot3

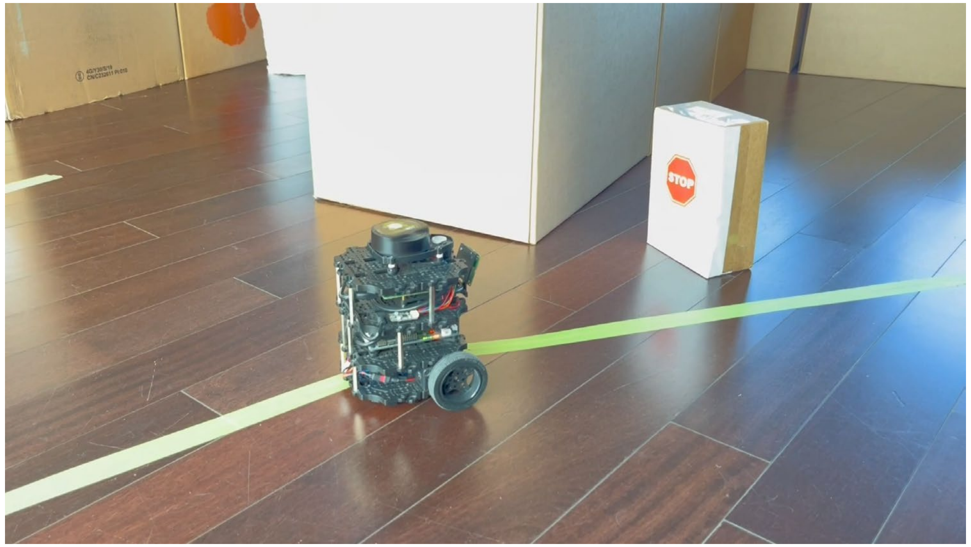

Activities
rviz2
Mar 29 1:50 AM
/home/csamak/ROS2_WS/AuE8230_Assignment/install/assignment_3b/share/assignment_3b/rviz/lane_following.rviz - RViz
File
Panels
Help
Camera
Reset
RViz is ready.
31 fps

Remote PC



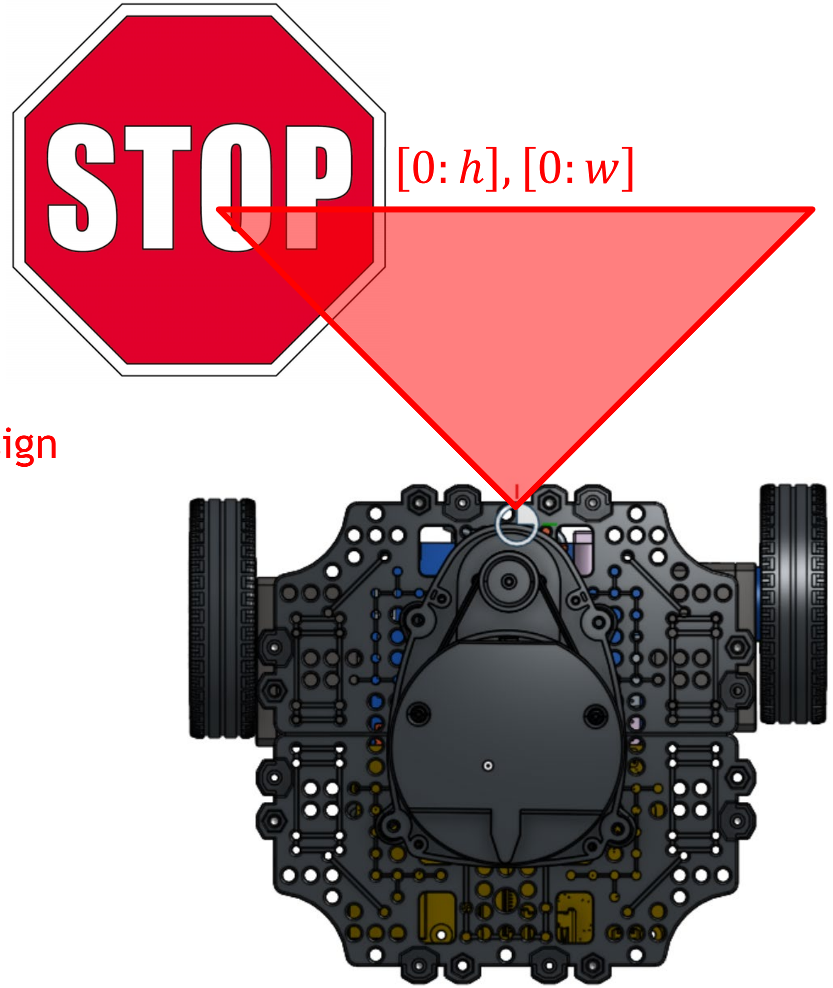
Task 4

Stop Sign Detection



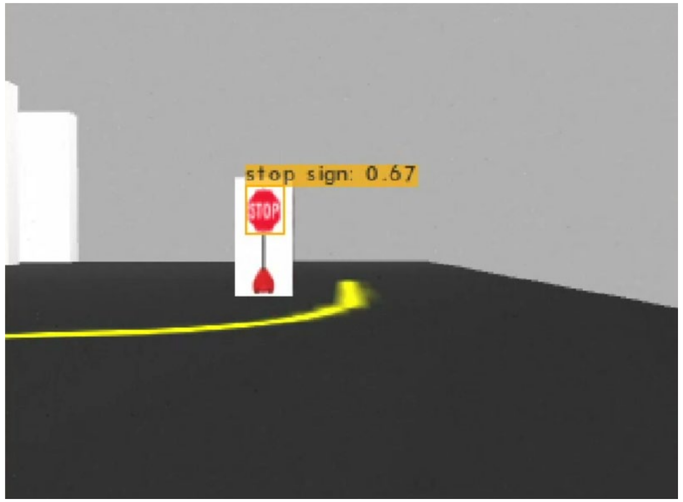
Approach

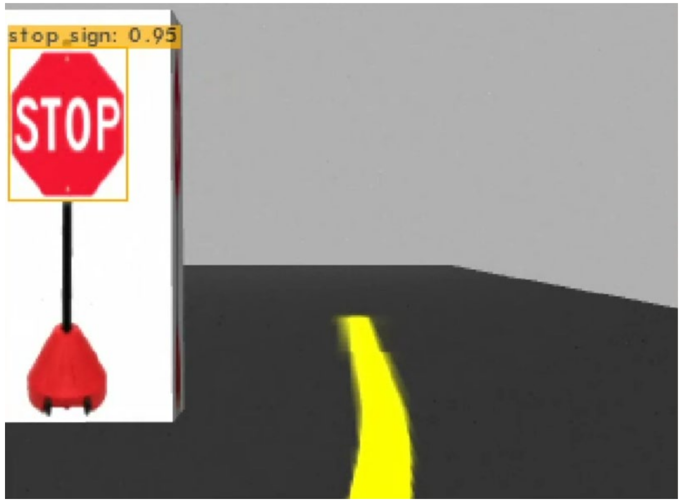
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run Tiny-YOLO V7 inference on incoming image
  - Verify detection class, confidence & bounding box area of stop sign
  - Come to a complete stop for predefined time (4 seconds)
  - Constant `lin_vel` $\begin{cases} v; & \text{stop sign absent} \\ 0; & \text{stop sign present} \end{cases}$ 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

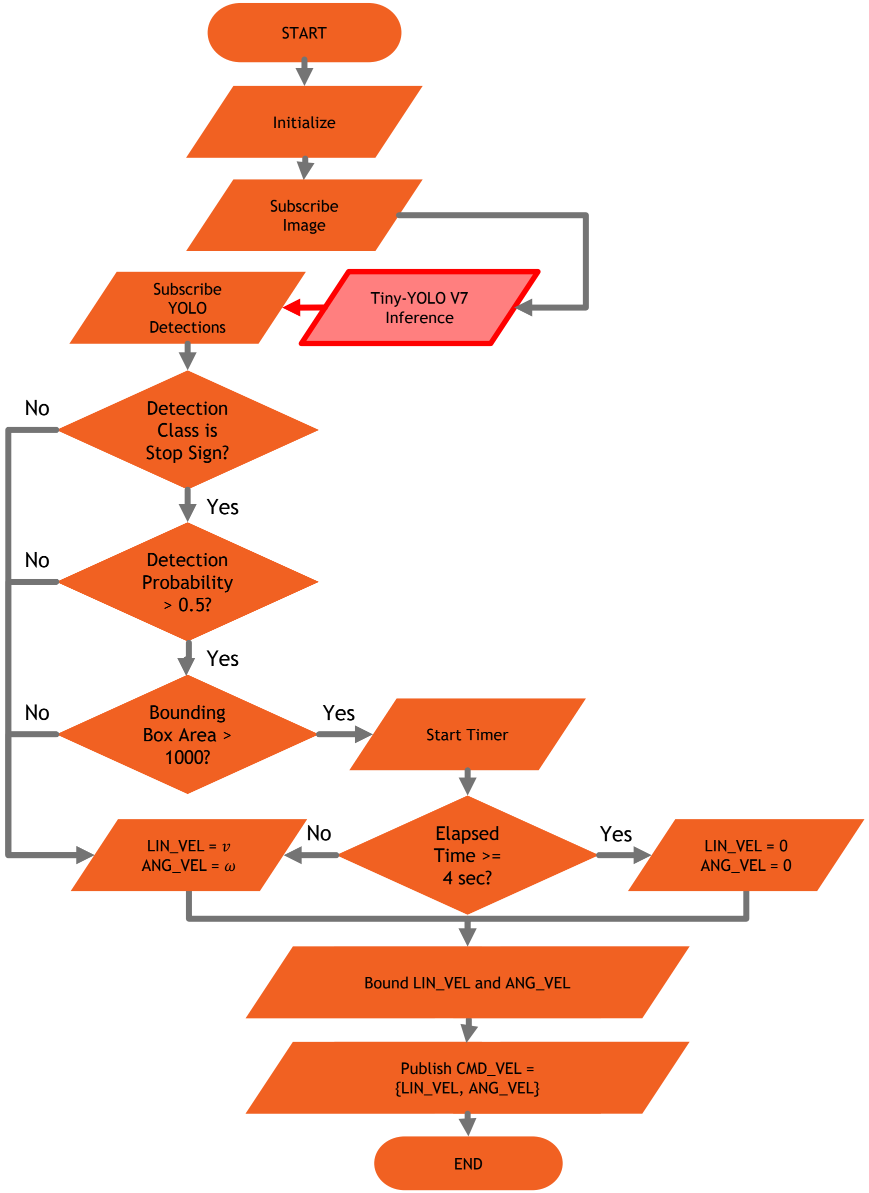
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run Tiny-YOLO V7 inference on incoming image
  - Verify detection class, confidence & bounding box area of stop sign
  - Come to a complete stop for predefined time (4 seconds)
  - Constant `lin_vel` $\begin{cases} v; & \text{stop sign absent} \\ 0; & \text{stop sign present} \end{cases}$ 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





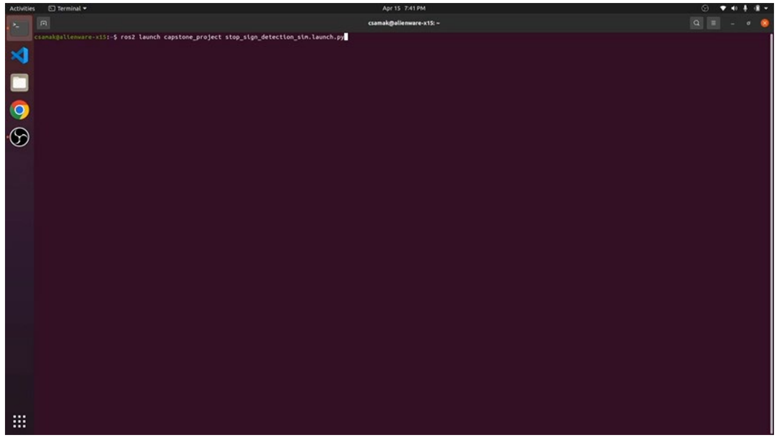
Approach

- **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



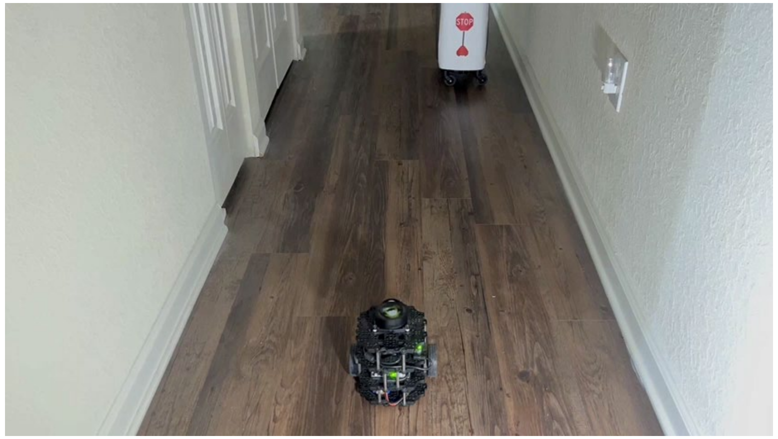
START
Initialize
Subscribe Image
Tiny-YOLO V7 Inference
Subscribe YOLO Detections
Detection Class is Stop Sign?
Yes
No
Detection Probability > 0.5?
Yes
No
Bounding Box Area > 1000?
Yes
No
Start Timer
Elapsed Time >= 4 sec?
No
Yes
LIN_VEL = v
ANG_VEL = ω
LIN_VEL = 0
ANG_VEL = 0
Bound LIN_VEL and ANG_VEL
Publish CMD_VEL = {LIN_VEL, ANG_VEL}
END

Results

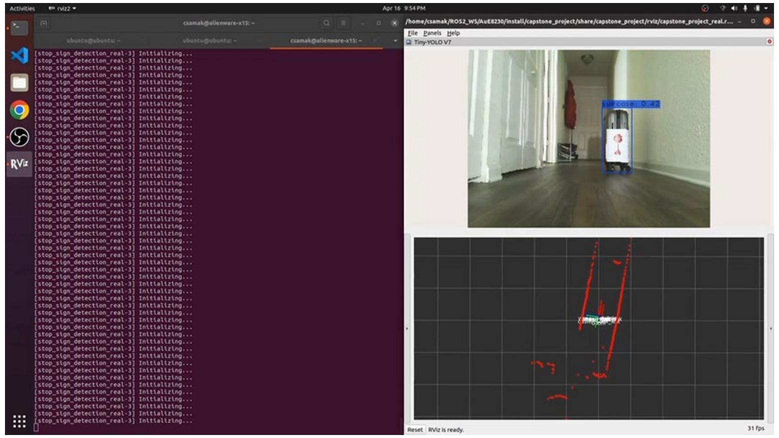


```
csamak@alienware-x15:~$ ros2 launch capstone_project stop_sign_detection_sim.launch.py
```

Simulation

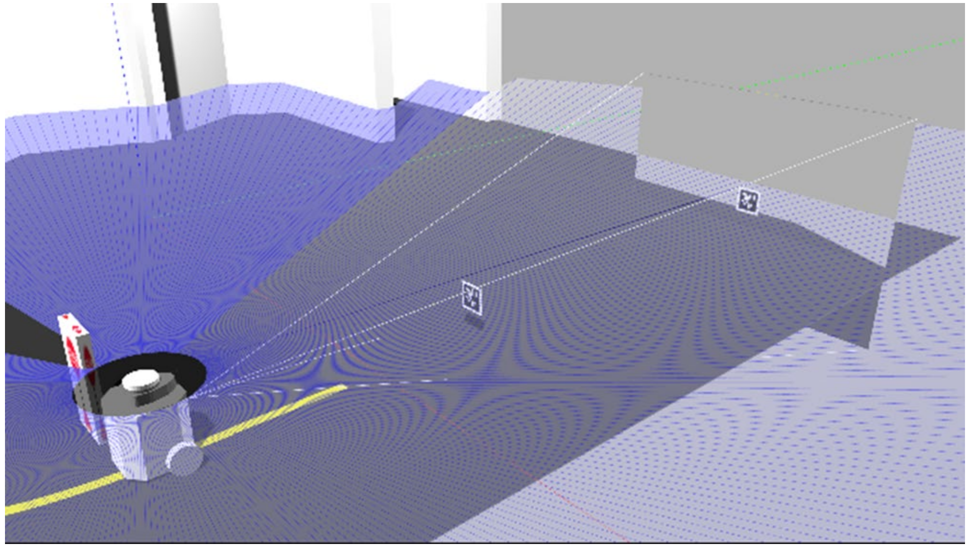


TurtleBot3



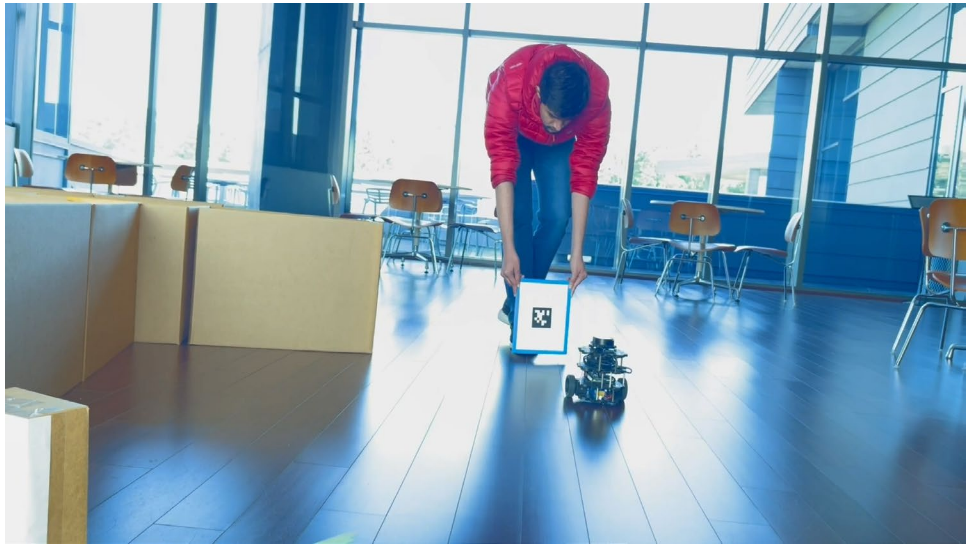
```
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
[stop_sign_detection_real-3] Initializing...
```

Remote PC



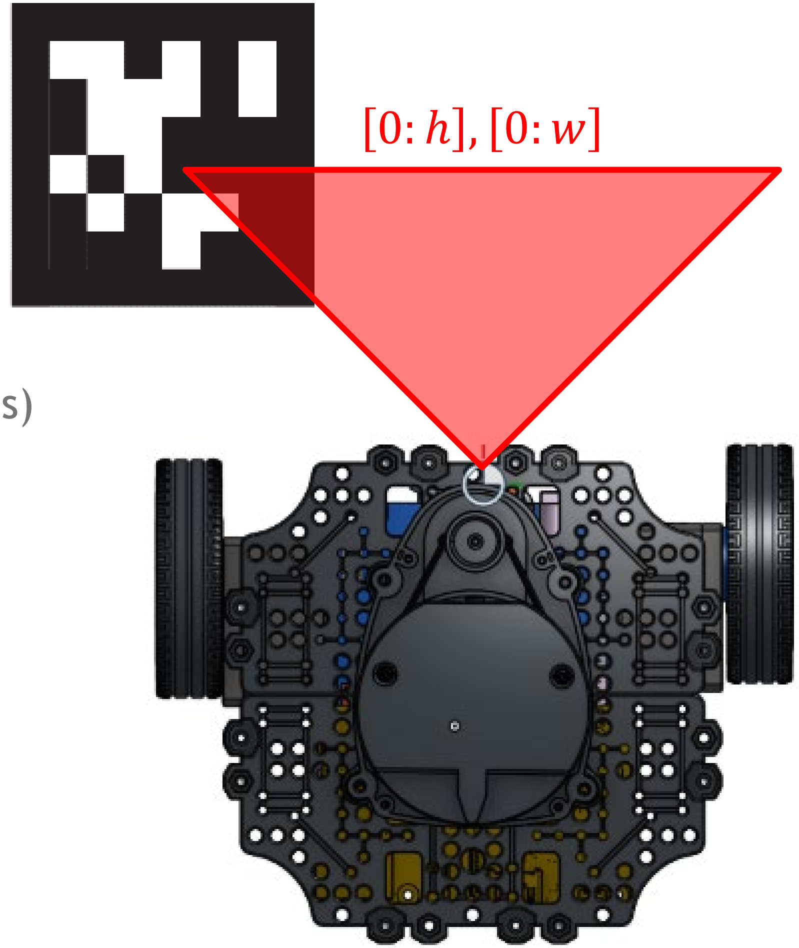
Task 5

AprilTag Tracking



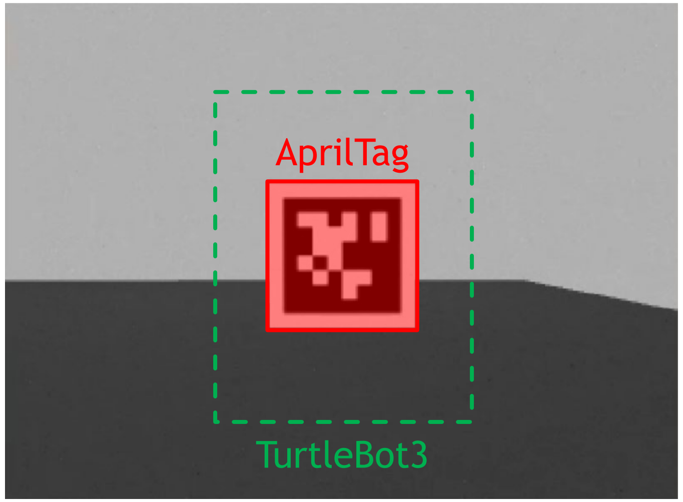
Approach

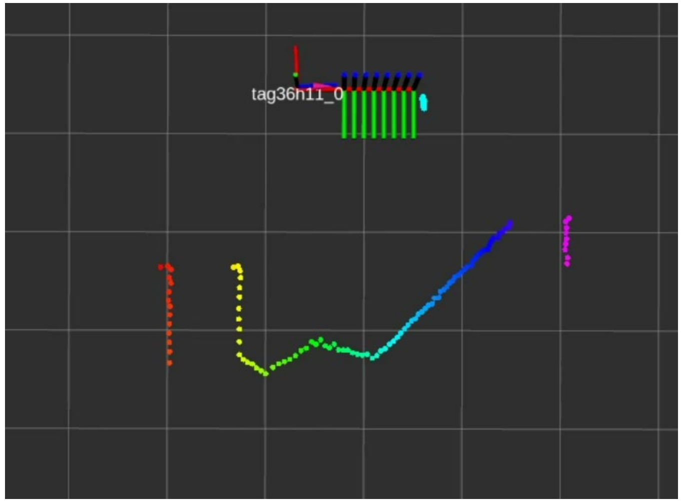
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run AprilTag detection on incoming image
  - Estimate 6D metric pose of marker w.r.t. robot's camera (intrinsics)
  - Apply static transform from `base_link` to `camera`
  - Operate on lateral (-x) & longitudinal (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

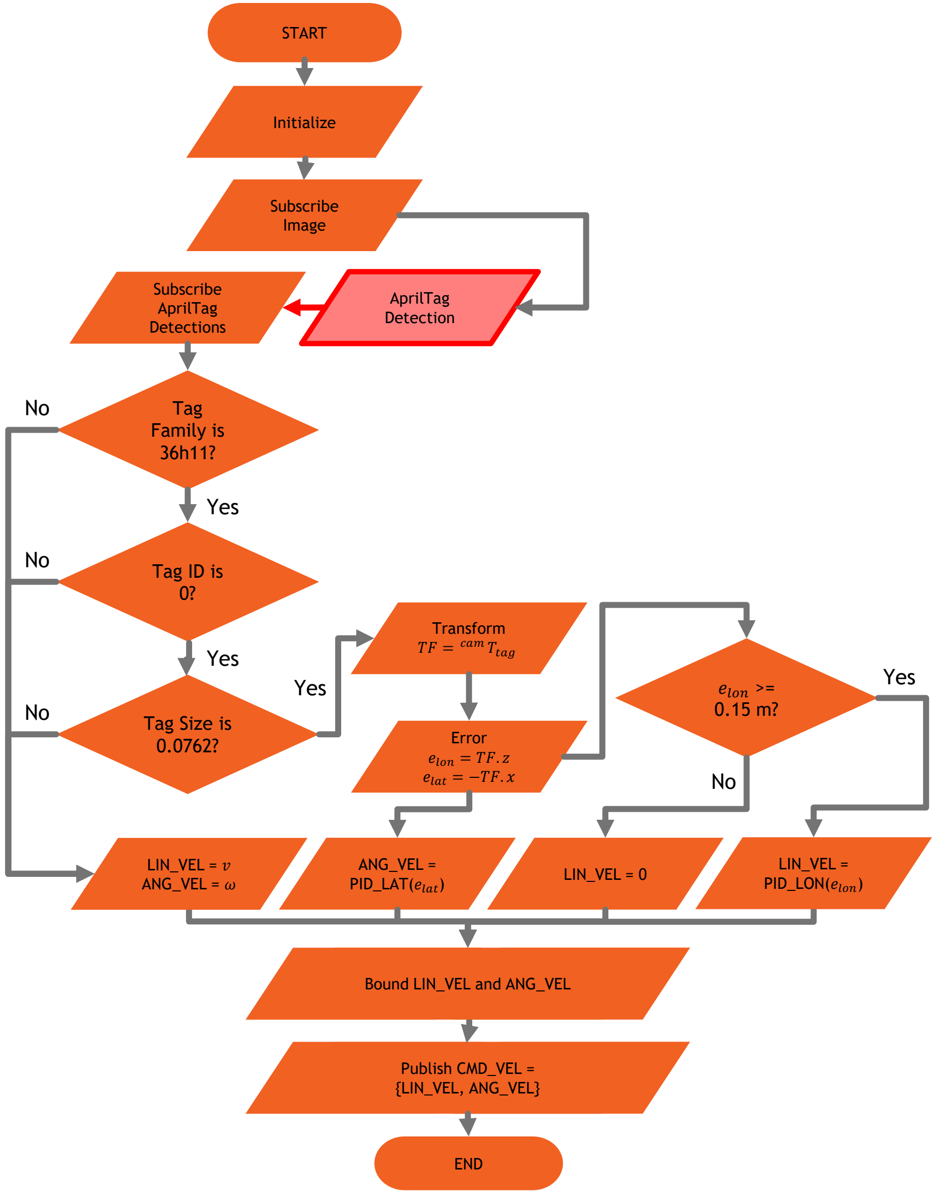
- **Sensors:** Camera (compressed/uncompressed | 320x240 px)
- **Algorithm:**
  - Configure & run AprilTag detection on incoming image
  - Estimate 6D metric pose of marker w.r.t. robot's camera (intrinsics)
  - Apply static transform from `base_link` to `camera`
  - Operate on lateral (-x) & longitudinal (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



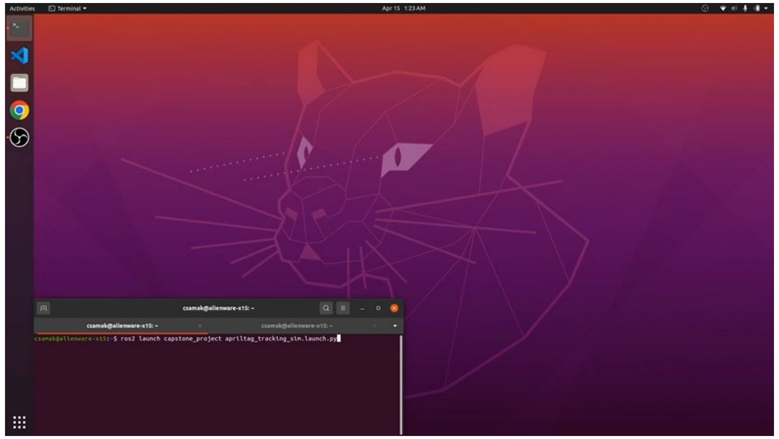


Approach

- **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

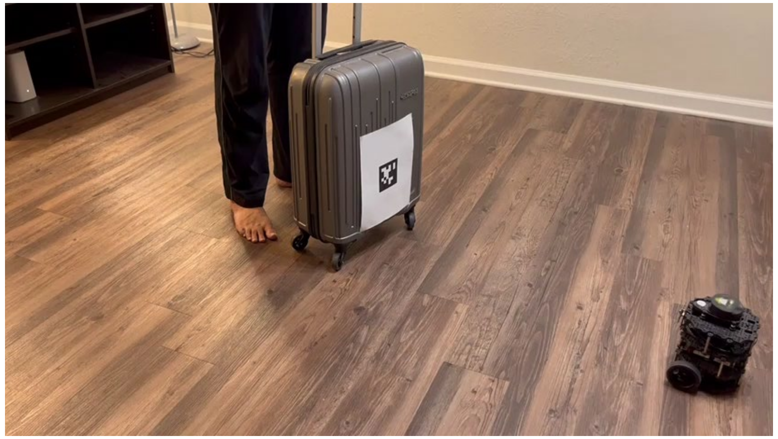

START
Initialize
Subscribe Image
AprilTag Detection
Subscribe AprilTag Detections
Tag Family is 36h11?
No
Yes
Tag ID is 0?
No
Yes
Tag Size is 0.0762?
No
Yes
Transform $TF = {}^{cam}T_{tag}$
Error $e_{lon} = TF.z$ $e_{lat} = -TF.x$
e_{lon} >= 0.15 m?
Yes
No
LIN_VEL = v ANG_VEL = ω
ANG_VEL = PID_LAT(e_{lat})
LIN_VEL = 0
LIN_VEL = PID_LON(e_{lon})
Bound LIN_VEL and ANG_VEL
Publish CMD_VEL = {LIN_VEL, ANG_VEL}
END

Results

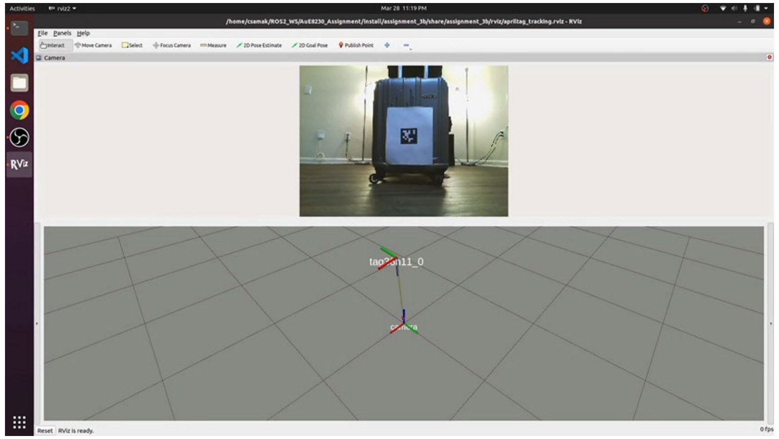


```
csamak@alienware-x15:~$ ros2 launch capstone_project apriltag_tracking_sim.launch.py
```

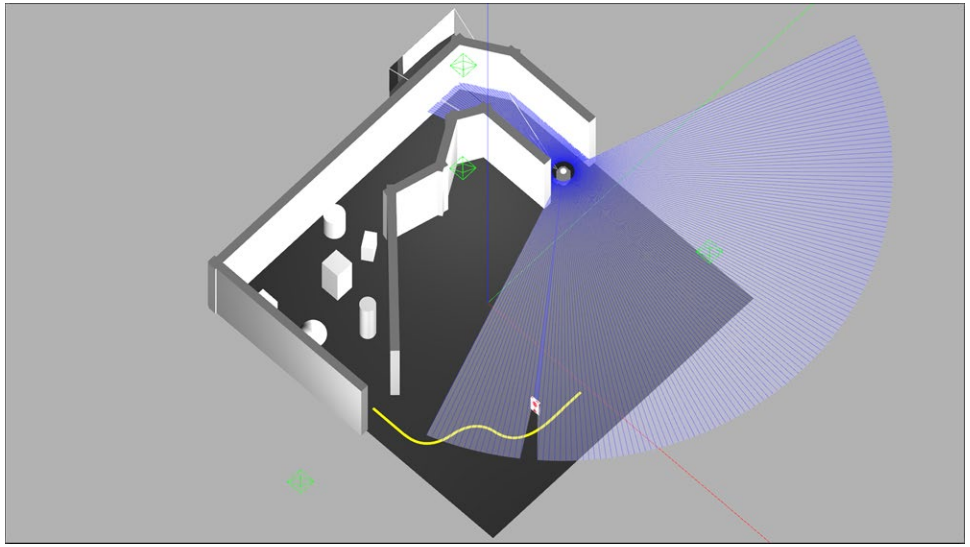
Simulation



TurtleBot3

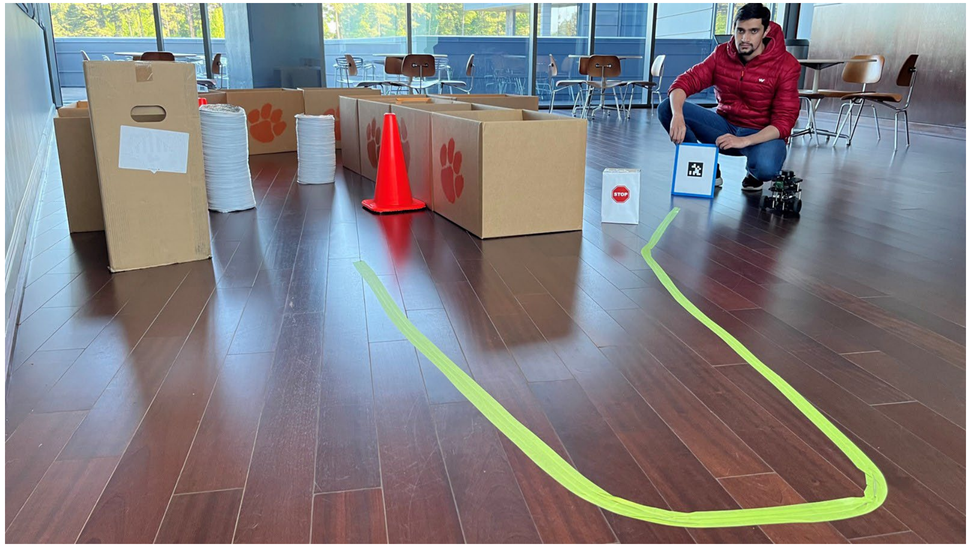


Remote PC



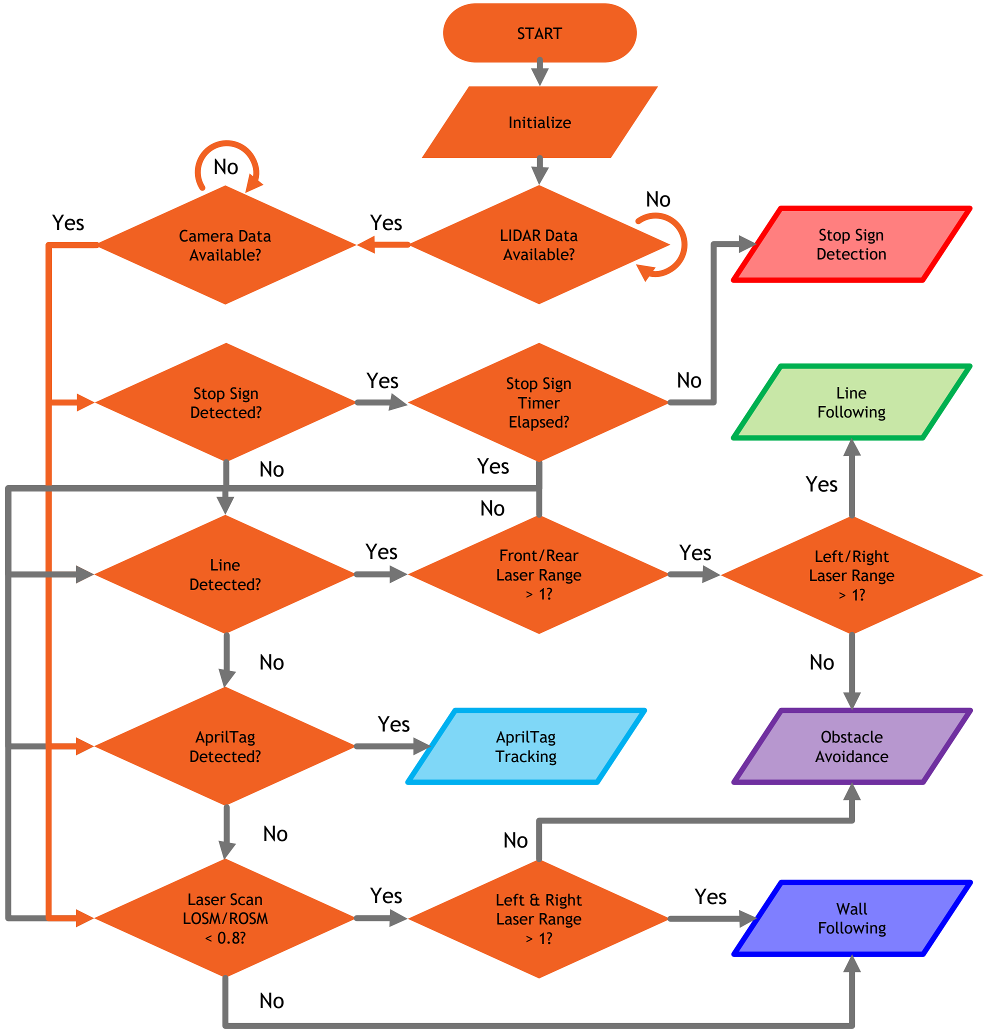
Integration

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

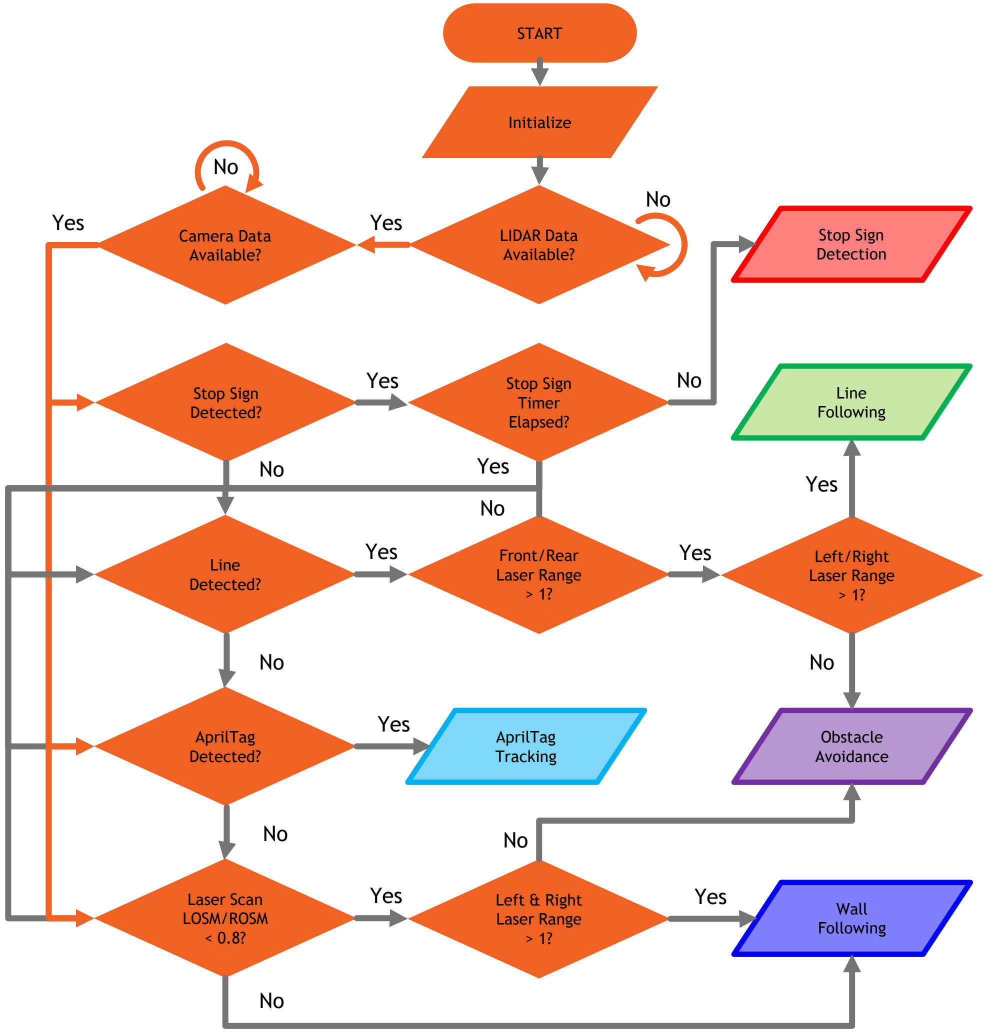
  True if stop sign detected and timer not elapsed
- Priority 2: Line Following

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

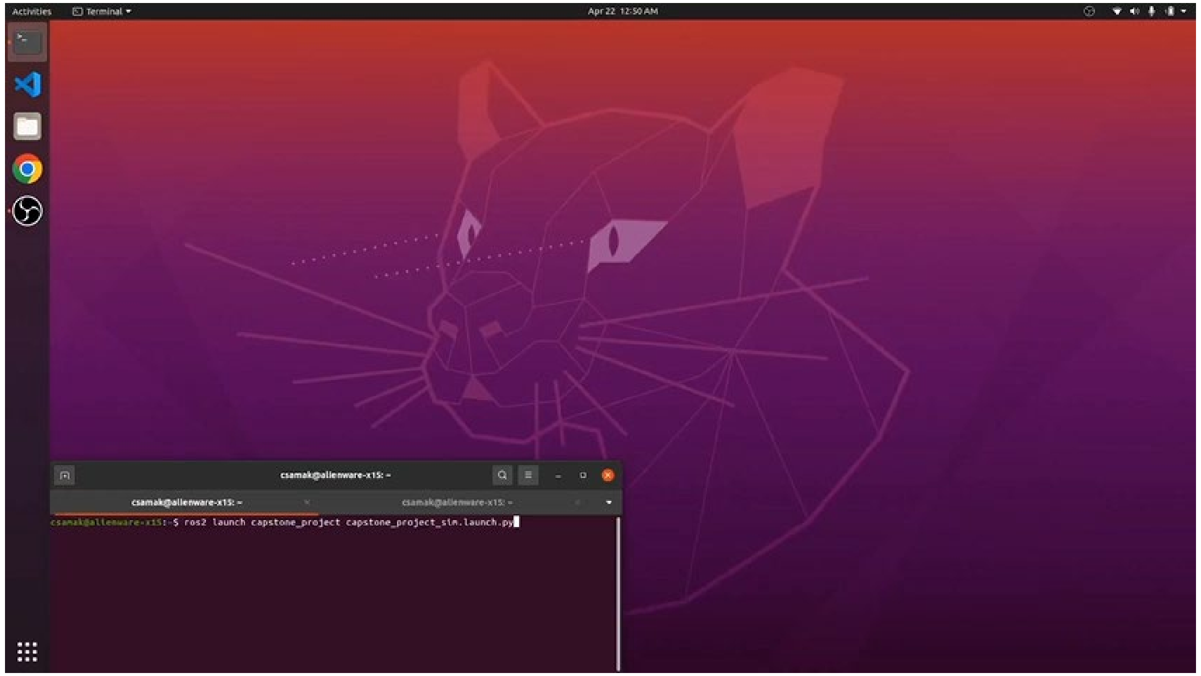
  True if tag detected, line not detected and not stopped at stop sign
- Priority 4: Obstacle Avoidance

  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



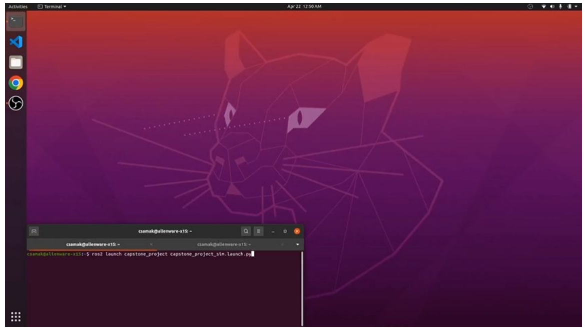
```
csamak@alienware-x15:~$ ros2 launch capstone_project capstone_project_sim.launch.py
```

Simulation

Reality

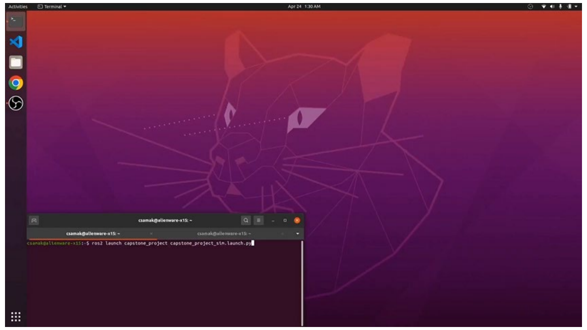
Robustness Testing

Simulation



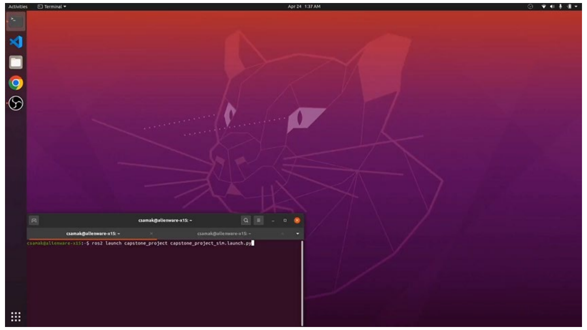
```
csamak@alienware-x15:~$ ros2 launch capstone_project capstone_project_sim.launch.py
```

Forward Direction;
Start with Wall Following



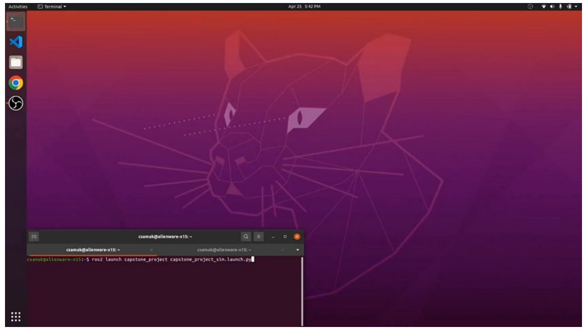
```
csamak@alienware-x15:~$ ros2 launch capstone_project capstone_project_sim.launch.py
```

Forward Direction;
Start with Line Following



```
csamak@alienware-x15:~$ ros2 launch capstone_project capstone_project_sim.launch.py
```

Forward Direction;
Start with AprilTag Tracking

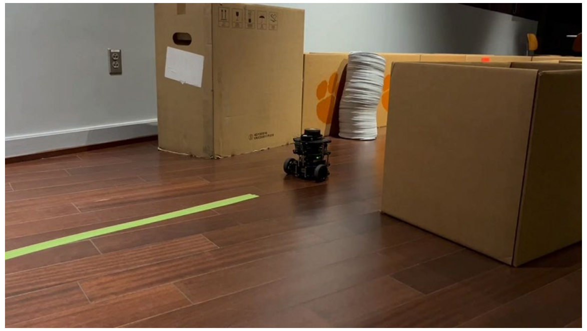


```
csamak@alienware-x15:~$ ros2 launch capstone_project capstone_project_sim.launch.py
```

Inverse Direction;
Start with Line Following

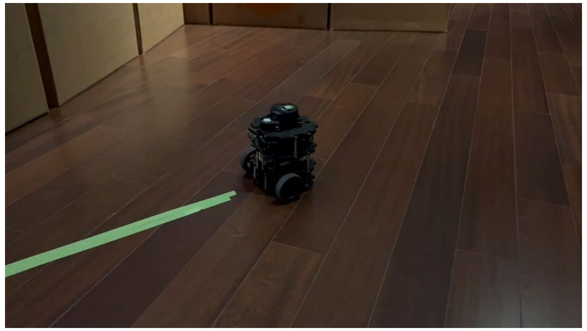
Reality

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