# RL and Autonomous Driving: Week 6

By Brandon Dominique, John Nguyen, Hoang Huynh, and Eric Av

# Problem Statement

- Collision Prevention with CAT Vehicle in Gazebo
  - Reinforcement learning to train CAT Vehicle in Gazebo to detect and avoid potential collisions
  - Train the vehicle on a variety of situations such as collision detection and pedestrian avoidance
  - Use a meta-cognitive radio to relay information to nearby vehicles

# Lit Review: Reinforcement Learning for Cooperative Overtaking

- Proposes 2 general models for cooperative overtaking problems.
    - Cooperative Overtaking: Cutting other cars off.
- Models vehicle dependencies using a Coordinated Graph.
- Introduces new Multiagent Reinforcement Learning framework.

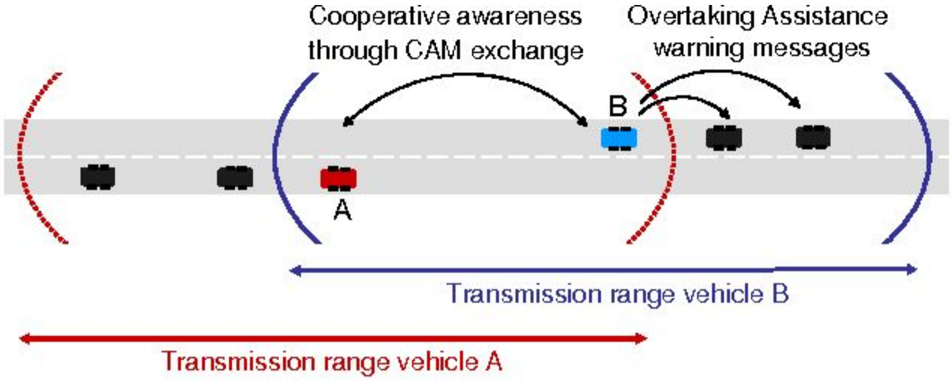
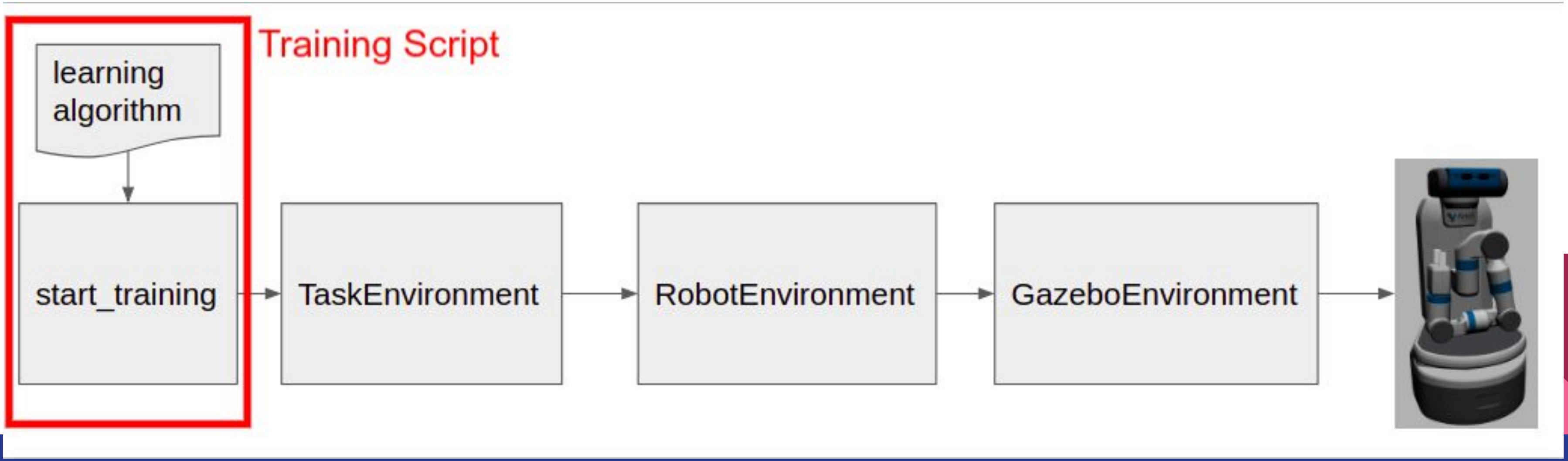

Cooperative awareness through CAM exchange
Overtaking Assistance warning messages
B
A
Transmission range vehicle B
Transmission range vehicle A

Figure 1: Overtaking Assistance scenario.

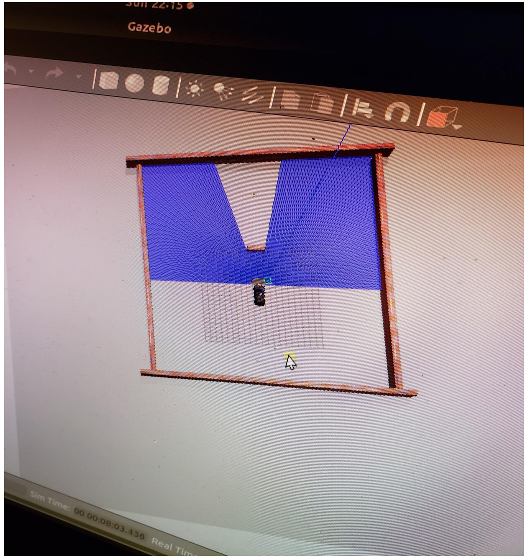
# OpenAI ROS Implementation: Done

- RobotEnvironment, TaskEnvironment for CATVehicle and collision prevention works.



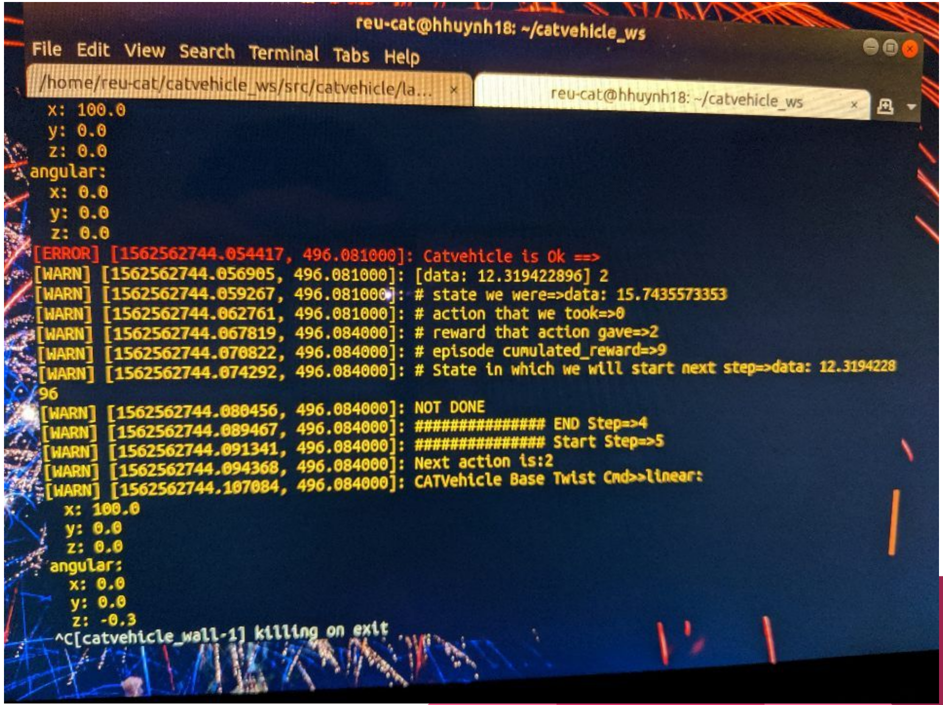
# ROS and Gazebo Environment Setup

- Refined the test environment world to accommodate for failed iterations with boundary
- Debugged the world file to spawn obstacles where they should be.
  - By resetting the world, all obstacles are spawned, but not moved to initial positions.



# ROS and Gazebo Training

- Connected the /catvehicle/distanceEstimatorSteeringBased/.. parameters to our training model
- Implemented a basic Q-learning algorithm
- Can convert CATVehicle and the world its in into an OpenAI Gym Environment


reu-cat@hhuynh18: ~/catvehicle_ws
File Edit View Search Terminal Tabs Help
/home/reu-cat/catvehicle_ws/src/catvehicle/la...
reu-cat@hhuynh18: ~/catvehicle_ws

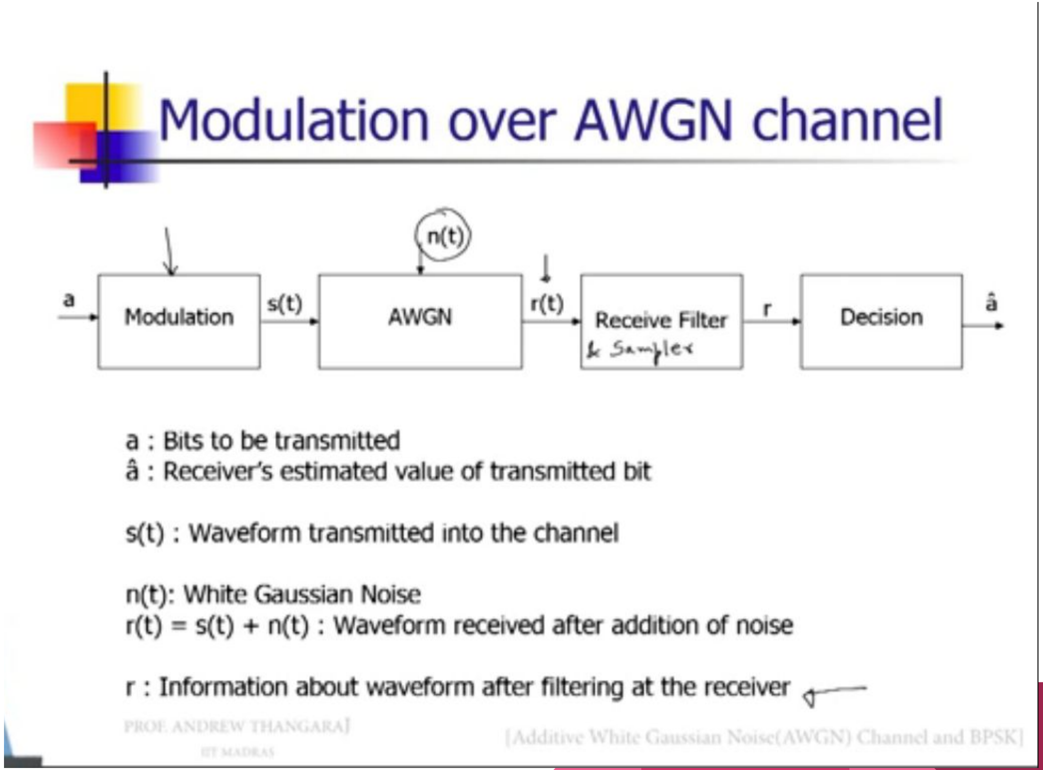
```
  x: 100.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
[ERROR] [1562562744.054417, 496.081000]: Catvehicle is Ok ==>
[WARN] [1562562744.056905, 496.081000]: [data: 12.319422896] 2
[WARN] [1562562744.059267, 496.081000]: # state we were=>data: 15.7435573353
[WARN] [1562562744.062761, 496.081000]: # action that we took=>0
[WARN] [1562562744.067819, 496.084000]: # reward that action gave=>2
[WARN] [1562562744.070822, 496.084000]: # episode cumulated_reward=>9
[WARN] [1562562744.074292, 496.084000]: # State in which we will start next step=>data: 12.3194228
96
[WARN] [1562562744.080456, 496.084000]: NOT DONE
[WARN] [1562562744.089467, 496.084000]: ############### END Step=>4
[WARN] [1562562744.091341, 496.084000]: ############### Start Step=>5
[WARN] [1562562744.094368, 496.084000]: Next action is:2
[WARN] [1562562744.107084, 496.084000]: CATVehicle Base Twist Cmd>>linear:
  x: 100.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: -0.3
^C[catvehicle_wall-1] killing on exit
```

# Lit Review: Interference-Based Detection for Spectrum Sensing

- ns3-gym: Extending OpenAI Gym for Networking Research
  - observation — occupation on each channel in the current time slot, i.e. wideband-sensing,
  - actions — set the channel to be used for the next time slot,
  - reward — +1 in case of no collision with interferer; otherwise -1,
  - gameover — if more than three collisions happened during the last ten time-slots
- Cognitive ALE for HF Radios
  - 3 Types of Spectrum Sensing
    - Transmitter Detection, Cooperative Detection, and Interference Based Detection

1. Input Text (“Hello World”)
2. Convert to Hex/Binary
3. Modulate
4. Channel
5. Demodulate

- Expand for images and other data?



## Modulation over AWGN channel

a : Bits to be transmitted
â : Receiver’s estimated value of transmitted bit

s(t) : Waveform transmitted into the channel

n(t): White Gaussian Noise
r(t) = s(t) + n(t) : Waveform received after addition of noise

r : Information about waveform after filtering at the receiver

PROF. ANDREW THANGARAJ
IIT MADRAS

[Additive White Gaussian Noise(AWGN) Channel and BPSK]

# Future Plans

- Improve and optimize the Robot and Task Environments
  - Current problem: After ~250 iterations, our machine runs out of RAM.
  - Utilize all sensors of the car.
  - Come up with a better reward system.
  - Introduce a random pedestrian to walk in front of the car.
- Write and train on more complex reinforcement learning algorithms.
  - Then, upload the trained neural network to a database.
- Set up meta learning architecture.
  - Will need many more worlds to train in.
- Complete 2 channels and 2 Modulation Schemes
  - Test them with sample bits in ns3 Environment (needs to be built)