

Notes for Reinforcement Learning for CAT Vehicle

July 18, 2019

1 Notes on Machine Learning for Autonomous Driving Week 2

-Probability Theory:

-Bayes Theorem: A theorem describing how the conditional probability of each of a set of possible causes for a given observed outcome can be computed from knowledge of the probability of each cause and the conditional probability of the outcome of each cause.

-Conditional Probability:

-Classifiers:

-Supervised and Unsupervised learning:

-Reinforcement Learning:

A way for programming agents to be rewarded and punished without needing to specify how a task is to be achieved. Problems faced with trial and error learning interactions with the programming agents. Two ways of solving these problems. The first, is to search in the space of behaviors in order to find one that performs well in the environment. The second is to use statistical techniques and dynamic programming methods to estimate the utility of taking actions in states of the world. <https://www.jair.org/index.php/jair/article/view/10166>

-Meta Learning: It introduces realistic class imbalances. This varies the number of classes in each task and the size of the training set.

2 ROS and Gazebo learning Week 3

- Studied ROS and Gazebo relationship and importance
 - Followed Car movement and forward sensor tutorial on GazeboSim
 - Created first workspace with ROS and Gazebo to create a basic vehicle
 - Used Simulink to command movement of car, essential for further implementation ideas
 - Tutorials help but outdated tutorials are often misleading
 - Troubles creating first workspace with ROS and Gazebo (again with the folders and different extensions ie: .launch, .world., .urdf or .sdf)
 - Miscellaneous struggles concerning Ubuntu and console command learning curve

3 Implementing simple reinforcement learning simulation week 4 6/25/2019

- Trying to publish and subscribe sensor data to cat vehicle topic in gazebo ros
 - Following Sprinkles tutorial at <https://cps-vo.org/node/26594>

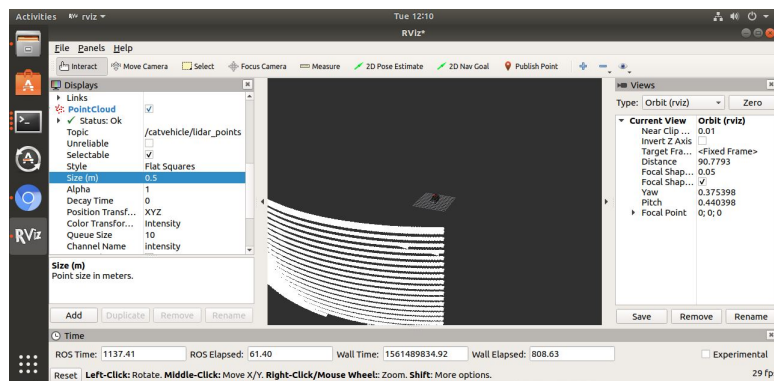


Figure 1: RVIZ

- Followed tutorial on publishing to a node using python: "[ROS In 5 Minutes] 003 - How to create a ROS Publisher"
 - Connected tutorial to catvehicle in src folder
 - Created working publisher with catvehicle running in Gazebo

```
#!/usr/bin/env python

import rospy
from std_msgs.msg import String

rospy.init_node('subscriber') #position of car, , distEstimator

publisher = rospy.Publisher('hi', String, queue_size=1)
rate = rospy.Rate(3)

while not rospy.is_shutdown():
    publisher.publish('Hey!')
    rate.sleep()
```

Figure 2: Publisher

4 Implementing simple reinforcement learning simulation week 4 6/26/2019

-Attempting to add subscriber to receive information from a catvehicle sensor and manipulate that with a publisher.

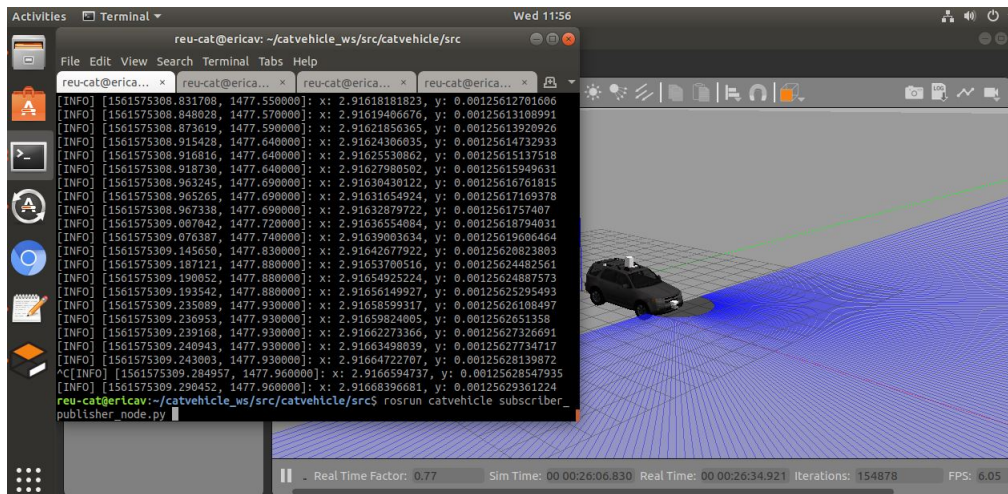


Figure 3: Odometry information

- Succeeded in subscribing odometry information of the CAT vehicle
- Found out how to access position of any topic on python

```
1 #!/usr/bin/env python
2
3 import rospy
4 from nav_msgs.msg import Odometry
5
6 def callback(msg):
7     x = msg.pose.pose.position.x
8     y = msg.pose.pose.position.y
9     rospy.loginfo('x: {}, y: {}'.format(x,y))
10
11 def main():
12     rospy.init_node('subscriber_publisher')
13     rospy.Subscriber("/catvehicle/odom", Odometry, callback)
14     rospy.spin()
15
16 if __name__ == '__main__':
17     main()
```

```
reu-cat@ericav: ~/catvehicle_ws$ rostopic show nav_msgs/Odometry
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
string child_frame_id
geometry_msgs/PoseWithCovariance pose
  geometry_msgs/Pose pose
    geometry_msgs/Point position
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion orientation
      float64 x
      float64 y
      float64 z
      float64 w
    float64[36] covariance
  geometry_msgs/TwistWithCovariance twist
    geometry_msgs/Twist twist
      geometry_msgs/Vector3 linear
        float64 x
        float64 y
        float64 z
```

Figure 4: topic position

5 Implementing simple reinforcement learning simulation week 4 6/27/2019

-Created an openai working example of reinforcement learning

6 Implementing simple reinforcement learning simulation week 4 6/28/2019

-Learn and deconstruct the reinforcement learning openai example for use in CAT Vehicle

7 Implementing simple reinforcement learning simulation week 5 7/2/2019

-Finished world for testing simple object collision

-Made draft for task environment and training script for openai in relation to the CAT Vehicle

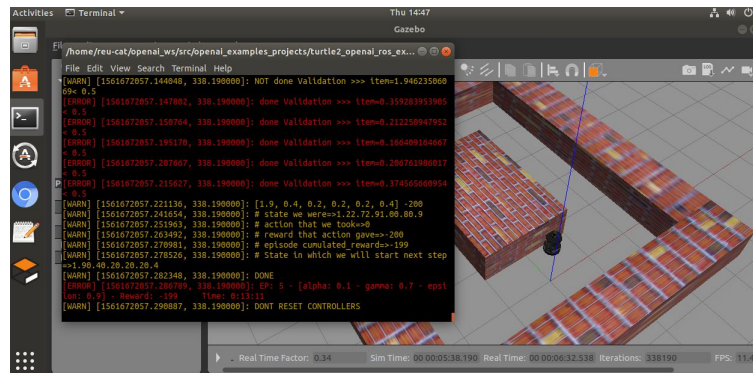


Figure 5: openai working example

8 Implementing simple reinforcement learning simulation week 6 7/8/2019

-Plan to implement more worlds for training

9 Implementing simple reinforcement learning simulation week 6 7/9/2019

-Create a brick(person) moving in a world with the catvehicle

10 Implementing simple reinforcement learning simulation week 6 7/10/2019

-Created CAT vehicle moving across the reinforcement learning induced CAT Vehicle world

11 Implementing simple reinforcement learning simulation week 7 7/17/2019

-Built world with human collision path

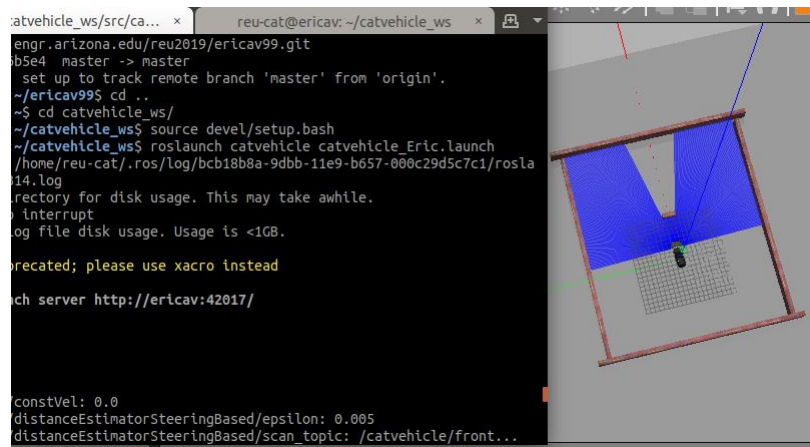


Figure 6: openai environment for CAT Vehicle object collision test

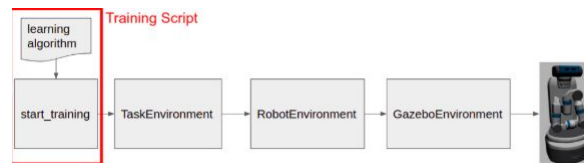


Figure 7: openai and how it works with ROS and Gazebo

12 Implementing simple reinforcement learning simulation week 8 7/18/2019

-Build world with curved road and human collision path

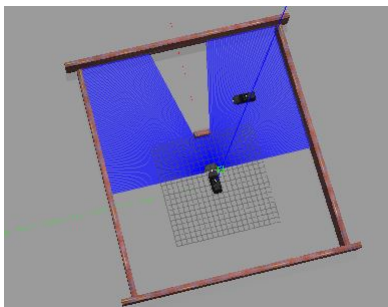


Figure 8: CAT vehicle moving across the reinforcement learning induced CAT Vehicle world

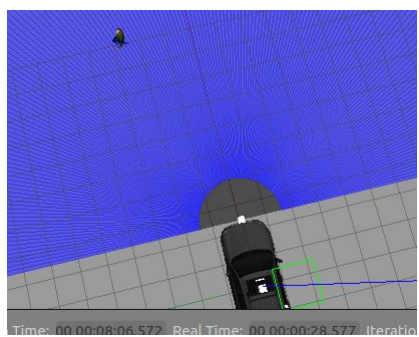


Figure 9: CAT vehicle and a person