ROS Noetic Assignment — Differential-Drive Kinematics on turtlesim

Dheepikha. V (2022105507)

Nodes to Implement
fk_wheels_to_twist.cpp

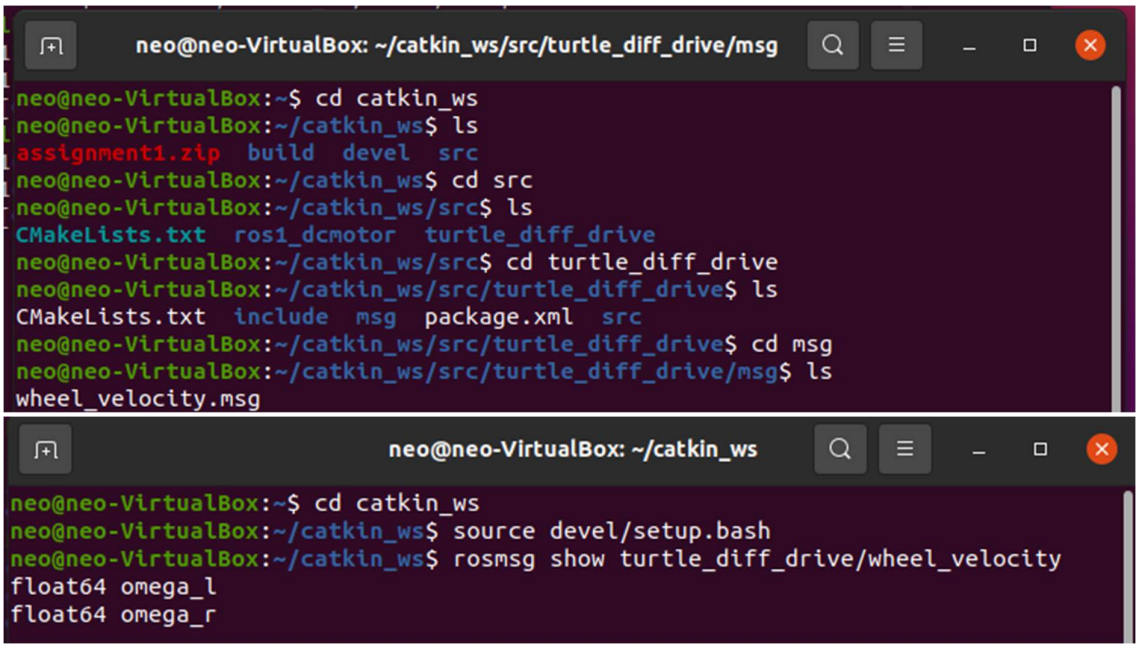
- Sub: ~wheel_vel_in → WheelVel
- Pub: /turtle1/cmd_vel → Twist
- Implements FK.

go_to_goal_controller.cpp

- Sub: /turtle1/pose
- Pub: /turtle1/cmd_vel → Twist
- Implements IK.
- Pub: ~wheel_vel_dbg → WheelVel (from IK)
- Implements proportional controller.

1. fk_wheels_to_twist.cpp
wheel_velocity_pub.cpp

- This node published to the **topic - wheel_vel** - it publishes the **message - wheel_velocity (omega_l,omega_r)** the left and right wheel velocities of the differential drive system resp.
- First created a **catkin_ws** - with the package **turtle_diff_drive** which contains all the nodes which are to be implemented.
- In the package first created **subdirectory - msg** in which the custom msg type was defined **wheel_velocity.msg with omega_l, omega_r.**
- Compilation using catkin_make was successful and after modifying CMakeLists.txt and package.xml, sourcing the setup.bash ros custom msg type was displayed.



wheel_velocity_pub.cpp

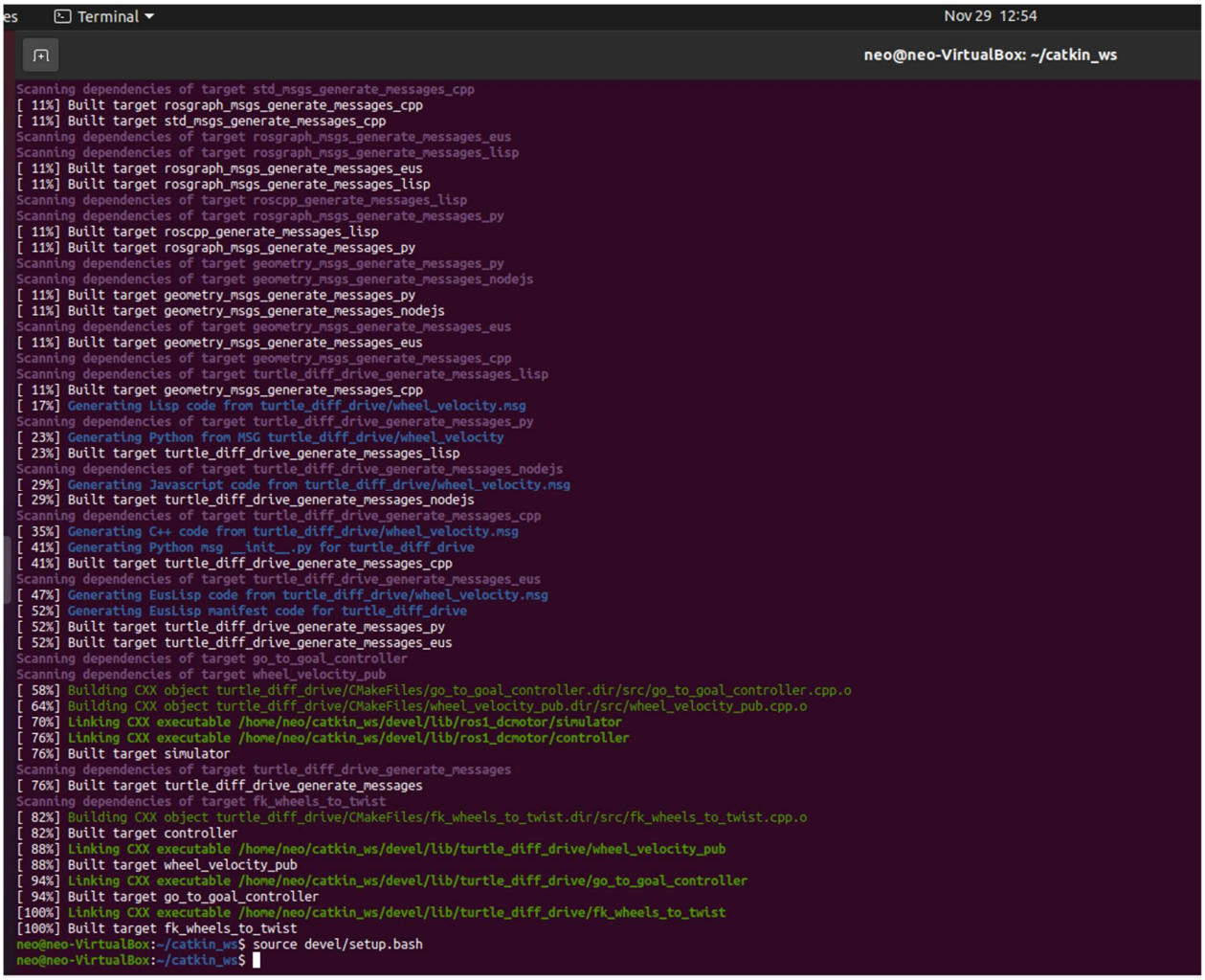
- This node publishes constant values for left and right wheel velocity - **omega_l = 12 rad/s and omega_r = 8 rad/s** (used unequal quantities to demonstrate circular motion- experimented using equal values for demonstrating linear motion as well).
- This node published the **msg of type wheel_velocity to the topic - wheel_vel.**

```
turtle_diff_drive::wheel_velocity msg;
msg.omega_l=12;
msg.omega_r=8;
```

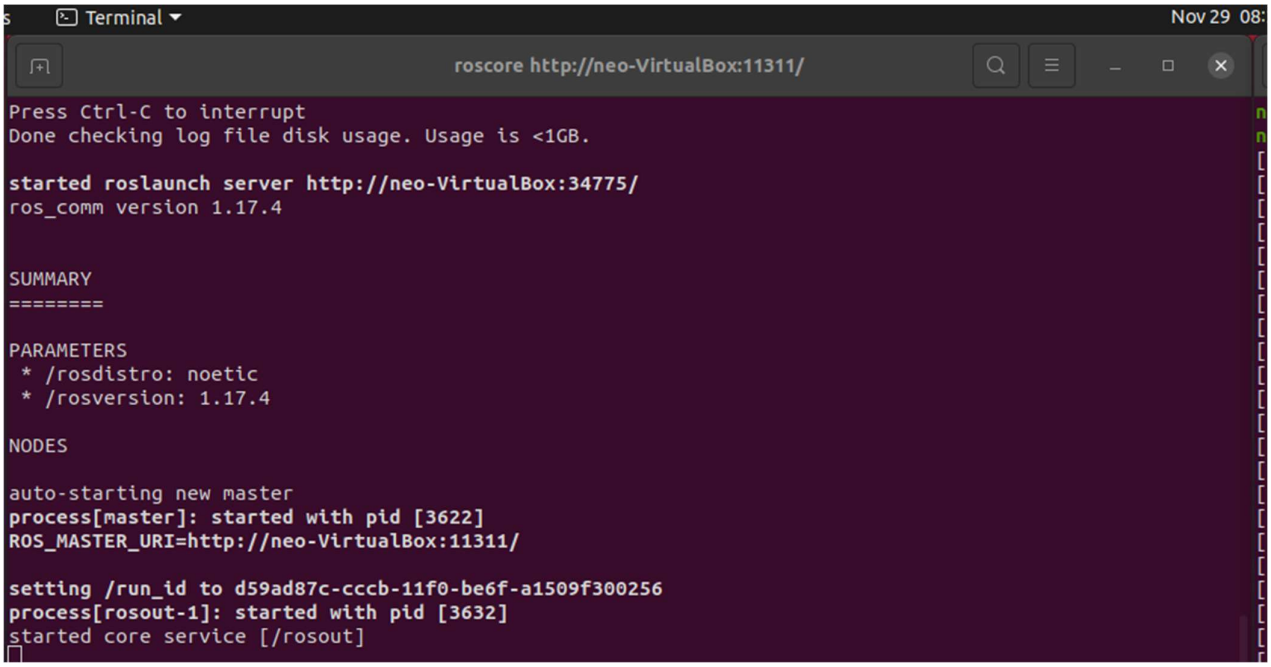
fk_wheels_to_twist.cpp

- This node subscribes to the **topic wheel_vel** and listens to the msgs published by the **wheel_velocity_pub node.**
- In this node - the linear and angular velocities v and w which are **the message fields of the message type std_msgs/Twist** were calculated using the formula given with constant wheel radius =0.05m and axle length =0.20 m.
- This message of type std_msgs/Twist was then **published to the topic turtle1/cmd_vel** which the turtlesim_node listens to and responds.

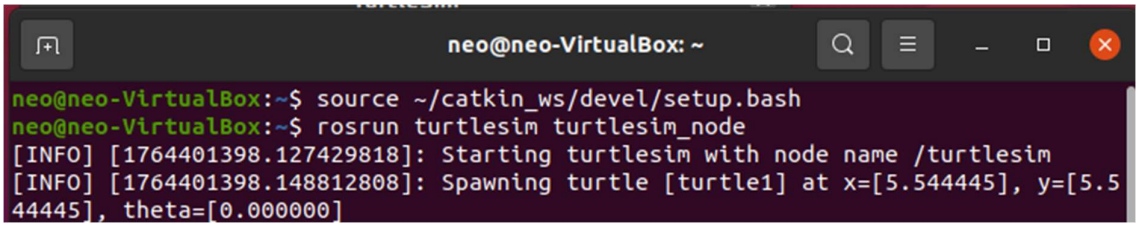
Successful build using catkin_make command

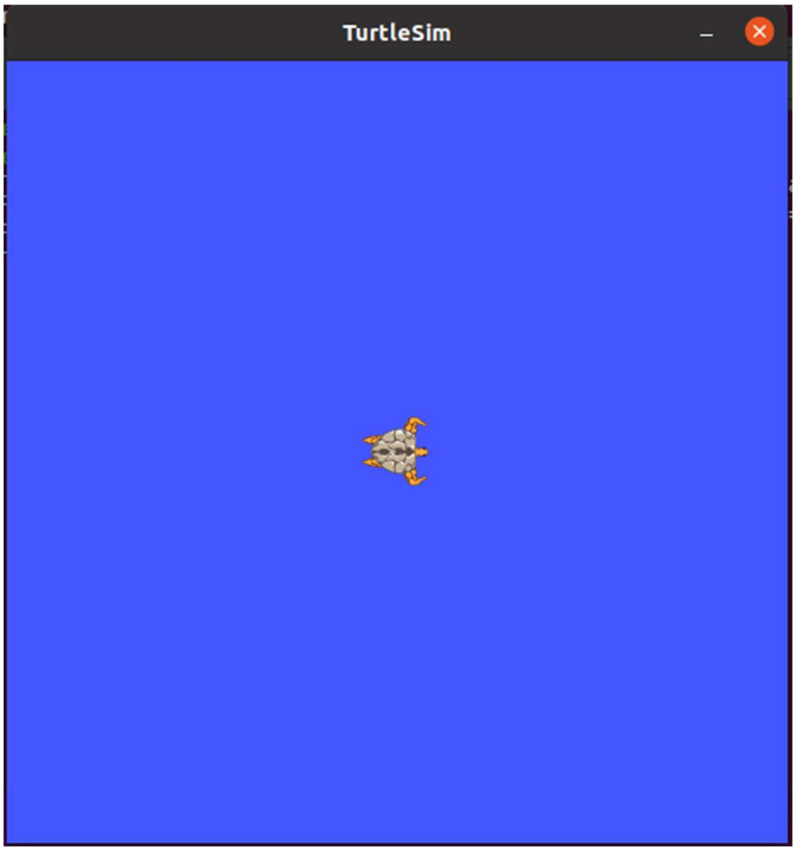


EXECUTION:

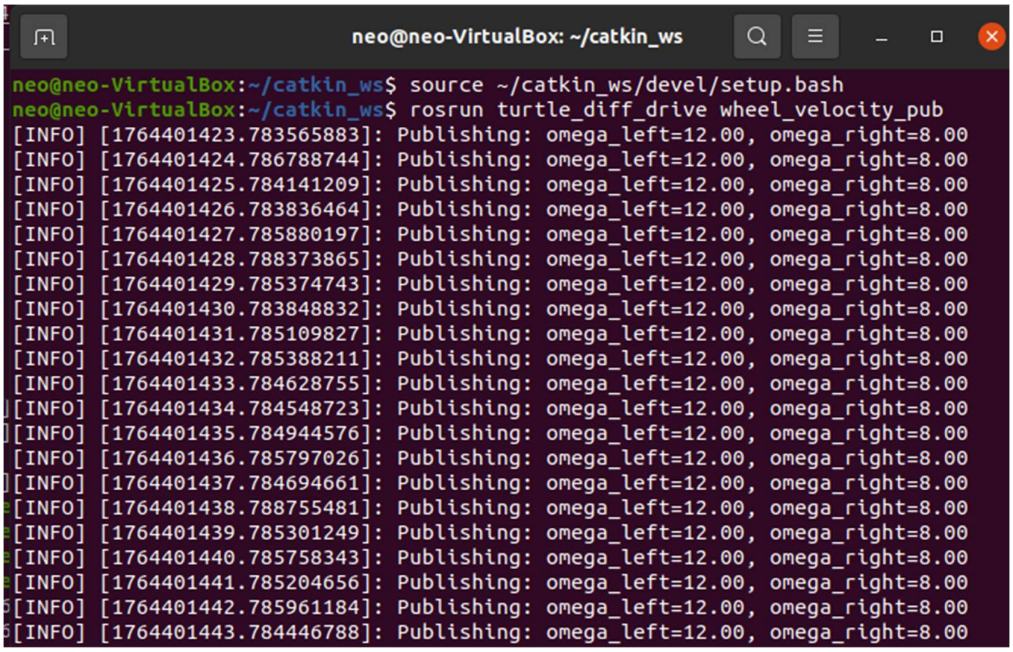


Turtlesim node :

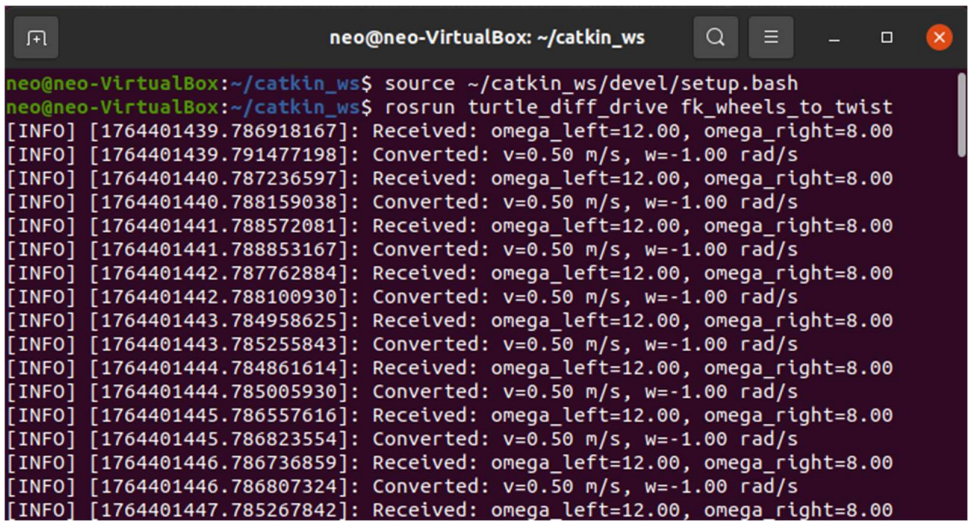




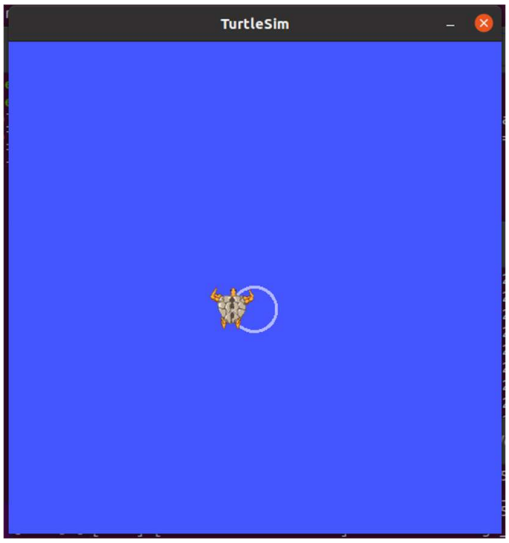
wheel_velocity_pub node:



fk_wheels_to_twist node:



Turtlesim output:

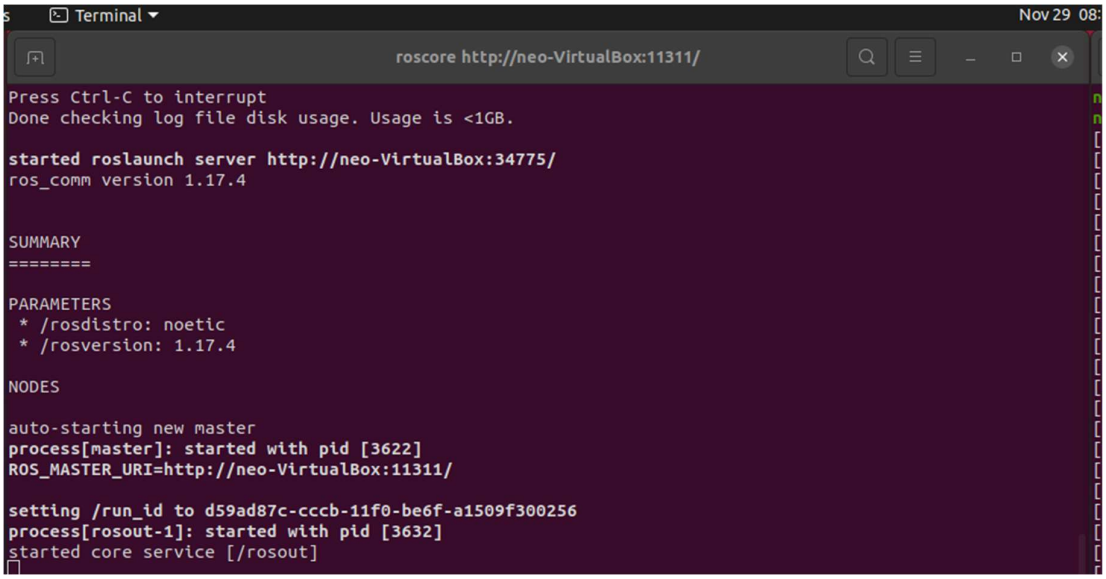


2. go_to_goal_controller.cpp

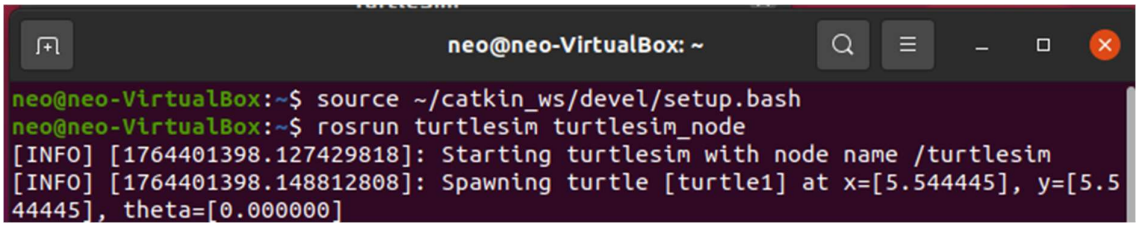
- the go_to_goal_controller.cpp subscribed to the **topic turtle1/pose in which the position of the turtle was continuously published.**
- The message fields of the message type turtlesim/pose were -
    - x
    - y
    - theta
- The goal position was defined by **x_g, y_g, theta_g.**
- **In the** go_to_goal_controller node the Euclidean distance was calculated followed by calculation of the euclidean vector's orientation wrt x axis and final orientation using the formulae.
- The controller gains given were then used to find the linear and angular velocities according to the control law.

Successful navigation to the goal point was demonstrated when x_g, y_g and theta were given values 9,8,0.9 rad resp.

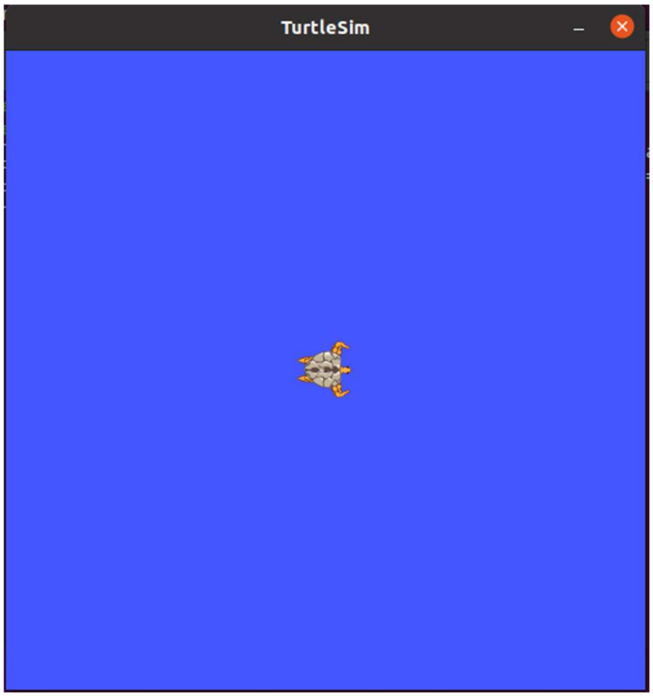
EXECUTION
Roscore



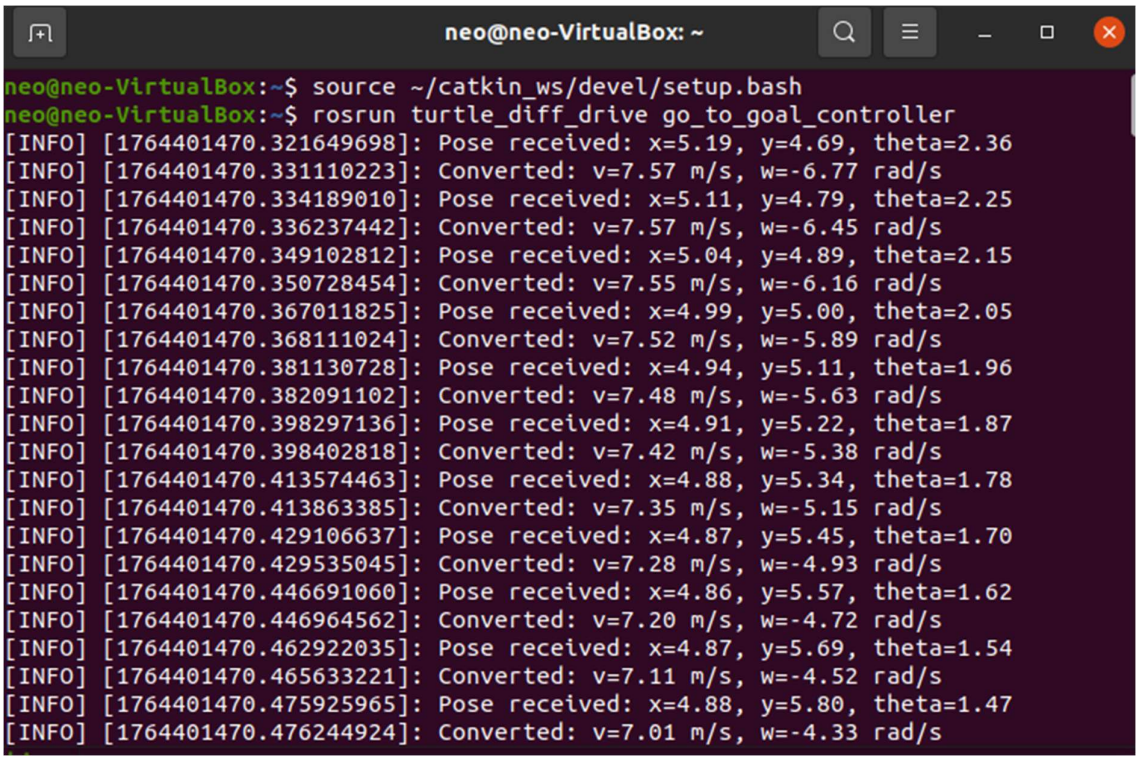
turtlesim_node:

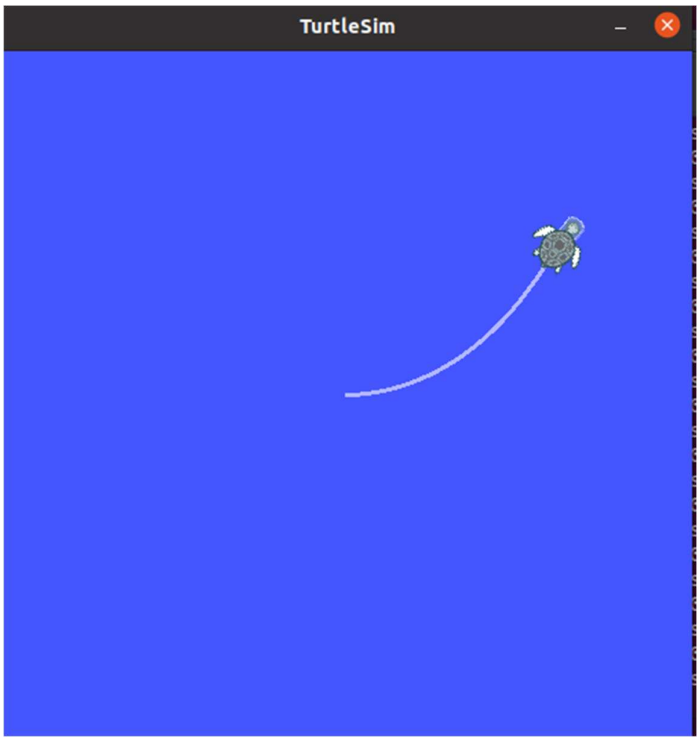


initial orientation of turtle:



go_to_goal_controller node





Go to controller after wheel twist: