ROS Noetic Assignment

Differential Drive Kinematics on Turtlesim

Submitted By: Sharvesh
Department of ECE

Roll No: 2022105538
College: College Of Engineering, Guindy

Table of Contents

1. Introduction

This assignment implements a differential-drive robot controller on the ROS Noetic platform using the Turtlesim simulator. The objective is to:

- Convert wheel angular velocities to linear and angular twist commands (Forward Kinematics).
- Convert robot velocity commands into left and right wheel velocities (Inverse Kinematics).
- Implement a proportional Go-To-Goal controller.
- Verify robot motion using Turtlesim and custom ROS topics.

2. Robot Kinematic Model

The differential-drive robot model relates wheel angular velocities (ωL, ωR) to body velocities (v, ω).

Forward Kinematics:

$v = (r/2) * (\omega R + \omega L)$

$\omega = (r/L) * (\omega R - \omega L)$

Inverse Kinematics:

$\omega R = (1/r)(v + (L/2)*\omega)$

$\omega L = (1/r)(v - (L/2)*\omega)$

Default Parameters:

- Wheel Radius $r = 0.05$ m
- Wheel Base $L = 0.20$ m

3. Forward Kinematics Node

The FK node subscribes to wheel angular velocities and publishes geometry_msgs/Twist to /turtle1/cmd_vel. It implements the forward kinematic equations and ensures the turtle moves according to the wheel inputs.

4. Inverse Kinematics + Go-To-Goal Controller

The Go-To-Goal controller computes linear velocity v and angular velocity ω based on:

ρ = distance to goal

α = heading error

β = orientation error

Control Law:

$v = K\rho * \rho$

$\omega = K\alpha * \alpha + K\beta * \beta$

Recommended Gains:

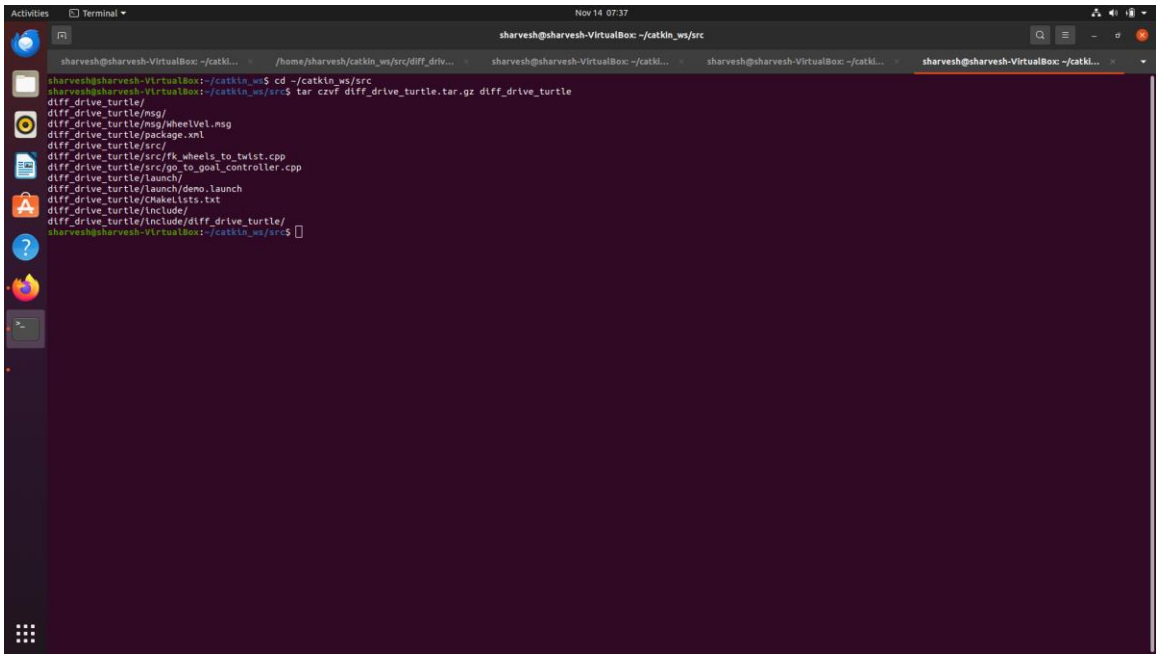
$K\rho = 1.5$, $K\alpha = 4.0$, $K\beta = -1.0$

The node also publishes wheel velocity debug values through /wheel_vel_dbg.

5. Custom Message Definition

msg/WheelVel.msg:
float64 omega_left
float64 omega_right

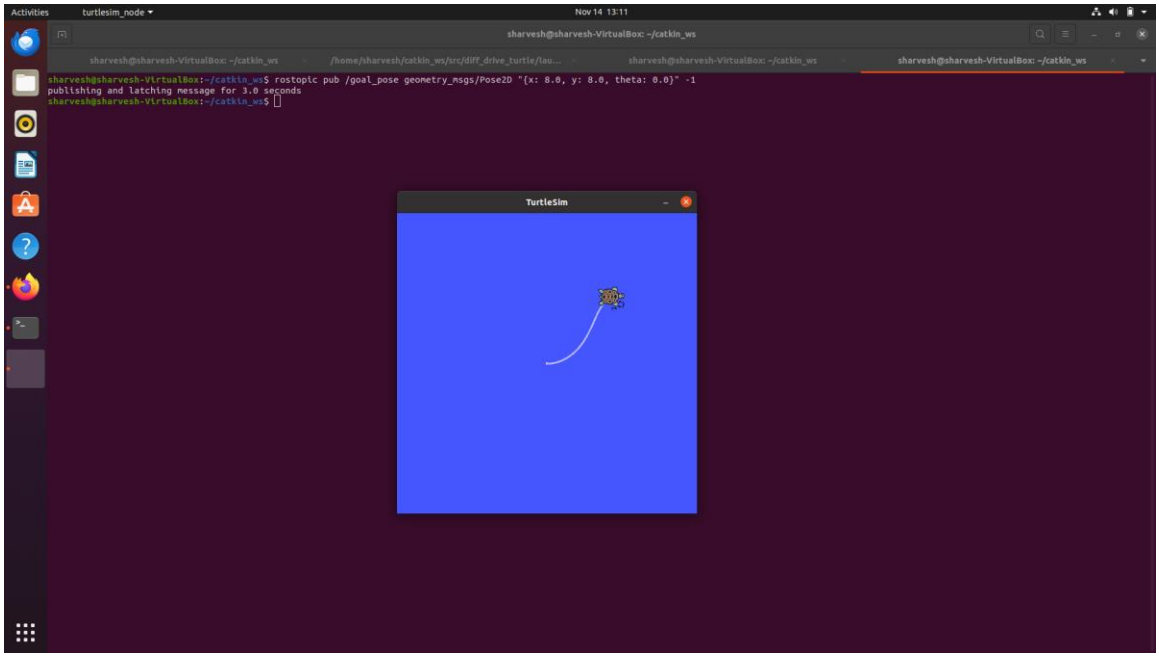
6. Package Structure

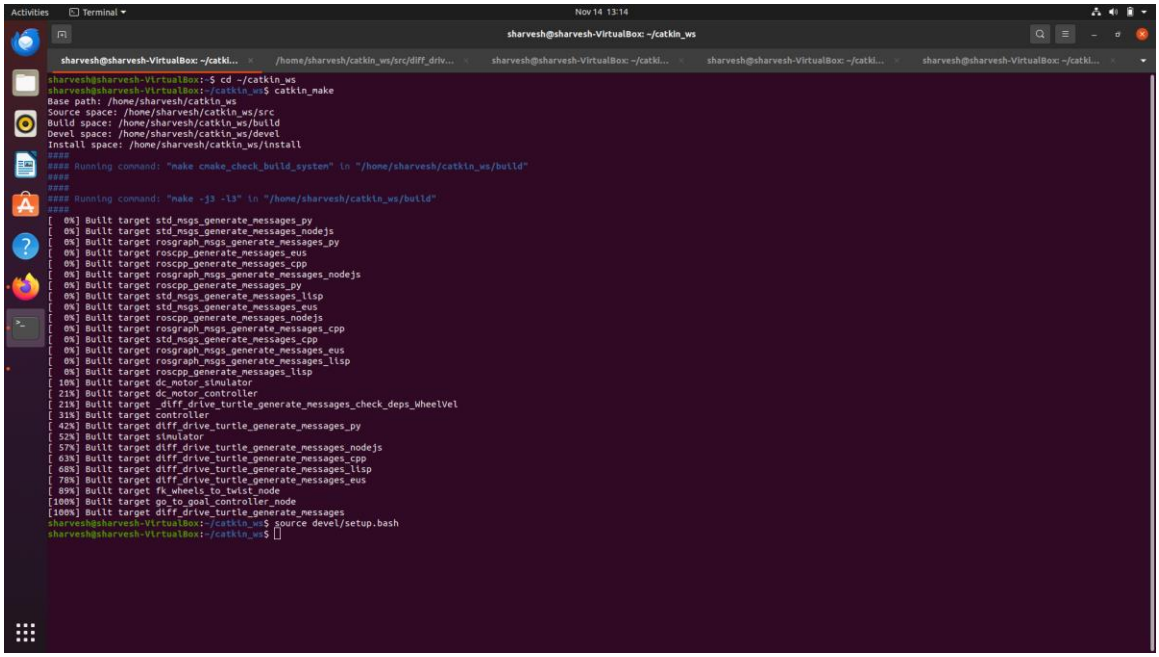


7. Execution Procedure

1. Open terminal and start roscore.
2. Run the diff_drive_turtle demo launch file:
   roslaunch diff_drive_turtle demo.launch
3. Send a goal pose to /goal_pose.
4. Observe robot motion in turtlesim.
5. Echo /wheel_vel_dbg to verify IK output.

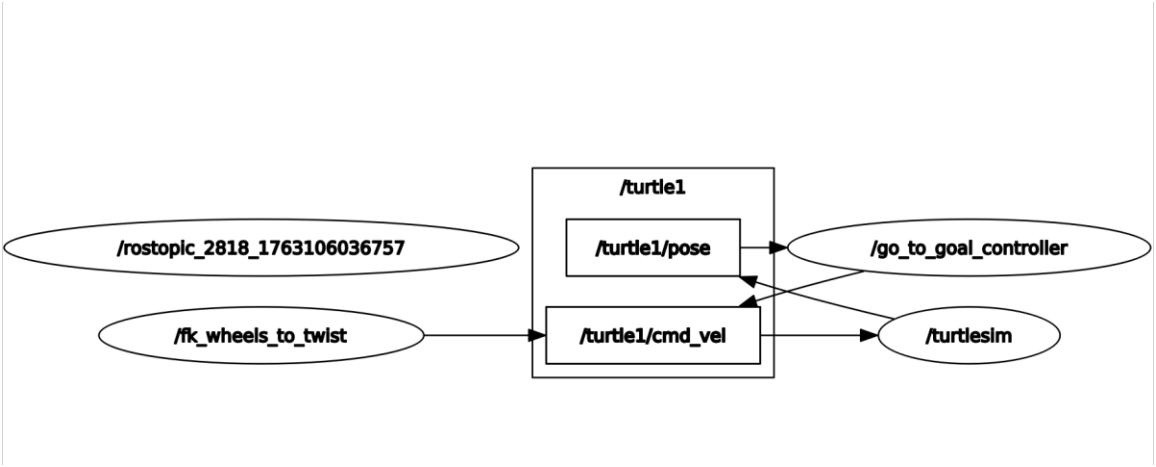
8. Results


Activities
turtlesim_node
Nov 14 13:11
sharvesh@sharvesh-VirtualBox: ~/catkin_ws
sharvesh@sharvesh-VirtualBox:~/catkin_ws$ rostopic pub /goal_pose geometry_msgs/Pose2D "{x: 8.0, y: 8.0, theta: 0.0}" -1
publishing and latching message for 3.0 seconds
sharvesh@sharvesh-VirtualBox:~/catkin_ws$
TurtleSim

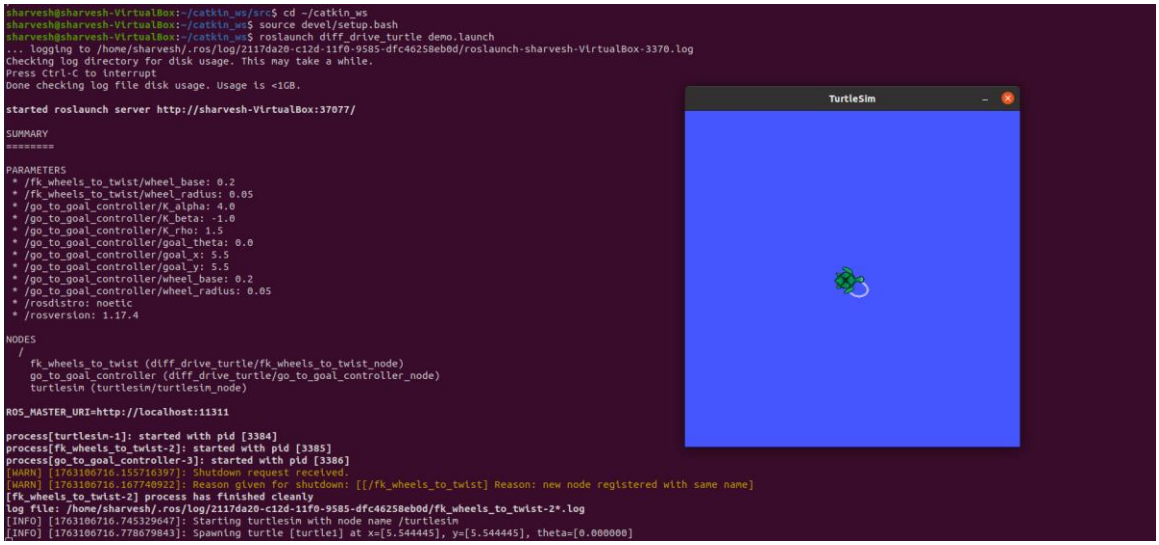

Activities
Terminal
Nov 14 13:14
sharvesh@sharvesh-VirtualBox: ~/catkin_ws
sharvesh@sharvesh-VirtualBox:~$ cd ~/catkin_ws
sharvesh@sharvesh-VirtualBox:~/catkin_ws$ catkin_make
Base path: /home/sharvesh/catkin_ws
Source space: /home/sharvesh/catkin_ws/src
Build space: /home/sharvesh/catkin_ws/build
Devel space: /home/sharvesh/catkin_ws/devel
Install space: /home/sharvesh/catkin_ws/install
Running command: "make cmake_check_build_system" in "/home/sharvesh/catkin_ws/build"
Running command: "make -j3 -l3" in "/home/sharvesh/catkin_ws/build"
[0%] Built target std_msgs_generate_messages_py
[0%] Built target std_msgs_generate_messages_nodejs
[0%] Built target rosgraph_msgs_generate_messages_py
[0%] Built target roscpp_generate_messages_eus
[0%] Built target roscpp_generate_messages_cpp
[0%] Built target rosgraph_msgs_generate_messages_nodejs
[0%] Built target roscpp_generate_messages_py
[0%] Built target std_msgs_generate_messages_lisp
[0%] Built target std_msgs_generate_messages_eus
[0%] Built target roscpp_generate_messages_nodejs
[0%] Built target rosgraph_msgs_generate_messages_cpp
[0%] Built target std_msgs_generate_messages_cpp
[0%] Built target rosgraph_msgs_generate_messages_eus
[0%] Built target rosgraph_msgs_generate_messages_lisp
[0%] Built target roscpp_generate_messages_lisp
[10%] Built target dc_motor_simulator
[21%] Built target dc_motor_controller
[21%] Built target _diff_drive_turtle_generate_messages_check_deps_wheelVel
[31%] Built target controller
[42%] Built target diff_drive_turtle_generate_messages_py
[52%] Built target simulator
[57%] Built target diff_drive_turtle_generate_messages_nodejs
[63%] Built target diff_drive_turtle_generate_messages_cpp
[68%] Built target diff_drive_turtle_generate_messages_lisp
[78%] Built target diff_drive_turtle_generate_messages_eus
[89%] Built target fk_wheels_to_twist_node
[100%] Built target go_to_goal_controller_node
[100%] Built target diff_drive_turtle_generate_messages
sharvesh@sharvesh-VirtualBox:~/catkin_ws$ source devel/setup.bash
sharvesh@sharvesh-VirtualBox:~/catkin_ws$

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sharvesh@sharvesh-VirtualBox:~/catkin_ws$ cd ~/catkin_ws/src
sharvesh@sharvesh-VirtualBox:~/catkin_ws/src$ tree diff_drive_turtle
diff_drive_turtle
├── CMakeLists.txt
├── include
│   └── diff_drive_turtle
├── launch
│   └── demo.launch
├── msg
│   └── WheelVel.msg
├── package.xml
└── src
    ├── fk_wheels_to_twist.cpp
    └── go_to_goal_controller.cpp

5 directories, 6 files
sharvesh@sharvesh-VirtualBox:~/catkin_ws/src$
```

```
sharvesh@sharvesh-VirtualBox:~/catkin_ws$ source ~/catkin_ws/devel/setup.bash
sharvesh@sharvesh-VirtualBox:~/catkin_ws$ rostopic pub /fk_wheels_to_twist/wheel_vel_in diff_drive_turtle/WheelVel "{omega_left: 2.0, omega_right: 2.0}" -r 5

rostopic echo /turtle1/cmd_vel
```




```
^Csharvesh@sharvesh-VirtualBox:~/catkin_ws/src$ rosnode list
/fk_wheels_to_twist
/go_to_goal_controller
/rosout
/rostopic_2818_1763106036757
/turtlesim
sharvesh@sharvesh-VirtualBox:~/catkin_ws/src$
```



9. Conclusion

The assignment successfully demonstrates a complete differential-drive robot simulation using ROS Noetic. Forward kinematics, inverse kinematics, and Go-To-Goal control were implemented and verified through the Turtlesim environment. Custom ROS messages, debugging tools, and launch files were created, forming a fully functional robotics simulation package.