

ECSE 4965 - Intro to Robot Programming

Lab 2

Yifan Wang

Jan 29, 2024

Task 2

Starting nodes

3. Action: Start a simulated turtle ros2 run turtlesim turtlesim_node

4. Question: What is the name and starting pose of this simulated turtle?

```
parallels@ubuntu-linux-22-04-02-desktop:~/Desktop/Parallels Shared Folders/Git-GitKraken/GitKraken-Test-Repo$ ros2 run turtlesim turtlesim_node
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland anyway.
[INFO] [1706405715.178747032] [turtlesim]: Starting turtlesim with node name /turtlesim
[INFO] [1706405715.183258616] [turtlesim]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
libGL error: pci id for fd 34: 1ab8:0010, driver (null)
pci id for fd 35: 1ab8:0010, driver (null)
```

Figure 2.1 - Step 4 Screenshot

Name: turtle1

Starting position: x=[5.544445], y=[5.544445], theta=[0.000000]

7. Action: Open a new terminal. Position the terminals and simulation window such that you can see all three simultaneously. Start a node to interact with the turtle. `ros2 run turtlesim turtle_teleop_key`

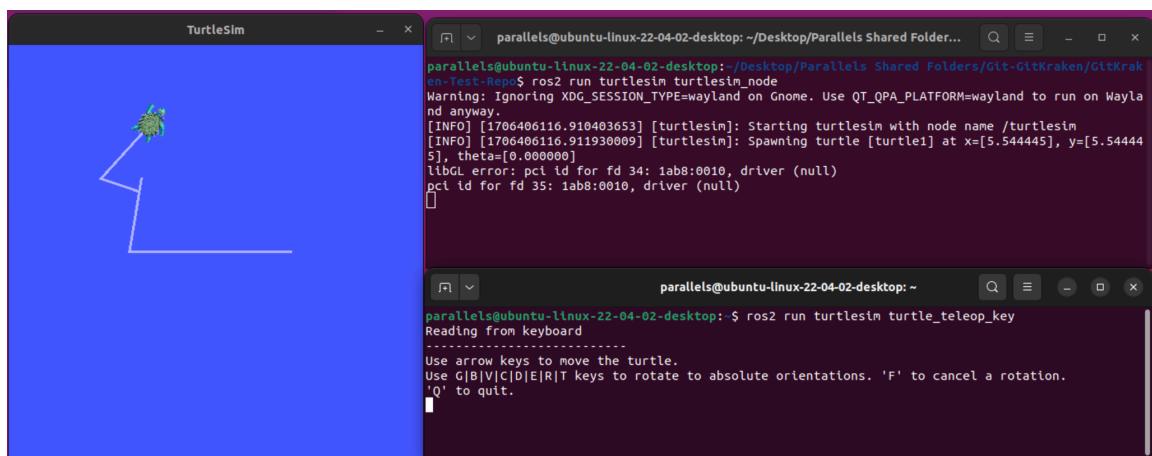


Figure 2.2 - Step 7 Screenshot

9. Question: What happens in the turtlesim window if you try to drive the turtle out of the window?

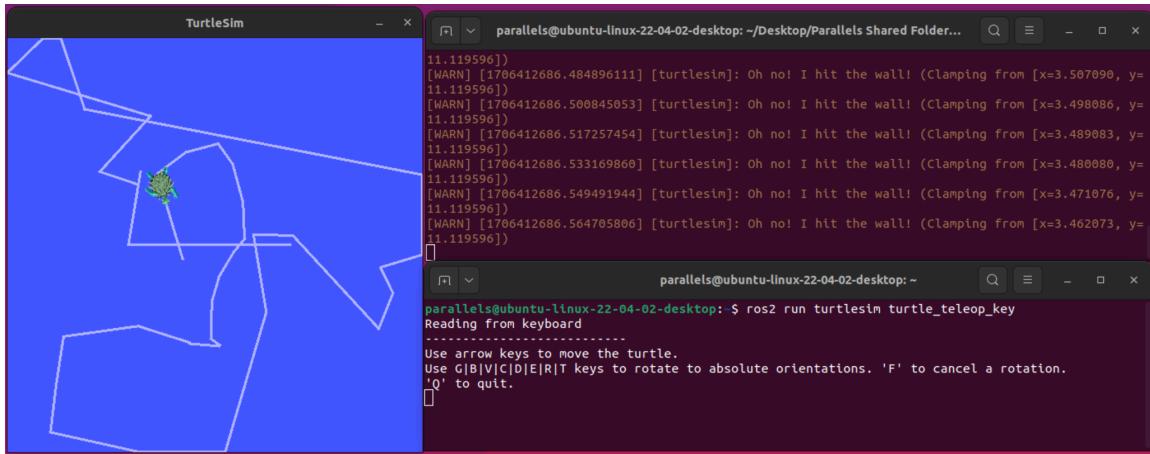


Figure 2.3 - Step 9 Screenshot

Gives a warning: [WARN] [1706406247.065365108] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=-0.027782, y=0.928784])

Topics

7 Action: Type `rqt_graph`. After the window opens, in the drop-down menu on the left select 'Nodes/Topics (all)' and ensure all the check boxes are selected except 'Dead Sinks' and 'Leaf Topics'.

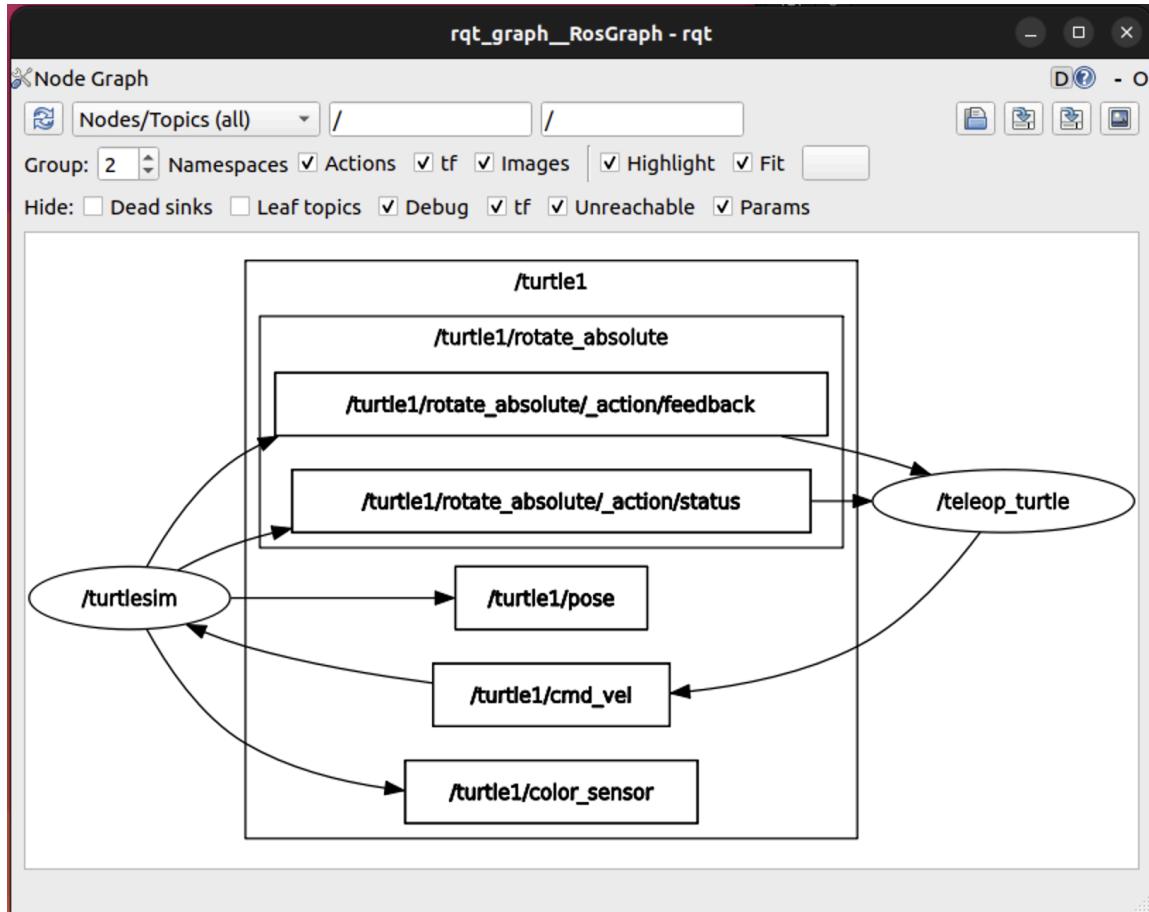


Figure 2.4 - Step 7 Screenshot

21. Action: use topic info, interface show, and topic echo to find similar information for the topic `/turtle1/pose`

Topic info

```
parallels@ubuntu-linux-22-04-02-desktop:~$ ros2 topic info /turtle1/pose
Type: turtlesim/msg/Pose
Publisher count: 1
Subscription count: 0
```

Figure 2.5 - Step 21 Screenshot

Interface show

```
parallels@ubuntu-linux-22-04-02-desktop:~$ ros2 interface show turtlesim/msg/Pose
float32 x
float32 y
float32 theta

float32 linear_velocity
float32 angular_velocity
```

Figure 2.6 - Step 21 Screenshot

Topic echo

```
parallels@ubuntu-linux-22-04-02-desktop:~$ ros2 topic echo /turtle1/pose
x: 4.4932403564453125
y: 3.2572176456451416
theta: 0.06719999760389328
linear_velocity: 0.0
angular_velocity: 0.0
---
x: 4.4932403564453125
y: 3.2572176456451416
theta: 0.06719999760389328
linear_velocity: 0.0
angular_velocity: 0.0
---
```

Figure 2.7 - Step 21 Screenshot

22. Question: With the pose topic notice that with echo we are getting continuous data. At what rate is the simulator publishing? (Hint: topic hz)

```
^Cparallels@ubuntu-linux-22-04-02-desktop:~$ ros2 topic hz /turtle1/pose
average rate: 62.356
    min: 0.008s max: 0.021s std dev: 0.00229s window: 64
average rate: 62.491
    min: 0.007s max: 0.021s std dev: 0.00252s window: 127
average rate: 62.509
    min: 0.007s max: 0.026s std dev: 0.00262s window: 190
average rate: 62.457
    min: 0.007s max: 0.026s std dev: 0.00243s window: 253
```

Figure 2.8 - Step 22 Screenshot

25. Action: Instead of using the teleop window, we can directly publish to the `/turtle1/cmd_vel` topic. Type

```
ros2 topic pub --once /turtle1/cmd_vel  
geometry_msgs/msg/Twist "{linear: {x: 2.0, y: 0.0, z: 0.0},  
angular: {x: 0.0, y: 0.0, z: 1.8}}"
```

Try changing these numbers and performing some executions.

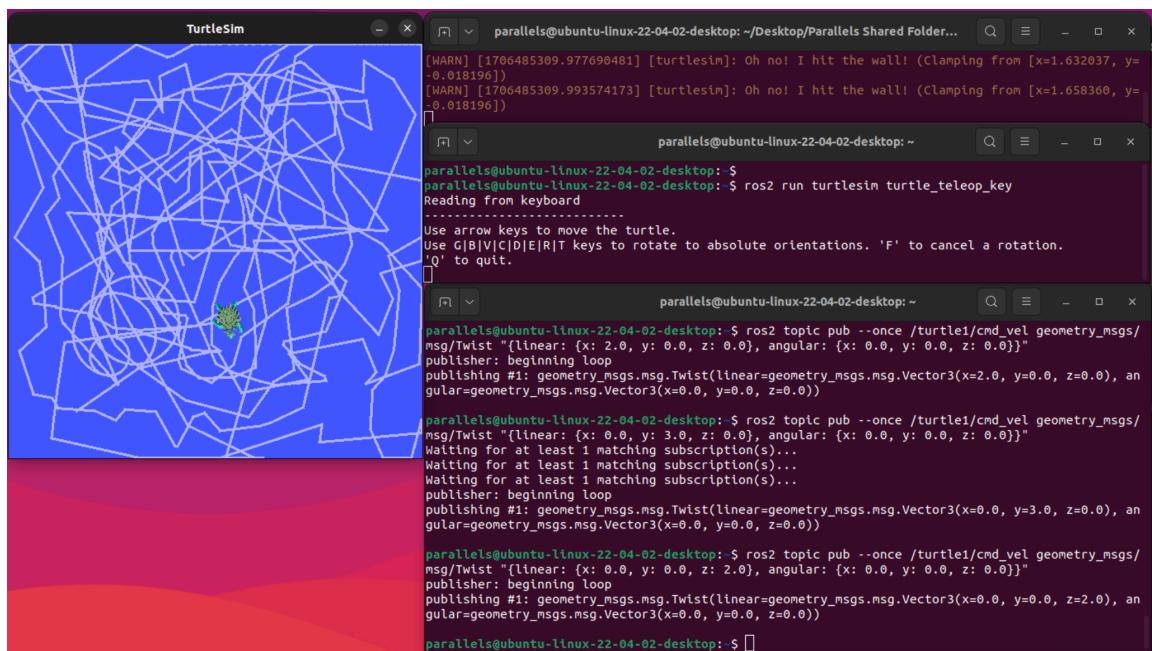


Figure 2.9 - Step 25 Screenshot

Services

Challenge: Based on what we have done so far, change the turtle line color to red.

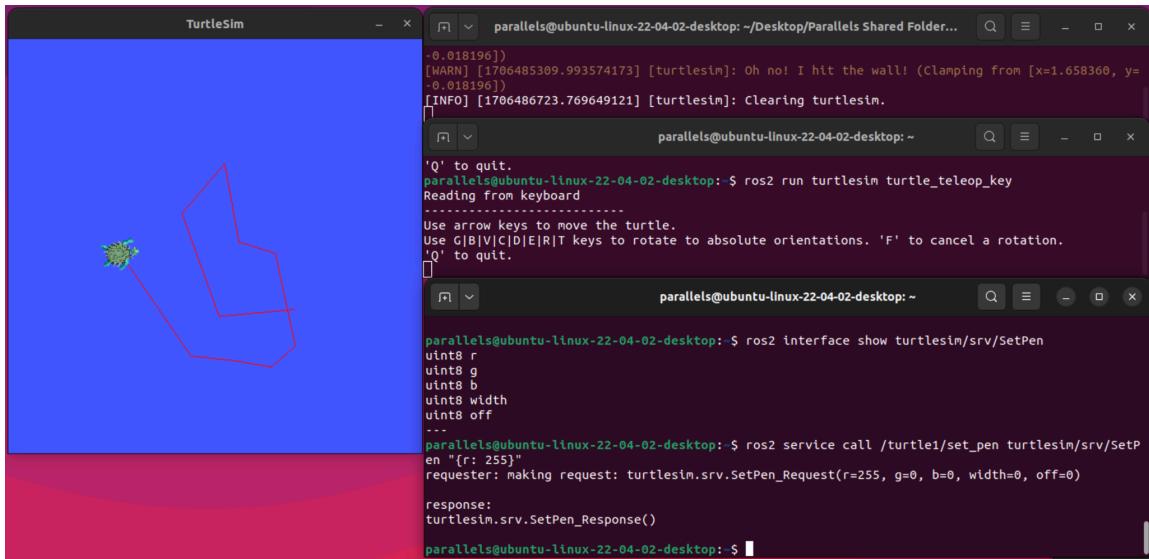


Figure 2.10 - Challenge Screenshot

Actions

7. Action: Let's send a goal to this action.

```
ros2 action send_goal /turtle1/rotate_absolute
turtlesim/action/RotateAbsolute "{theta: 1.57}"
```

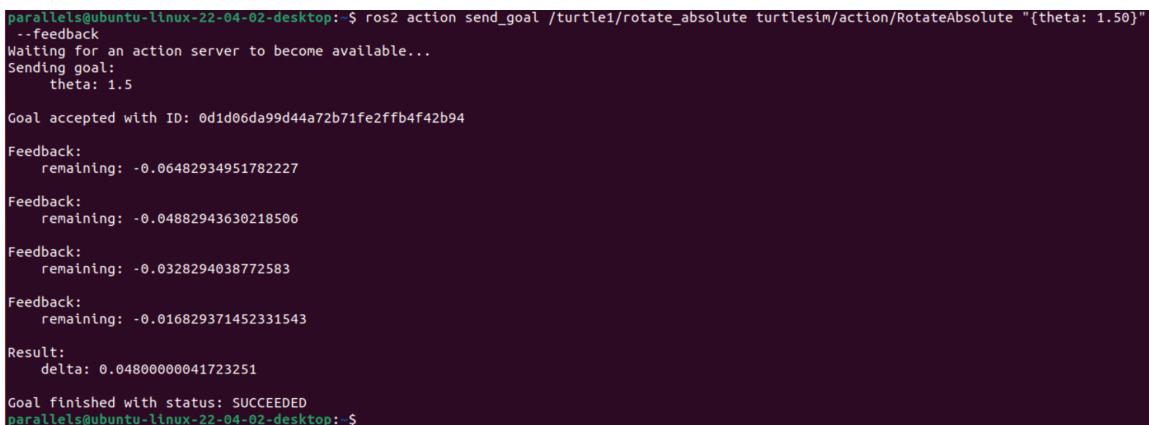


Figure 2.11 - Step 7 Screenshot

8. Action: Rerun this command with the addition of --feedback at the end to see the feedback.

```
parallels@ubuntu-linux-22-04-02-desktop:~$ ros2 action list -t
/turtle1/rotate_absolute [turtlesim/action/RotateAbsolute]
parallels@ubuntu-linux-22-04-02-desktop:~$ ros2 interface show turtlesim/action/RotateAbsolute
# The desired heading in radians
float32 theta
---
# The angular displacement in radians to the starting position
float32 delta
---
# The remaining rotation in radians
float32 remaining
parallels@ubuntu-linux-22-04-02-desktop:~$ 
```

Figure 2.12 - Step 8 Screenshot

Task 3

1. Action: Using what we have learned, draw a black square on the screen.

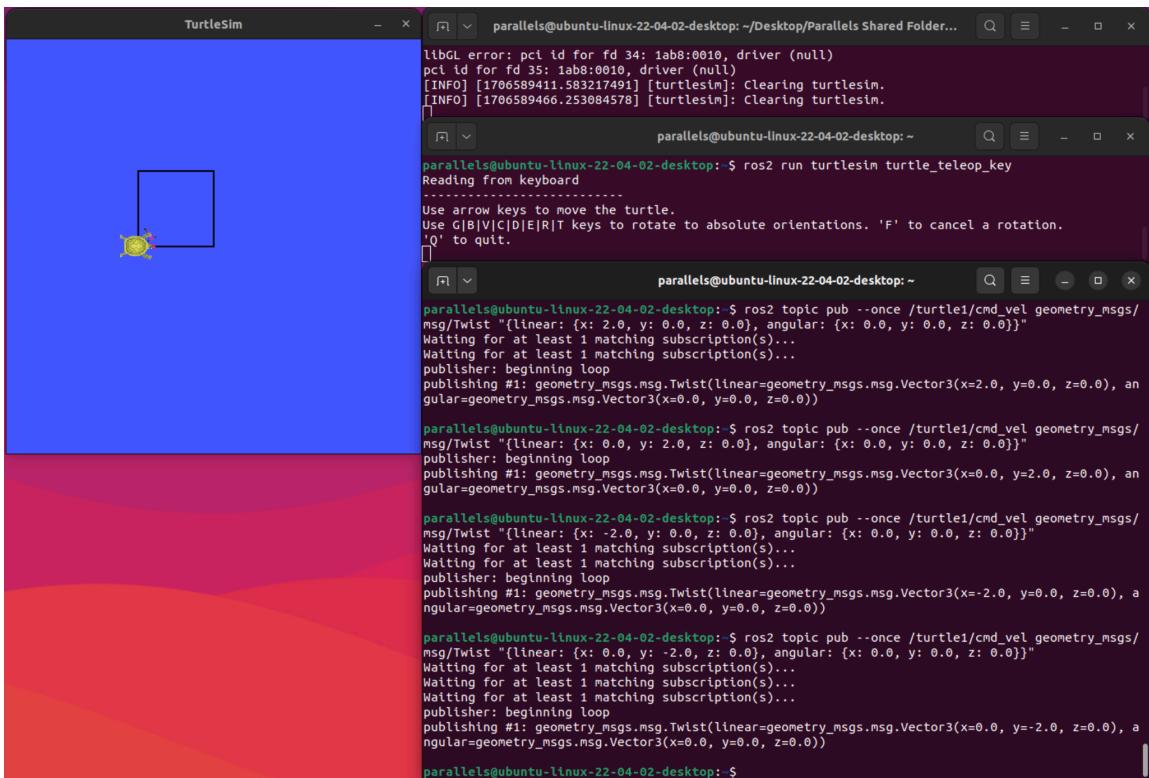


Figure 3.1 - turtlesim Black Square

One way to do this is by using the following steps:

```
// Clear space
ros2 service call /clear std_srvs/srv/Empty

// Set pen color to black, width to 2
ros2 service call /turtle1/set_pen turtlesim/srv/SetPen "{r: 0, g: 0, b: 0,
width: 2}"

// Draw the first horizontal side
ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x:
2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"

// Draw the first vertical side
ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x:
0.0, y: 2.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"

// Draw the second horizontal side
ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x:
-2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"

// Draw the second vertical side
os2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x:
0.0, y: -2.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
```