

```

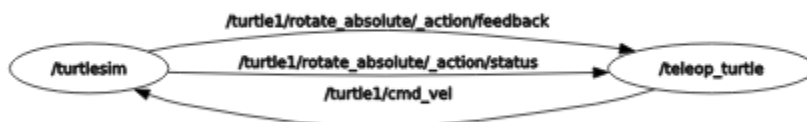
Action Clients:
/turtle1/rotate_absolute: turtlesim/action/RotateAbsolute
yahboom@VM:~$ sqrt_graph
sqrt_graph: command not found
yahboom@VM:~$ rqt_graph
WARNING: Package name "yahboomcar_KCFTracker" does not follow the naming conventions. It should start with a lower case letter and only contain lower case letters, digits, underscores, and dashes.
WARNING: Package name "yahboomcar_KCFTracker" does not follow the naming conventions. It should start with a lower case letter and only contain lower case letters, digits, underscores, and dashes.
yahboom@VM:~$ rqt_graph
yahboom@VM:~$ ros2 topic list
/parameter_events
/rosout
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
yahboom@VM:~$ ros2 topic info /turtle1/cmd_vel
Type: geometry_msgs/msg/Twist
Publisher count: 1
Subscription count: 1
yahboom@VM:~$

```

```

yahboom@VM:~$ ros2 interface show geometry_msgs/msg/Twist
# This expresses velocity in free space broken into its linear and angular parts
.
Vector3 linear
Vector3 angular
yahboom@VM:~$ ros2 topic echo /turtle1/cmd_vel
WARNING: topic [/turtle1/cmd_vel] does not appear to be published yet
Could not determine the type for the passed topic

```



```

0l ---
- x: 1.7465063333511353
k y: 9.178534507751465
C theta: 1.1200001239776611
t linear_velocity: 0.0
angular_velocity: 0.0
---
x: 1.7465063333511353
y: 9.178534507751465
theta: 1.1200001239776611
linear_velocity: 0.0
angular_velocity: 0.0
---
x: 1.7465063333511353
1 y: 9.178534507751465
3 theta: 1.1200001239776611
1 linear_velocity: 0.0
3 angular_velocity: 0.0
1 ---
3 x: 1.7465063333511353
y: 9.178534507751465

```

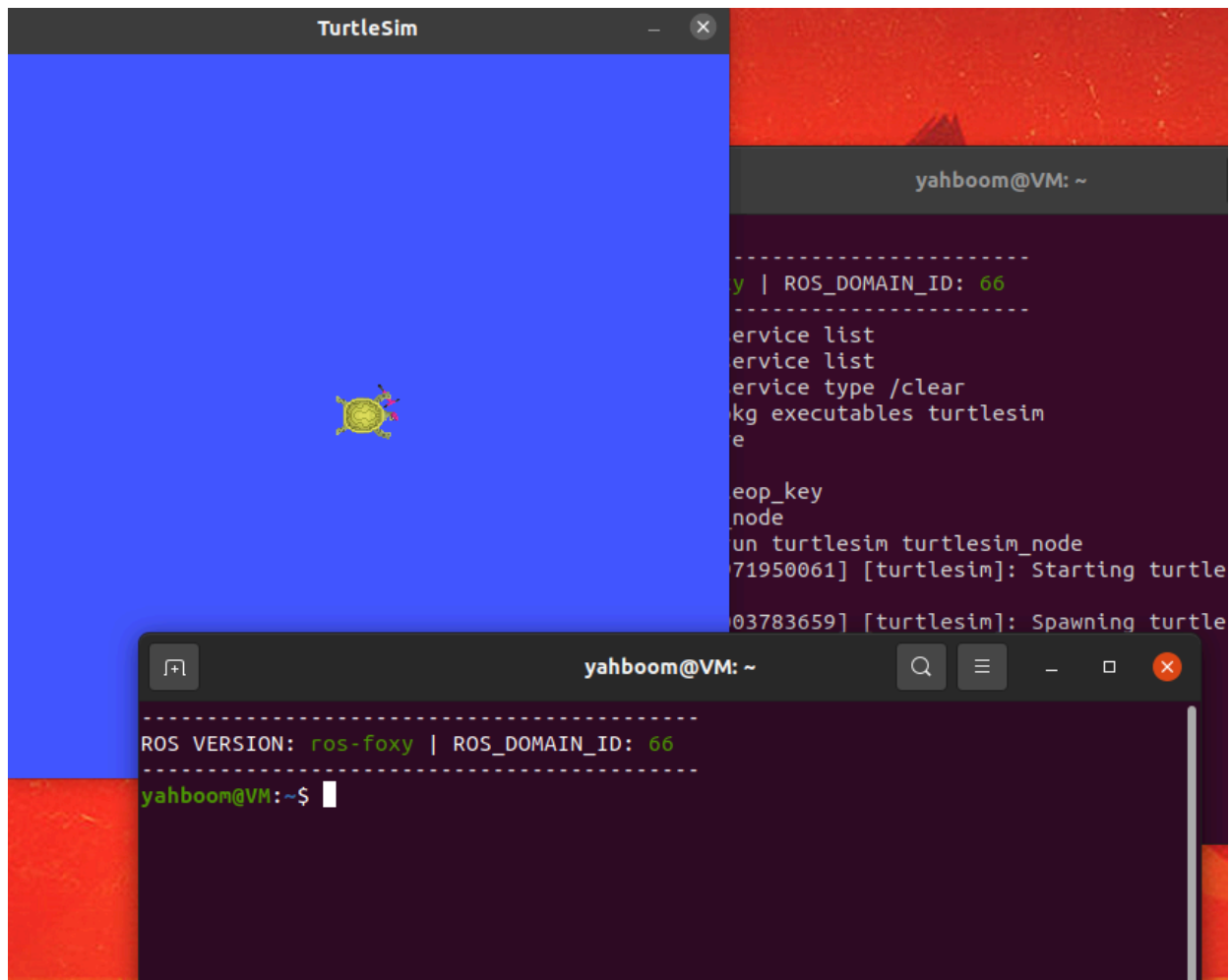
```

^Cyahboom@VM:~$ ros2 topic info /turtle1/pose
Type: turtlesim/msg/Pose
Publisher count: 1
Subscription count: 0
yahboom@VM:~$ ros2 interface show turtlesim/msg/Pose
float32 x
float32 y
float32 theta

float32 linear_velocity
float32 angular_velocity

```

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}}"



```

ros2 topic pub: error: argument -r/--rate: invalid positive_float value: '
/turtle1/cmd_vel'
yahboom@VM:~$ ros2 topic pub --rate 1 /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}}"
publisher: beginning loop
publishing #1: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=1.8))
publishing #2: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=1.8))
publishing #3: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=1.8))
publishing #4: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=1.8))

```

```

yahboom@VM:~$ 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: 5.0, z: 9.8}}"
publisher: beginning loop
publishing #1: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=5.0, z=9.8))

```

```

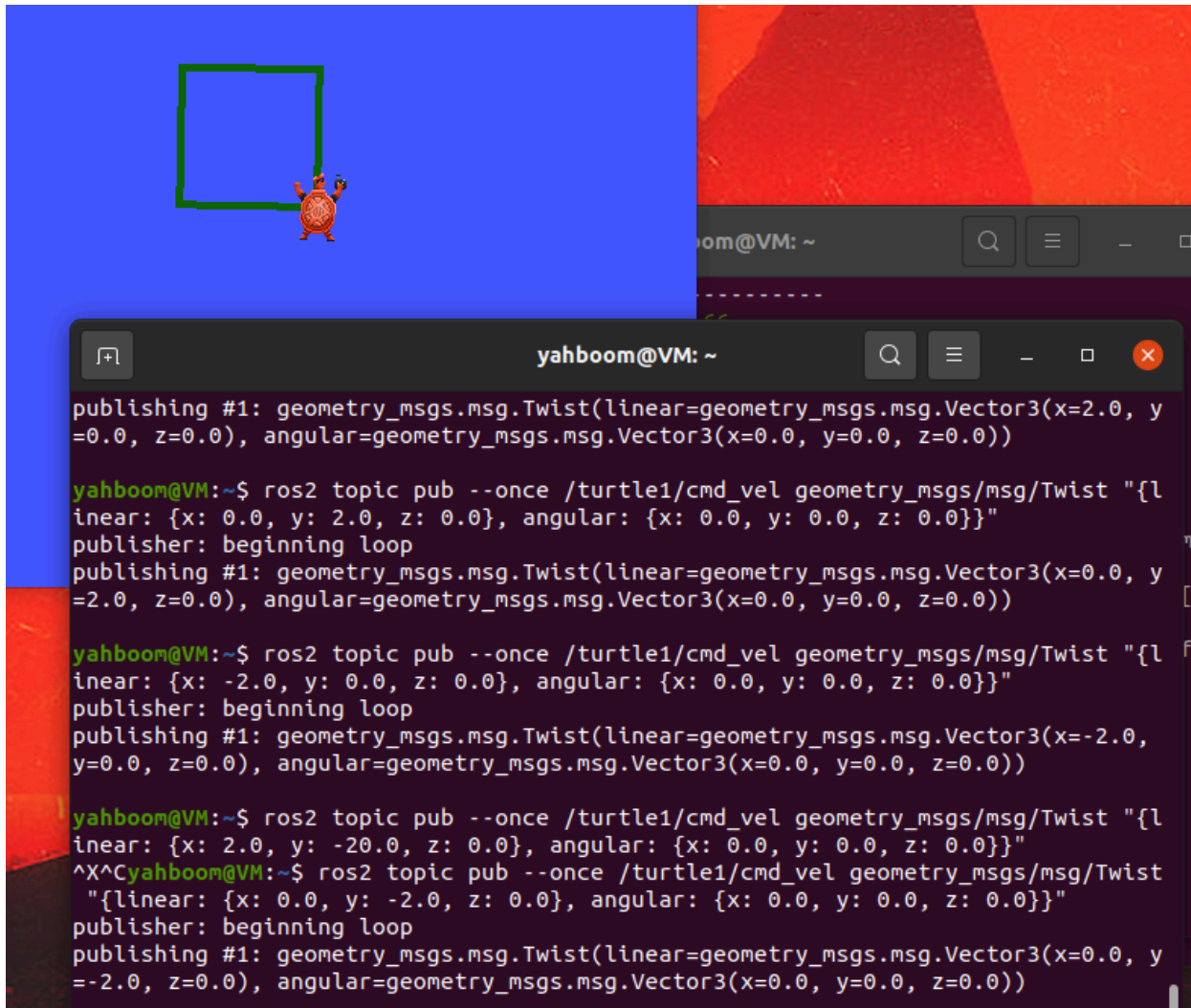
/turtlesim/set_parameters_atomically
yahboom@VM:~$ ros2 service list -t
/clear [std_srvs/srv/Empty]
/kill [turtlesim/srv/Kill]
/reset [std_srvs/srv/Empty]
/spawn [turtlesim/srv/Spawn]
/teleop_turtle/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/teleop_turtle/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/teleop_turtle/get_parameters [rcl_interfaces/srv/GetParameters]
/teleop_turtle/list_parameters [rcl_interfaces/srv/ListParameters]
/teleop_turtle/set_parameters [rcl_interfaces/srv/SetParameters]
/teleop_turtle/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
/turtle1/set_pen [turtlesim/srv/SetPen]
/turtle1/teleport_absolute [turtlesim/srv/TeleportAbsolute]
/turtle1/teleport_relative [turtlesim/srv/TeleportRelative]
/turtlesim/describe_parameters [rcl_interfaces/srv/DescribeParameters]
/turtlesim/get_parameter_types [rcl_interfaces/srv/GetParameterTypes]
/turtlesim/get_parameters [rcl_interfaces/srv/GetParameters]
/turtlesim/list_parameters [rcl_interfaces/srv/ListParameters]
/turtlesim/set_parameters [rcl_interfaces/srv/SetParameters]
/turtlesim/set_parameters_atomically [rcl_interfaces/srv/SetParametersAtomically]
]
yahboom@VM:~$

```

ros2 service call /turtle1/set_pen turtlesim/srv/SetPen "{ 'r': 10, 'g': 100, 'b': 0, 'width': 5, 'off': 0 }"



```
yahboom@VM: ~  
yahboom@VM:~$ os2 service call /turtle1/set_pen turtlesim/srv/SetPen "{r': 10,  
'g': 100, 'b': 0, 'width': 5, 'off': 0}"  
Command 'os2' not found, did you mean:  
  
  command 'osm' from snap osmclient (v11.0.0rc1-32-g6b9ac93)  
  command 'os8' from deb os8 (2.1-7)  
  command 'osc' from deb osc (0.167.1-1)  
  command 'osh' from deb omake (0.9.8.5-3-10)  
  command 'osx' from deb opensp (1.5.2-13ubuntu3)  
  
See 'snap info <snapname>' for additional versions.  
yahboom@VM:~$ ros2 service call /turtle1/set_pen turtlesim/srv/SetPen "{r': 10,  
'g': 100, 'b': 0, 'width': 5, 'off': 0}"  
waiting for service to become available...  
requester: making request: turtlesim.srv.SetPen_Request(r=10, g=100, b=0, width=  
5, off=0)  
  
response:  
turtlesim.srv.SetPen_Response()
```



Move the turtle to draw a square: Next, you would command the turtle to move in a way that would draw a square. This would involve publishing messages to the `/turtle1/cmd_vel` topic with specific linear and angular velocities.

A simple method to draw a square would involve moving the turtle forward a certain distance, then rotating it 90 degrees, and repeating these steps four times. However, because ROS topics typically work with velocities rather than distances or angles, you would need to control the time for which you publish the velocities to estimate the distance and angle turned.