

Taller 1 – Robótica

Fernanda Tarquino –201815447

Santiago Rueda – 201910506

Tony Montes – 202014562

Sebastian Guerrero – 202021249

Punto 1

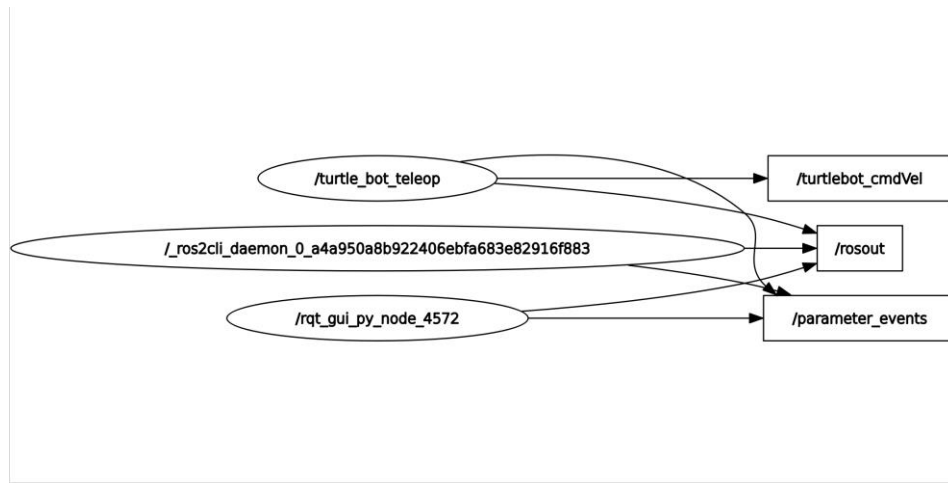


Ilustración 1 Grafo punto 1

En la ilustración 1 se puede observar el grafo del punto 1, en este se puede observar que se encuentra activo el nodo `/turtle_bot_teleop`, y que este nodo se encuentra publicando en el tópico `/turtlebot_cmdVel`, como se pedía en la guía, en el tópico `/turtlebot_cmdVel` publica las velocidades lineal y angular, y el desplazamiento requerido al robot. El tópico `/rosout` y `/parameter_events` son tópicos que Ros2 obliga a publicar a todas las librerías. Esto se logra gracias a la librería `pynput`, la cual cuenta con una función `OnPress`, la cual es usada para detectar cuando se oprimen las teclas. Se utilizan ifs para diferenciar que teclas hacen que.

```
robotica@robotica-VirtualBox:~/Downloads/robotica$ ros2 run taller1 turtle_bot_teleop
Ingrese la velocidad lineal: 4
Ingrese la velocidad angular: 5
[INFO] [1677717634.143074840] [turtle_bot_teleop]: Publishing: Adelante
[INFO] [1677717634.263713520] [turtle_bot_teleop]: Publishing: Stop
[INFO] [1677717634.433265559] [turtle_bot_teleop]: Publishing: Atras
[INFO] [1677717634.534014338] [turtle_bot_teleop]: Publishing: Stop
[INFO] [1677717634.691085907] [turtle_bot_teleop]: Publishing: Izquierda
[INFO] [1677717634.828655174] [turtle_bot_teleop]: Publishing: Stop
[INFO] [1677717634.972161705] [turtle_bot_teleop]: Publishing: Derecha
[INFO] [1677717635.063262125] [turtle_bot_teleop]: Publishing: Stop
```

Ilustración 2 Funcionamiento nodo

En la ilustración 2 se puede observar el correcto funcionamiento del nodo `/turtle_bot_teleop`, donde en primer lugar se pide al usuario ingresar la velocidad lineal, luego la angular, y finalmente se ve la publicación de los movimientos que se realizar.

Punto 2

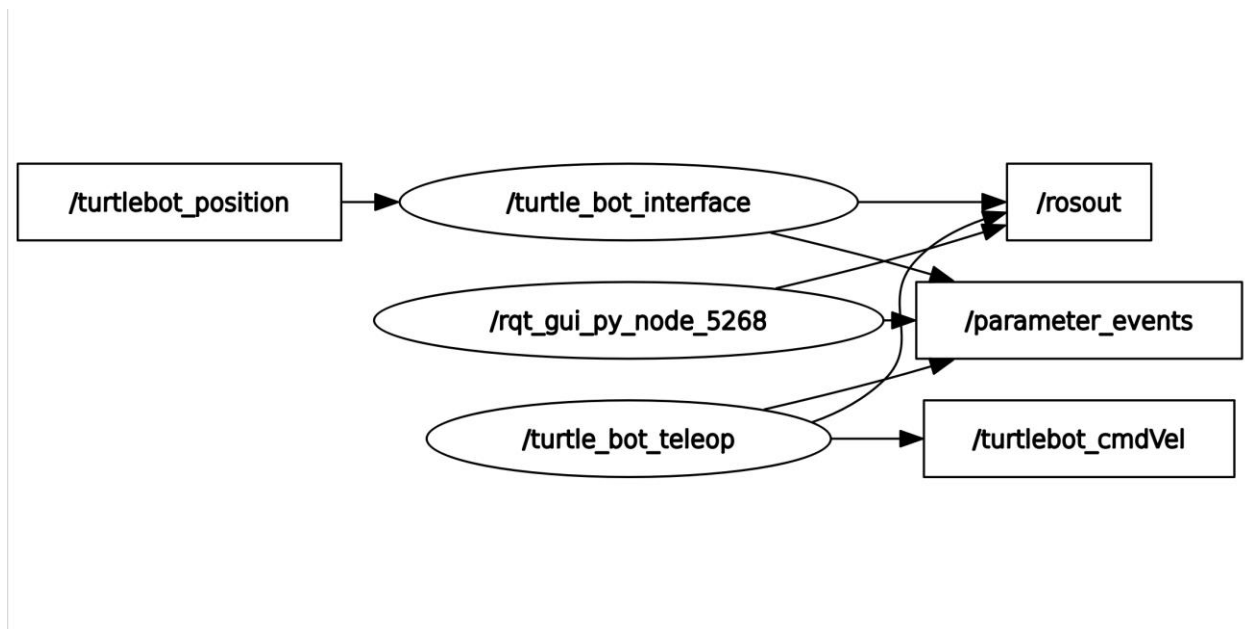


Ilustración 3 Grafo punto 2

En la ilustración 3 se puede observar que como sigue activo el nodo `/turtle_bot_teleop` este sigue publicando en los tópicos `/turtlebot_cmdVel`, `/parameter_event`, y `/rosout`. Pero para desarrollar el punto 2 del taller se tuvo que crear un nodo que se suscribiera al tópico `/turtlebot_position` para así poder trazar la trayectoria del robot, así mismo este se encuentra publicando en los tópicos `/parameter_events` y `/rosout` debido a que Ros2 lo hace publicar en estos tópicos.

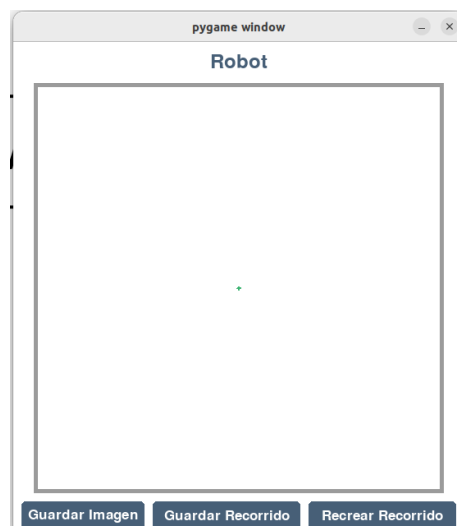


Ilustración 4 Grafica trayectoria robot

Para poder realizar la gráfica se utilizó pygames, para así poder crear la interfaz y los botones que se encuentran en la parte inferior. A medida que se realizan movimientos se puede observar en la ilustración 5 como se va trazando esta trayectoria.

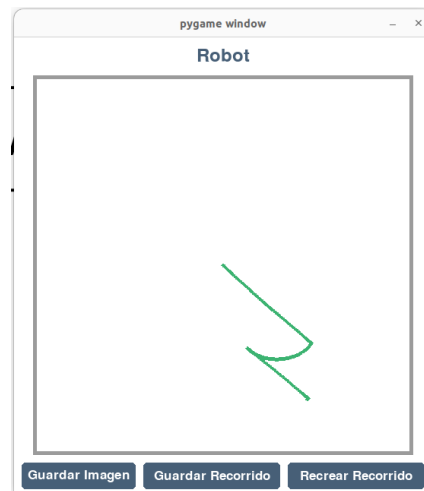


Ilustración 5 Trayectoria realizada por robot

Así mismo se agregó el botón guardar imagen para así poder guardar la grafica creada, una vez se presiona el botón se puede observar la interfaz presente en la Ilustración 6 donde se le pregunta al usuario donde quiere guardar la gráfica, y el nombre que se le quiere poner al archivo.

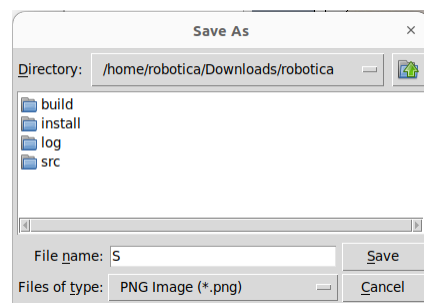


Ilustración 6 Interfaz para guardar grafica

Punto 3

Para el punto 3 al realizar el comando `rqt_graph` se obtiene el mismo grafo que se encuentra en la ilustración 3 esto se debe a que para este punto no hubo necesidad de crear nodos, lo que se realizo fue crear el servicio `save_motion` y modificar los nodos ya existentes `/turtle_bot_teleop` y `/turtle_bot_interface`

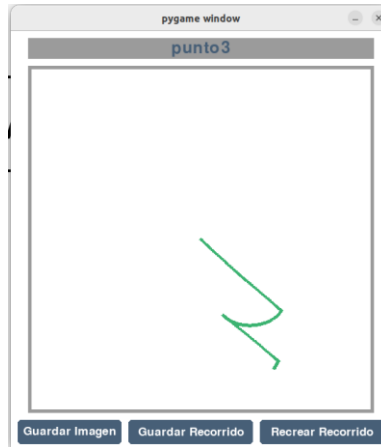


Ilustración 7 Opción guardar recorrido

Para la solución del punto 3 se agregó un botón en la interfaz creada en el punto 2 la cual se conecta al servicio `save_motion` el cual le pide al nodo `/turtle_bot_teleop` que escriba el recorrido con el nombre dado a la gráfica. El archivo TXT siempre se guardará en la carpeta `"/taller1/motion/"`, por ende, siempre se tiene que correr el nodo en la carpeta raíz del workspace. Por ejemplo, en la Ilustración 7, el nombre de la gráfica es "punto3" (este nombre se puede modificar desde la interfaz) y por ende se guardará como `punto3.txt`

Decidimos implementar que todos los archivos TXT se guardaran en la misma carpeta para facilitarle al usuario reproducir los recorridos (punto 4), ya que solo tendrá que especificar el nombre del recorrido a replicar, en vez de tener que especificar toda la ubicación del archivo TXT.

```
[INFO] [1677718619.663480179] [turtle_bot_interface]: Guardar Recorrido
[INFO] [1677718619.664555129] [turtle_bot_interface]: Calling service...
```

Ilustración 8 Conexión nodo `/turtle_bot_interface` con el servicio `save_motion`

```
[INFO] [1677718619.665717162] [turtle_bot_teleop]: Writing to file: /home/robotica/Downloads/robotica/src/taller1/motion/punto3.txt
```

Ilustración 9 nodo `/turtle_bot_teleop` escribiendo en la dirección

Es importante recalcar que para la creación del servicio `save_motion` se requería crear un nuevo tipo de servicio; este tipo de servicio (que denominamos `SaveMotions.srv`) tuvo que ser creado dentro de un nuevo paquete `'my_msgs'` ya que por la organización de ROS2, no se permitía importar desde el mismo paquete; es por esto que para que este servicio y el del punto 4 funcionen, es necesario construir el paquete `'my_msgs'` previamente.

Punto 4

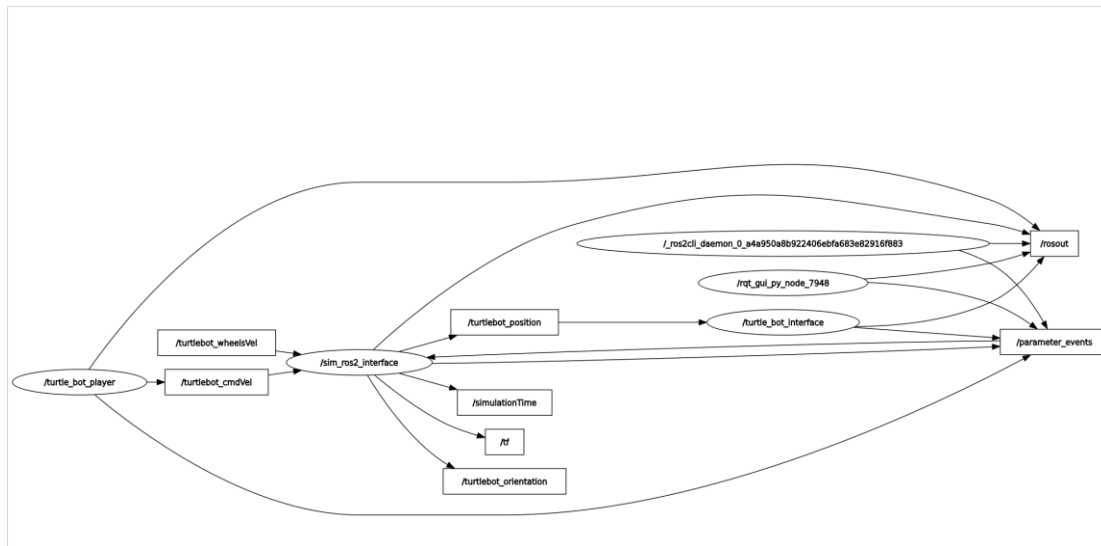


Ilustración 10 Grafo punto 4

Como se puede observar en la Ilustración 10 se puede observar que al correr el nodo /turtle_bot_platyer este publica en el tópicos /turtlebot_cmdVel que es el tópicos que permite mover el robot enviándole la velocidad lineal y angular con la que se desplaza.

Para poder replicar el recorrido del robot, debimos tener en cuenta varias cosas; inicialmente, cuál sería la estructura del archivo TXT que se guardaría de un recorrido para luego replicarlo, lo que decidimos en esta instancia fue guardar 3 cosas principalmente:

1. En la línea 0: Las velocidades lineal y angular del robot.
2. En todas las demás líneas: w,a,s o d por cada vez que se presionara (o mantuviera presionada) una tecla en el recorrido.
3. El tiempo que dura detenido el robot entre una tecla y la otra cada vez que hay una pausa.

De esta forma replicaríamos el movimiento ejecutado de la forma más precisa. A continuación, se observa el formato de dicho archivo TXT:

```

1 4.0,2.0
2 w
3 w
4 w
5 w
6 w
7 w
8 p=3.099064826965332
9 d
10 d
11 d
12 d
13 d
14 d
15 d
16 d
17 d
18 d

```

Para la creación del servicio y la publicación de las velocidades en el robot, se definió un nuevo nodo /turtle_bot_player; este nodo define el servicio 'recreate_motion' que va publicando las velocidades y

esperando cuando tiene que esperar. Además, se definió un botón en la interfaz para recrear un movimiento, dependiendo del título de la gráfica que se coloque, ese será el movimiento a recrear.

Para poder llamar este servicio, definimos también un botón en la interfaz que también toma el nombre de la gráfica como el nombre de la trayectoria que se desea recrear y llama el servicio indicándole que recree dicha trayectoria. Luego de esto, el nodo `/turtle_bot_player` comenzará a publicar las velocidades que lea del archivo del movimiento y esto terminará en el movimiento exitoso del robot en Coppelia.