



**Tecnológico  
de Monterrey**

**Instituto Tecnológico y de  
Estudios Superiores de  
Monterrey**

TE3003B.501

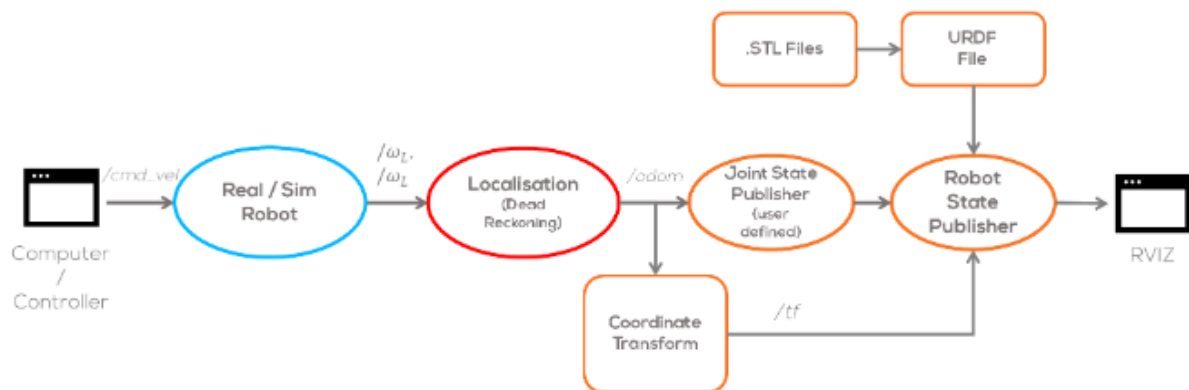
**Integración de robótica y sistemas inteligentes (Gpo 501)**

Semestre: febrero - junio 2024

**Actividad 3 (Cierre Mini challenge 2)**

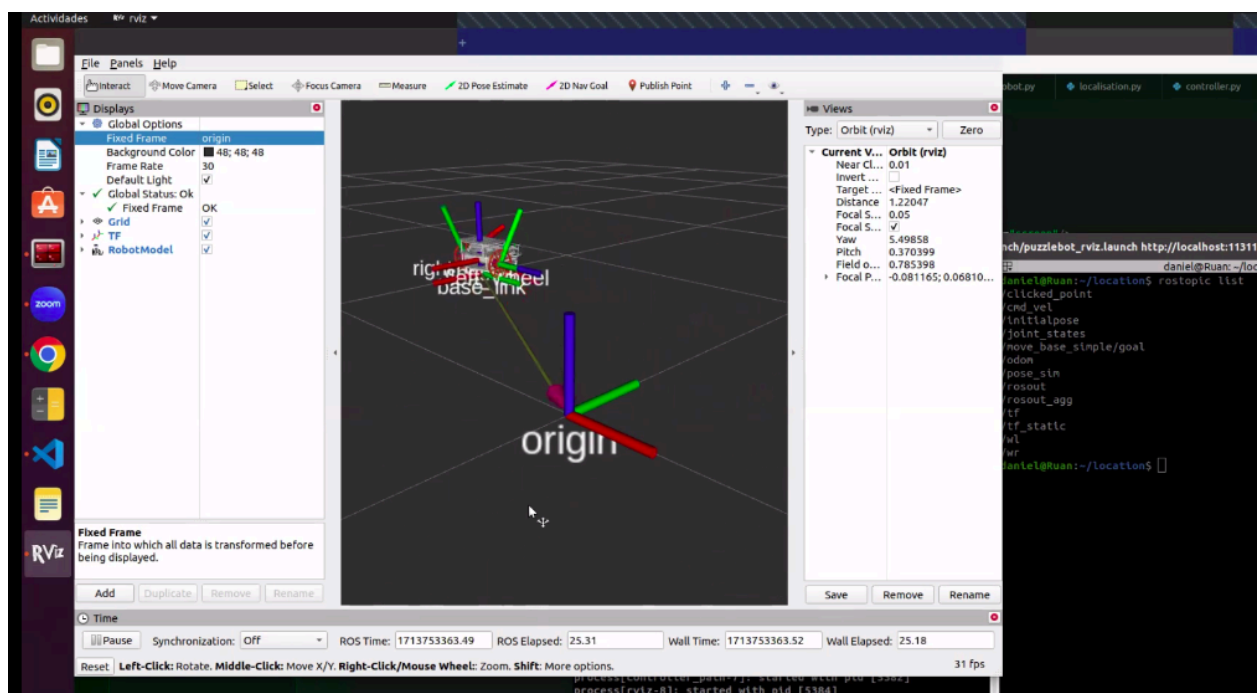
En esta actividad se simula el puzzlebot en rviz con ayuda de modelos.stl y un archivo urdf para ensamblar las partes del robot y poder controlar la posición mientras se traza una trayectoria dada por el usuario.

El objetivo es modelar el robot puzzlebot en ROS. Esto permitirá que, al proporcionarle una trayectoria específica, el robot pueda seguirla de manera precisa. Además, se pretende que esta trayectoria sea visualizable a través de RViz.



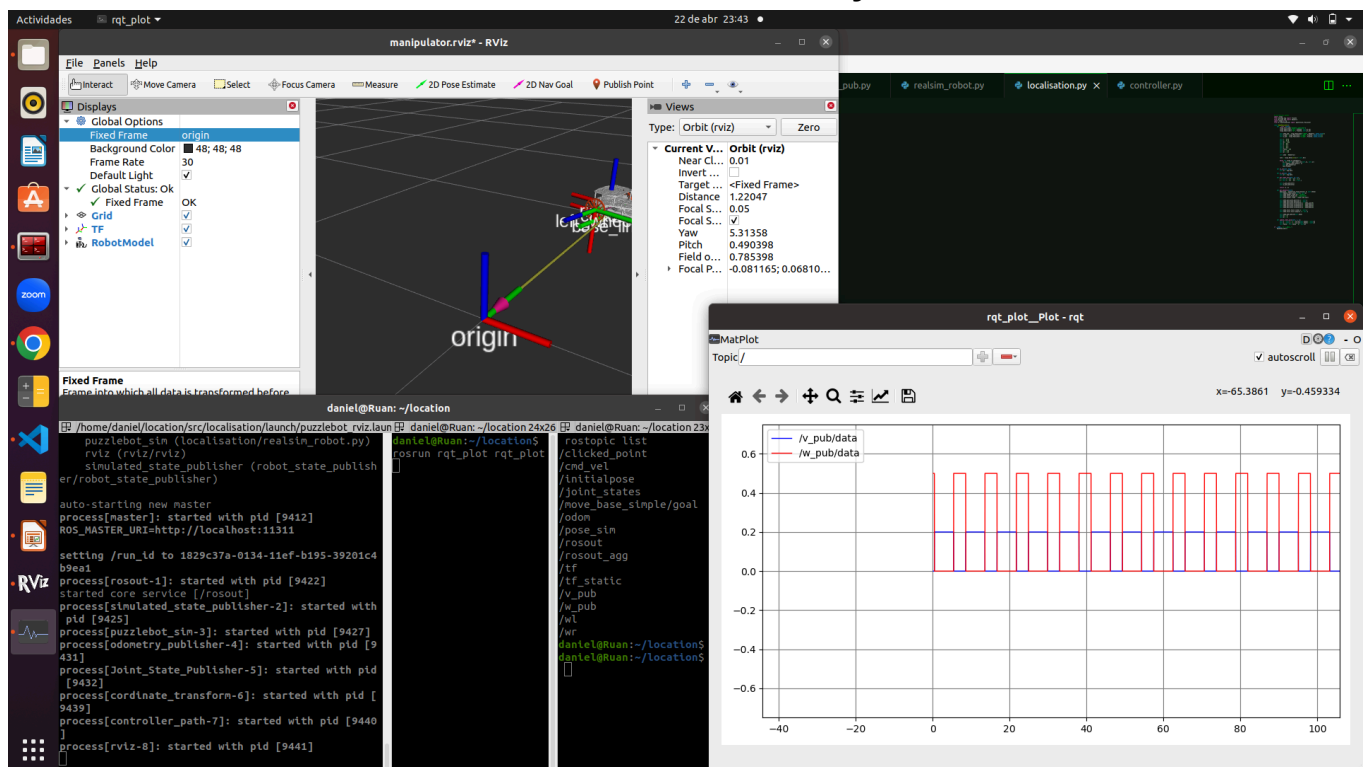
## Solución y análisis

Para la solución de este reto se tuvieron que crear seis nodos para el movimiento del robot, cinco programados en python (joint\_state\_pub.py, localisation.py, controller.py, realsim\_robot.py, transform.py) y uno en urdf para el ensamblaje y visualización del robot.



Como se puede observar en los resultados previos, el ensamblaje del robot se realizó con precisión, permitiendo que las ruedas giren de forma independiente al resto de la estructura con la publicación y subscripción de  $w_l - w_r$  para cada una de las ruedas, además de realizar la transformada para mover el resto del robot. Este hecho indica un manejo efectivo de los archivos URDF y STL. Además, el robot se desplaza adecuadamente dentro de la simulación, lo que sugiere una interpretación acertada del modelo dinámico del Puzzlebot. La trayectoria generada se envía correctamente y el controlador cumple su función de manera efectiva, lo que contribuye al éxito del proyecto en general.

## Publicación de las velocidades V y W



### Velocidad Lineal ( $v$ ):

La velocidad lineal se mantiene constante cuando el vehículo se desplaza en línea recta a una velocidad **constante**.

Durante las aceleraciones o desaceleraciones, la velocidad lineal aumentará o disminuirá, respectivamente.

Si hay una interrupción en el movimiento del vehículo, como detenerse y luego arrancar de nuevo, la velocidad lineal disminuirá a cero y luego aumentará a medida que el vehículo acelera, como se muestra en la gráfica.

### Velocidad Angular ( $w$ ):

La velocidad angular representa la velocidad de rotación del vehículo alrededor de su eje vertical.

Cuando el vehículo se desplaza en línea recta, la velocidad angular es cero, indicando que el vehículo no está girando.

Durante giros, la velocidad angular aumenta para indicar la tasa de giro del vehículo.

Si el vehículo está girando en su lugar (sin desplazamiento lineal), la velocidad angular será **constante** y diferente de cero.

Cabe mencionar que al comparar las velocidades ( $v$ ,  $w$ ) obtenidas a la salida de los nodos "RealSim\_Robot" y "Localisation", los resultados fueron los mismos, ya que las condiciones en una simulación son ideales y la respuesta de la simulación del robot con la localización son iguales, lo cual se estima que cambie al empezar a trabajar en el futuro con el robot real en físico, debido a que las condiciones de derrapamiento, ruido, deslizamiento, etc. alterarán el comportamiento del robot, por lo que será importante tomar en cuenta estas condiciones y usar un filtro para la localización.

## Video demostrativo

A continuación se muestra el video de demostración en ROS de la trayectoria deseada:

[https://drive.google.com/file/d/1My3uzNnN\\_-2\\_mpgzED4Zl1fJyPwMF16o/view?usp=sharing](https://drive.google.com/file/d/1My3uzNnN_-2_mpgzED4Zl1fJyPwMF16o/view?usp=sharing)

## Conclusiones

Después de completar la implementación de los nodos y realizar las pruebas correspondientes utilizando un enfoque de prueba y error, se lograron alcanzar los objetivos establecidos. Se definió una trayectoria específica, en este caso, un recorrido en forma de cuadrado, el cual el robot siguió con precisión. Además, esta trayectoria se visualizó correctamente en RViz, gracias a la adecuada preparación del archivo URDF y los archivos STL del robot.