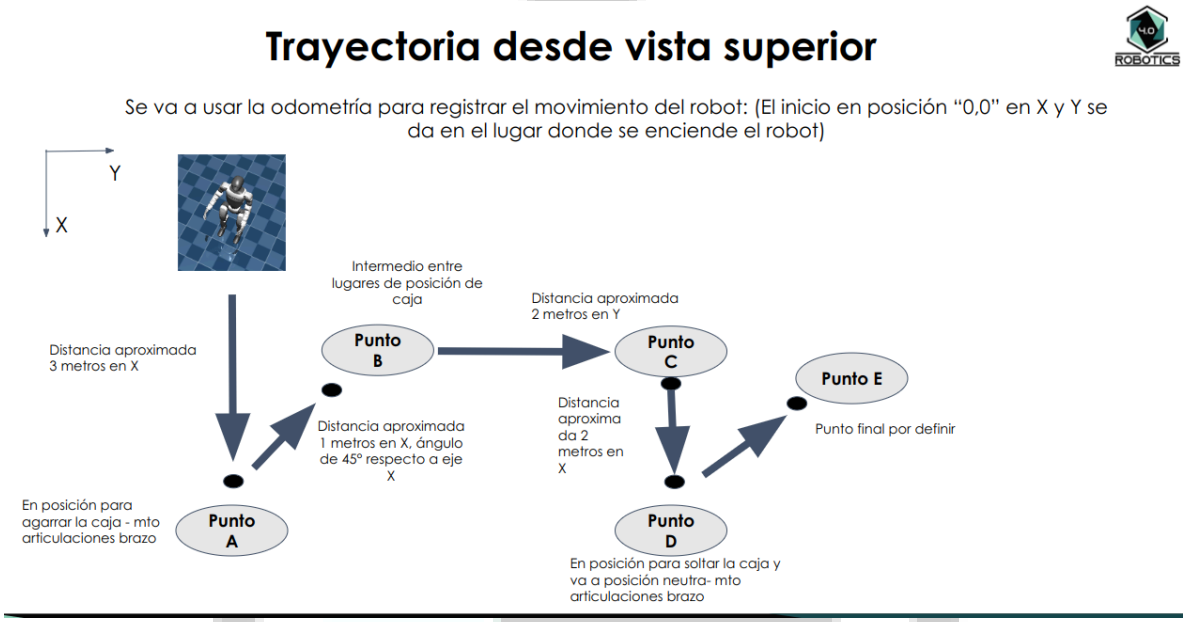
1. Pruebas de abril 24



En primer lugar usando los códigos anteriores se recomienda tener las posiciones de estos puntos definidos o similares. Así como las posiciones de agarre deseadas.

Para esta primer versión los valores de las articulaciones deben ponerse en el código a partir de la línea 190 en los pasos

Prueba 1: Navegación Autónoma por Odometría con Múltiples Objetivos y posibilidad de posiciones articulaciones superiores

Objetivo:
Verificar que el robot G1 de Unitree pueda desplazarse de manera autónoma hacia una secuencia de posiciones predeterminadas utilizando exclusivamente su odometría interna tambien con la posibilidad de coordinar movimientos de los brazos preconfigurados. Se usa el script `g1_autonomousV1.py`.

Procedimiento:

Conexión y Verificación Inicial:

- Asegúrate de tener conexión Ethernet estable entre el PC y el robot G1.
- Enciende el robot en modo normal (R1 + X).
- Verifica que el entorno Python esté correctamente configurado con la SDK2.

1. Ejecución del Script

1. Abre una terminal en el entorno previamente configurado (con ROS 2 y SDK Unitree disponibles).
2. Navega hasta el directorio donde se encuentra el archivo `g1_autonomousV1.py`.

Ejecuta el script:

```
python3 g1_autonomousV1.py
```

3. Ingresa la **interfaz de red** correspondiente (ejemplo: `eth0`).

2. Carga de Objetivos

Cuando se pregunte:

```
¿Cargar puntos desde archivo? (s/n):
```

5. Responde con `n` para ingresar manualmente los objetivos.

Ingresa los objetivos de la **Prueba 2** cuando se solicite:

```
0.5 0.0 0.0
1.0 0.5 1.57
1.0 1.0 3.14
fin
```

3. Ejecución Interactiva del Movimiento de Brazos

En cada objetivo alcanzado, el script preguntará:

```
¿Deseas ejecutar un movimiento de brazos en este objetivo?
(s/n):
```

7. Si deseas probar los movimientos de brazo, responde con `s`.

Luego, se solicitará el número del paso a ejecutar:

```
¿Qué paso deseas ejecutar? (1-6):
```

8. Ingresa un número entre 1 y 6, correspondiente a la configuración predefinida en `get_user_joint_positions(paso)`.

Tras ejecutar el primer paso de brazo, el sistema preguntará si deseas ejecutar otro movimiento:

```
¿Deseas ejecutar otro movimiento de brazos en este objetivo? (s/n):
```

9. Si se responde con `s`, se solicitará otro paso del 1 al 6.

10. En cualquier momento, si se responde con `n`, el robot continuará automáticamente hacia el siguiente punto de navegación.

4. Observaciones a Registrar Durante la Prueba

- Precisión de la orientación hacia el objetivo (¿se imprimió "orientación hacia objetivo lograda"?).
- Precisión en la llegada (¿se imprimió "alta precisión" o "suficientemente cercana"?).
- Tiempo de llegada (¿alcanzó el objetivo antes del `timeout` de 30 segundos?).
- Fluidez de movimiento del brazo (¿se alcanzó correctamente la configuración deseada?).
- Posible necesidad de realineamiento manual o fallos.

5. Finalización

Cuando todos los objetivos sean alcanzados, el sistema imprimirá:

nginx
CopiarEditar
`Todos los objetivos alcanzados y orientados correctamente.`

- Para detener el sistema en cualquier momento presiona `Ctrl+C`.

Criterios de Éxito:

Criterio	Umbral
Tolerancia de posición final	≤ 10 cm
Tolerancia de orientación final	≤ 0.3 rad
Tiempo máximo por objetivo	30 segundos
Respuesta ante interrupción	Robot se detiene de forma segura
Estabilidad del movimiento	No se presentan oscilaciones o caídas
Retroceso preventivo	Ejecutado si se detecta orientación opuesta

Notas Técnicas:

- Se utiliza control proporcional adaptativo con reducción de ganancia en errores pequeños (tipo "P con freno").

- El movimiento lateral es limitado (máximo ±0.2 m/s) y el retroceso permitido es leve (-0.15 m/s máx).

- El sistema normaliza automáticamente los valores de yaw para evitar discontinuidades.

- El script permite cargar objetivos por archivo `.txt` en pruebas futuras.