

1. Pruebas del 3 de abril

Prueba movimientos articulados superiores con ángulos más complejos

- Para este ejemplo se debe de tener el robot colgado
- La interfaz DDS la proporciona el servicio de movimiento incorporado. Al realizar pruebas, se recomienda suspender el robot y entrar en el modo de parada bloqueada (L1 + UP)
- Luego se debe contar con el archivo `g1_arm_sdk_moveV2.py` (`g1_arm_sdk_moveV3.py` para enviar ángulos secuenciadamente), se recomienda guardarlo en `~/unitree_sdk2_python/example/g1/high_level` y desde esa misma ubicación se ejecuta:

`python3 g1_arm_sdk_moveV2.py nombreInterfaz`

- Estos códigos controlan los movimientos del robot G1 de Unitree a nivel de articulaciones, utilizando la API `unitree_sdk2py`. Permite definir posiciones objetivo para las articulaciones del brazo y la cintura, interpolarlas suavemente mediante interpolación sinusoidal y enviarlas al robot mediante comandos de bajo nivel.

Este código implementa el control de articulaciones del robot cuadrúpedo G1 de Unitree, enviando comandos de posición mediante la SDK de Unitree y utilizando una estrategia de interpolación para movimientos suaves.

- ♦ Interpolación sinusoidal

Para evitar movimientos bruscos en las articulaciones, el código emplea interpolación basada en una función coseno:

$$q_{\text{interpolado}} = q_{\text{inicial}} + (q_{\text{objetivo}} - q_{\text{inicial}}) \cdot \frac{1 - \cos(\pi \cdot t/T)}{2}$$

Este método asegura transiciones suaves con aceleraciones continuas, evitando cambios abruptos en la velocidad.

- ♦ Uso de hilos (threads)

Para mantener el control en tiempo real sin bloquear la ejecución principal, se usa ejecución concurrente con hilos (threading):

Se crea un hilo recurrente (RecurrentThread) que ejecuta LowCmdWrite() cada 20 ms, enviando los comandos interpolados a las articulaciones.

Se usa un bloqueo (threading.Lock) para proteger el acceso concurrente a self.low_state, evitando condiciones de carrera.

Se emplea eventos (threading.Event()) para detener el movimiento cuando sea necesario, asegurando una finalización segura.

- ♦ Objetivo principal del código

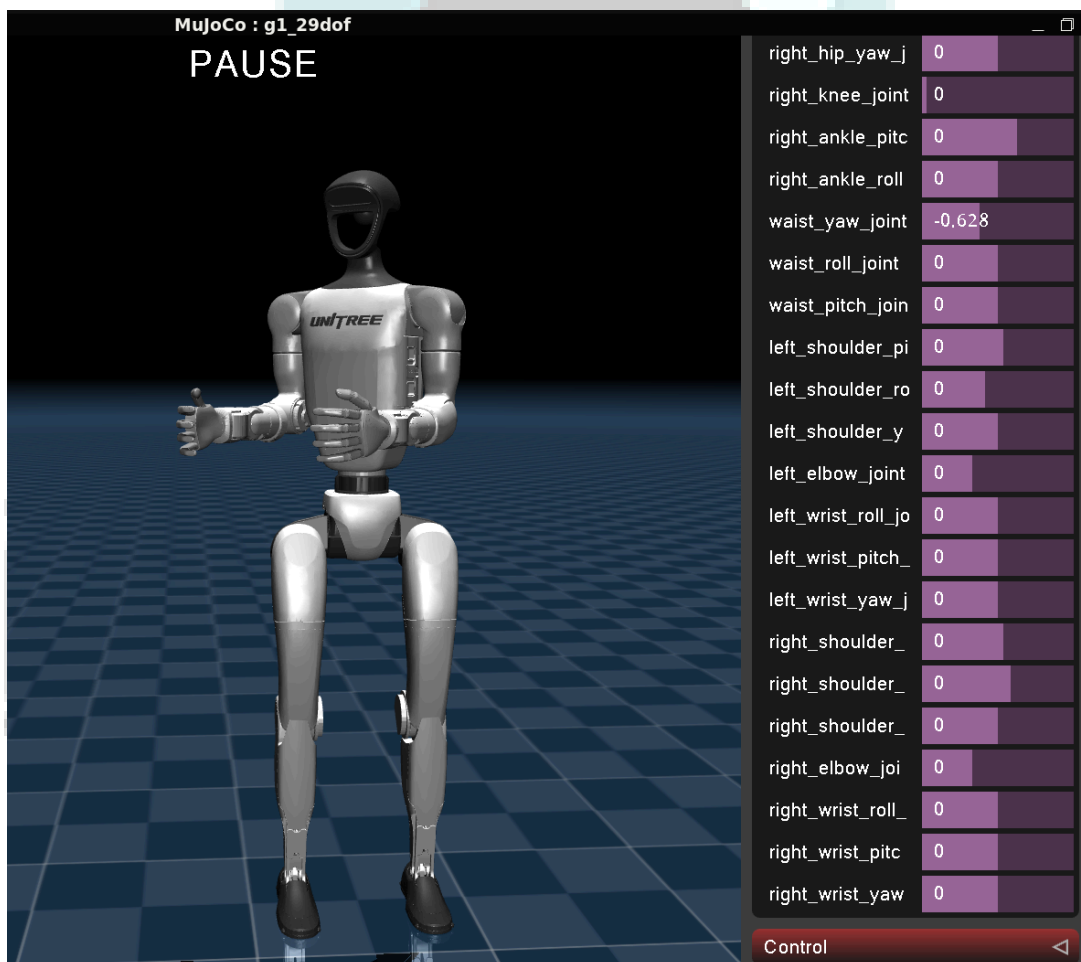
El código permite controlar las articulaciones del G1 de manera segura y precisa, utilizando una estrategia de interpolación sinusoidal para el movimiento y threads para garantizar una ejecución continua en tiempo real sin interrupciones.

- Seguidamente se realizará la ejecución de 10 posiciones personalizadas: considerando los límites articulares:

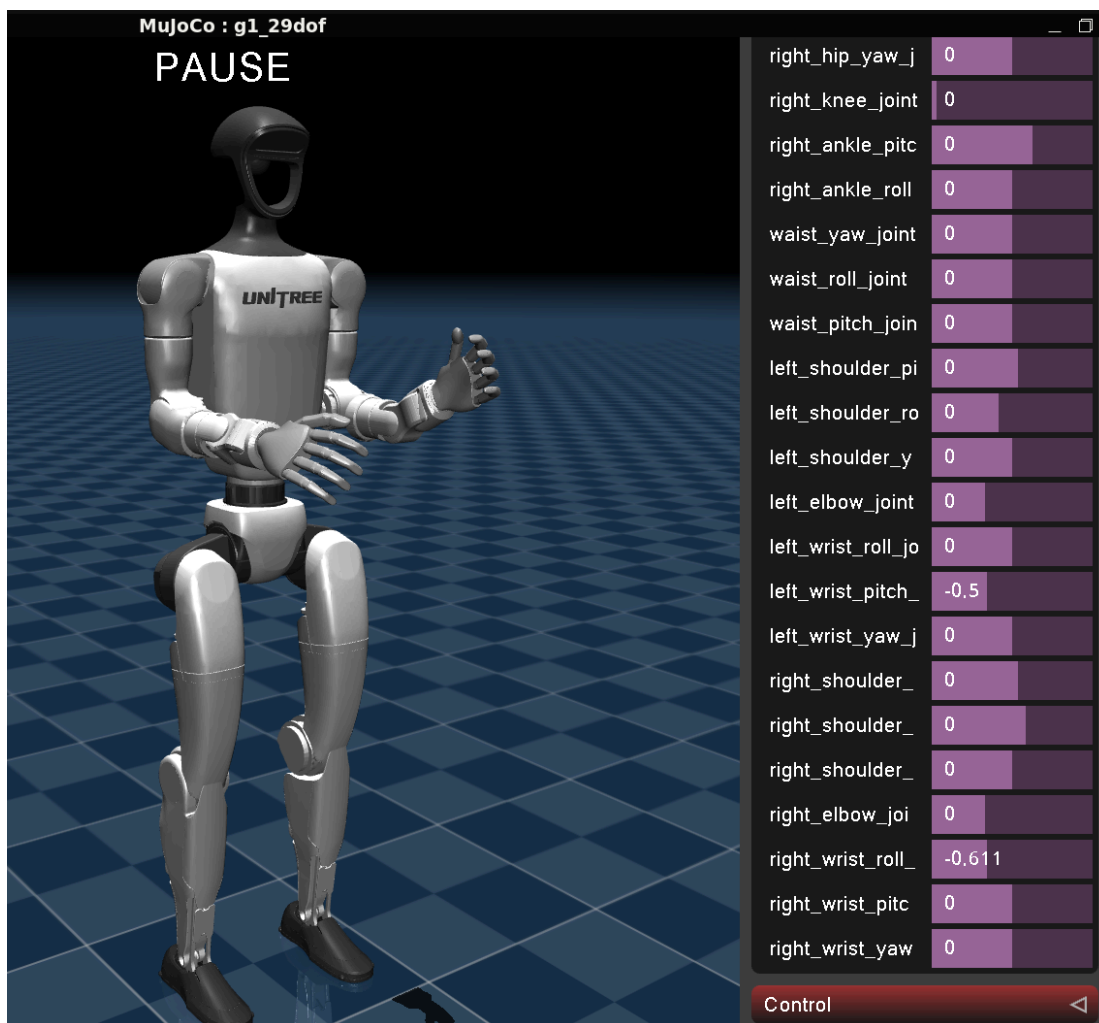
No.	Joint Name	Min (°)	Max (°)
16	L_SHOULDER_PITCH	-176.99°	153.00°
17	L_SHOULDER_ROLL	-90.99°	129.00°
18	L_SHOULDER_YAW	-150.00°	150.00°
19	L_ELBOU	-60.00°	120.00°
20	L_WRIST_ROLL	-113.00°	113.00°
21	L_WRIST_PITCH	-92.50°	92.50°
22	L_WRIST_YAW	-92.50°	92.50°
23	R_SHOULDER_PITCH	-176.99°	153.00°
24	R_SHOULDER_ROLL	-129.00°	90.99°
25	R_SHOULDER_YAW	-150.00°	150.00°
26	R_ELBOU	-60.00°	120.00°
27	R_WRIST_ROLL	-113.00°	113.00°
28	R_WRIST_PITCH	-92.50°	92.50°
29	R_WRIST_YAW	-92.50°	92.50°

13	WAIST_YAW	-150.00°	150.00°
14	WAIST_ROLL	-29.79°	29.79°
15	WAIST_PITCH	-29.79°	29.79°

1. Movimiento cintura

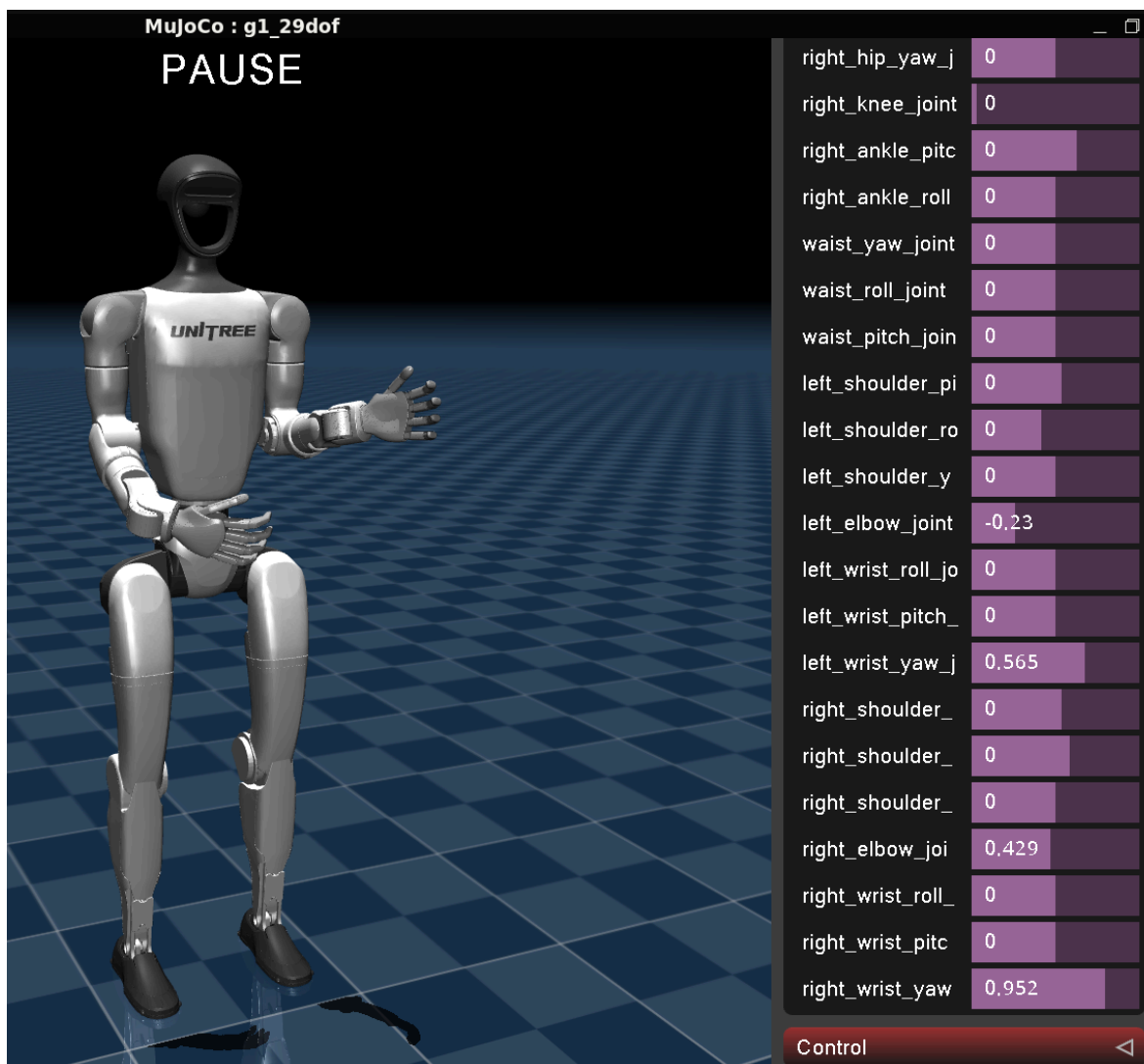


2. Movimiento muñecas

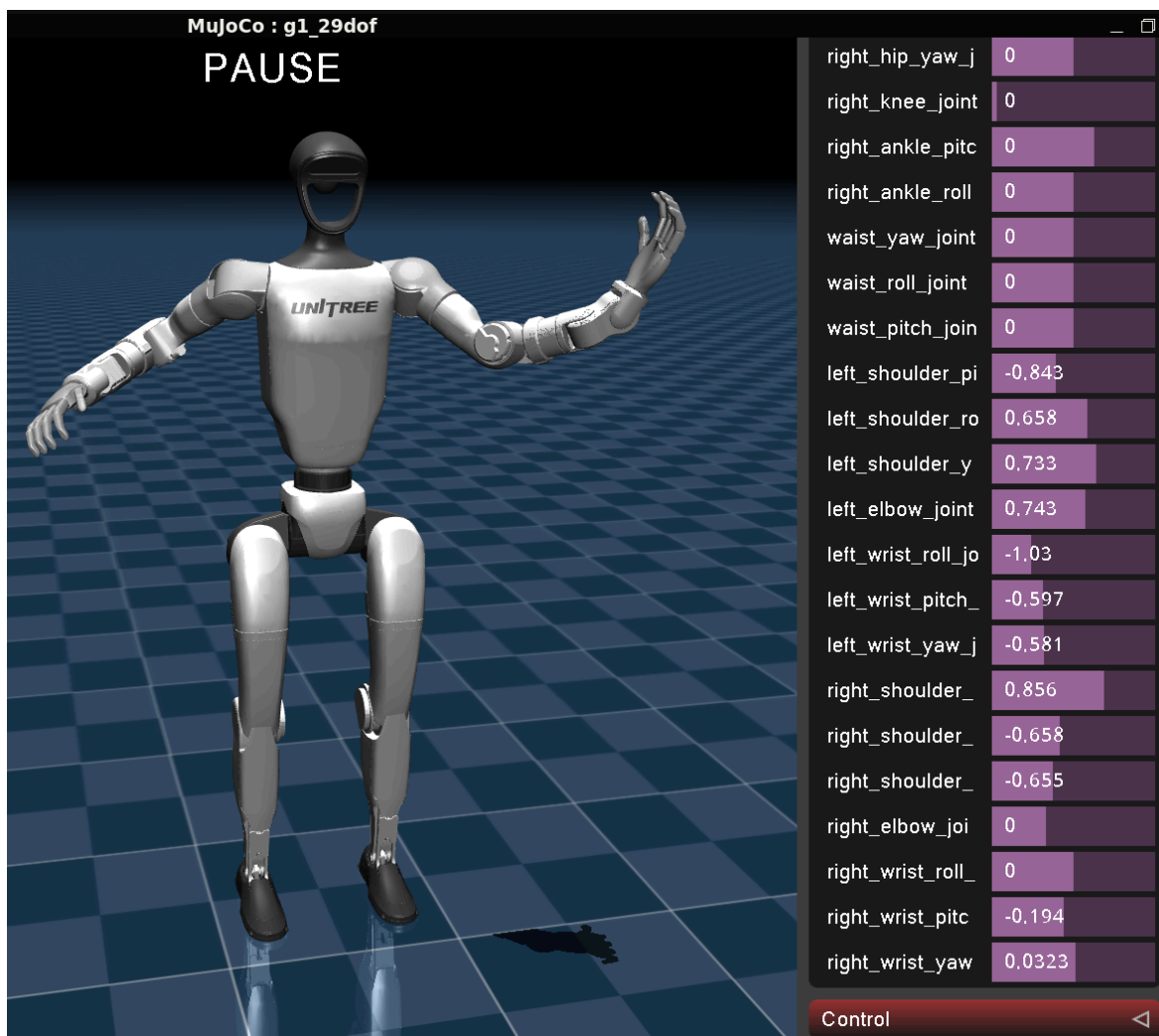


3. Movimiento muñecas y codos

ROBOTICS

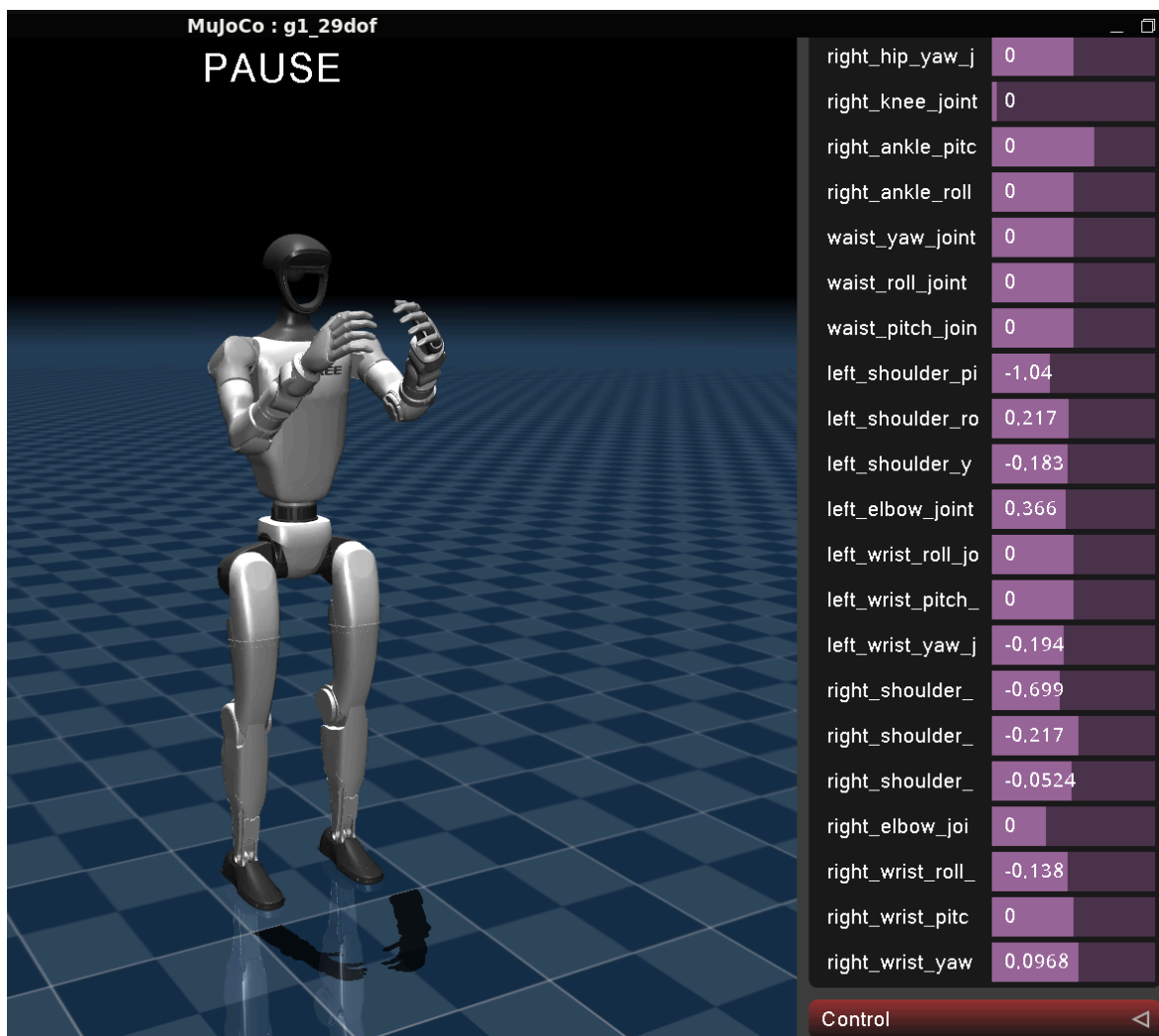


4. Movimiento varias articulaciones "sosteniendo palo imaginario"



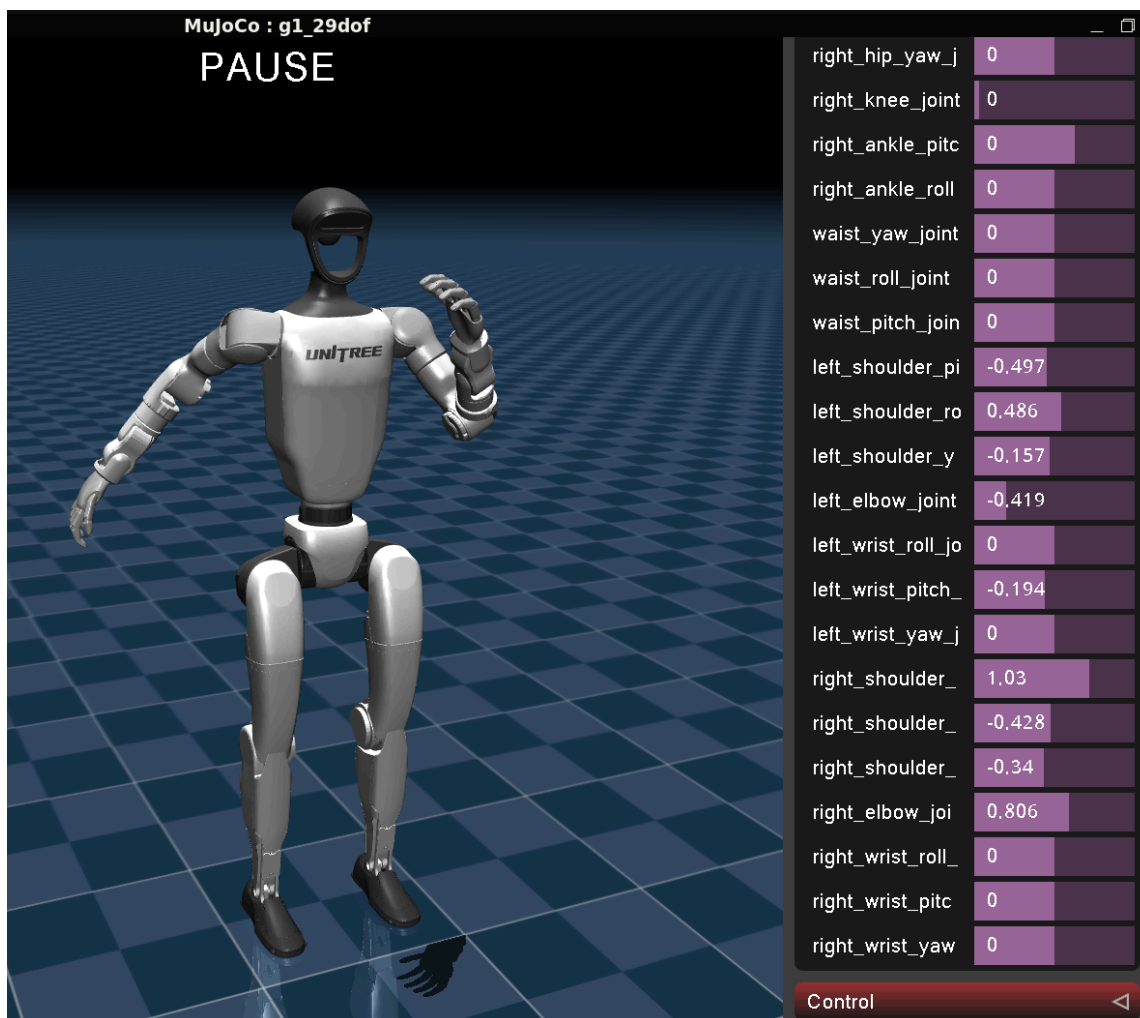
5. Movimiento agarrando una pelota

ROBOTICS



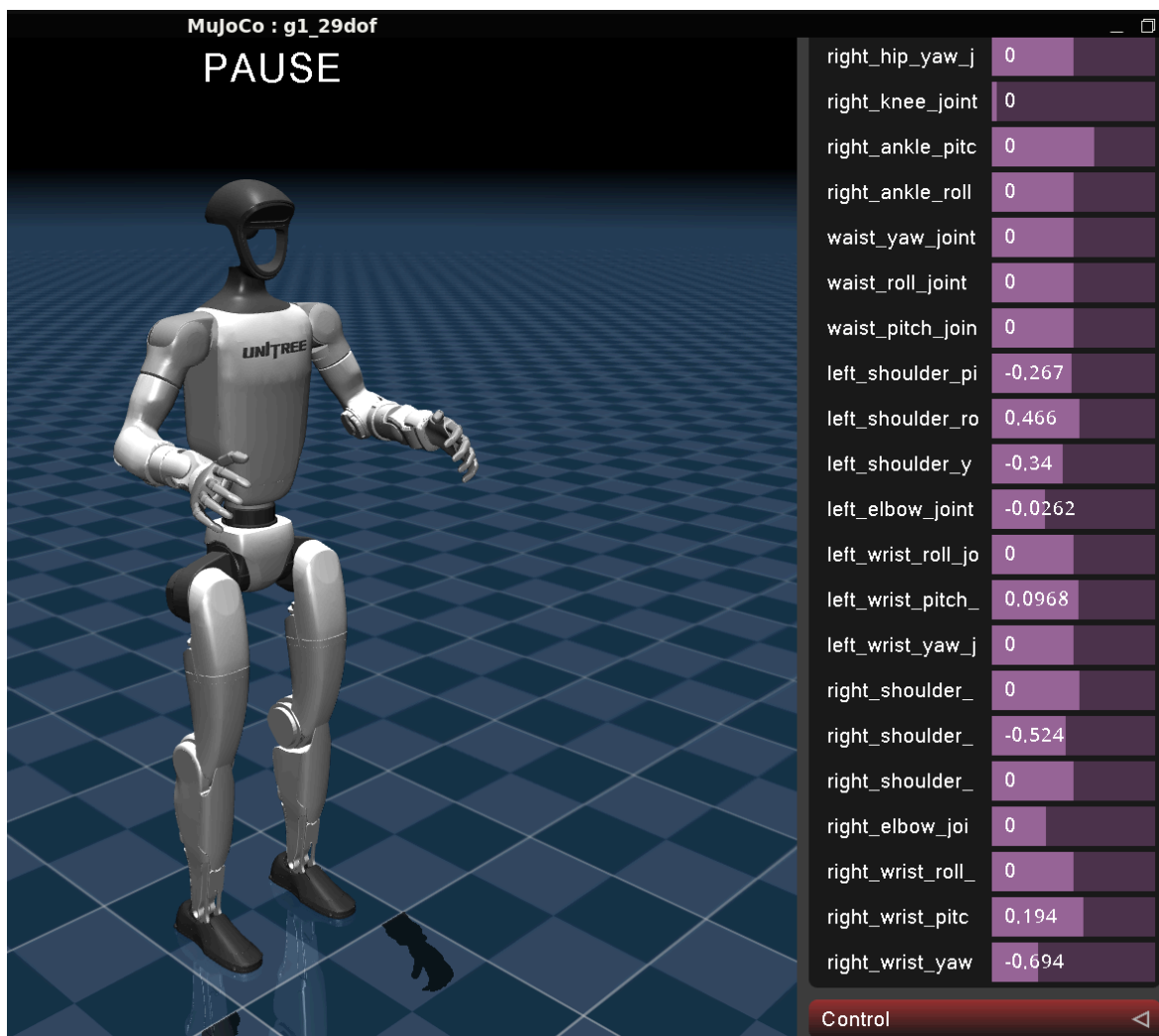
6. Movimiento preparado para correr

ROBOTICS



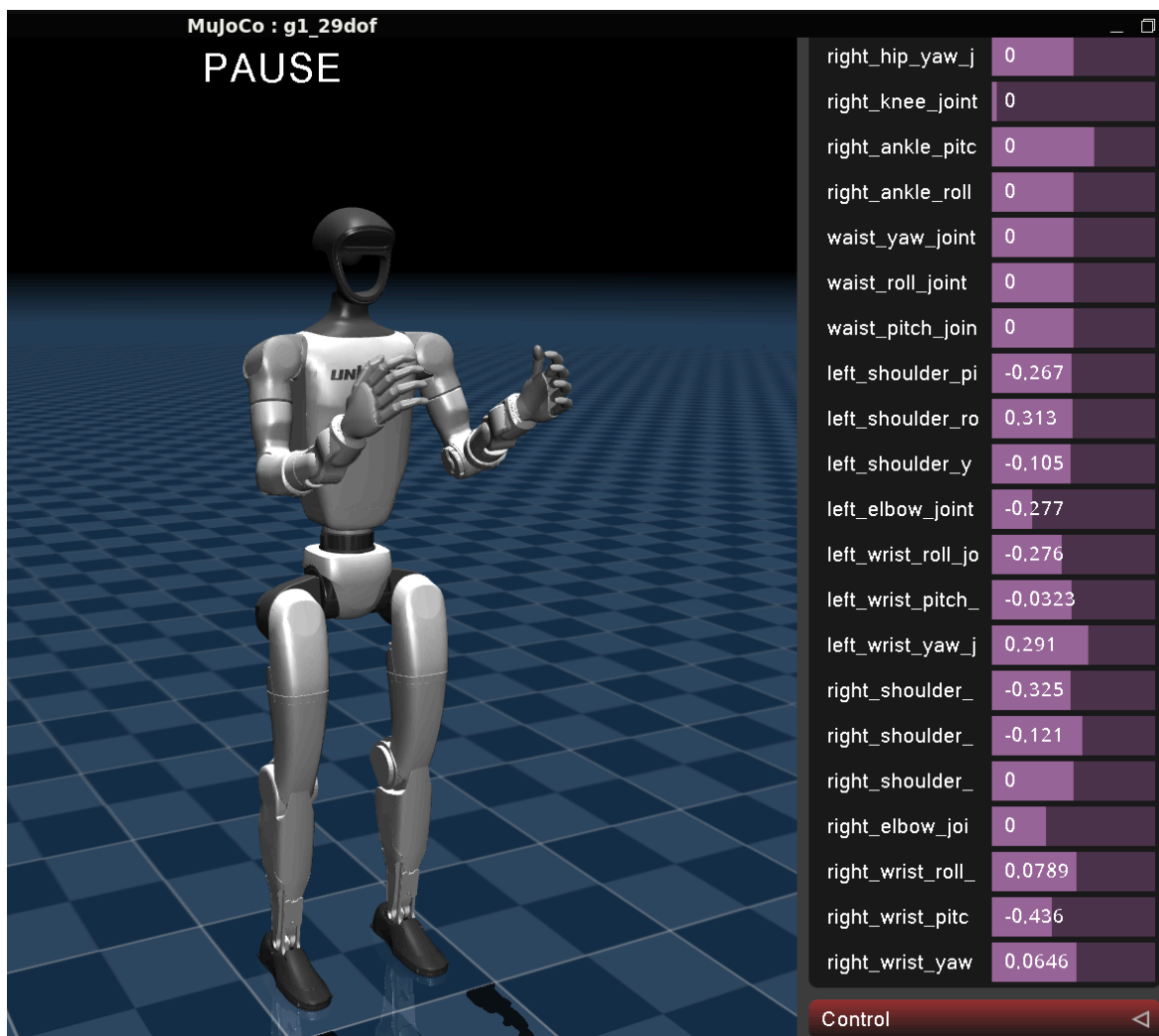
7. Inicio secuencia de movimiento caja

ROBOTICS



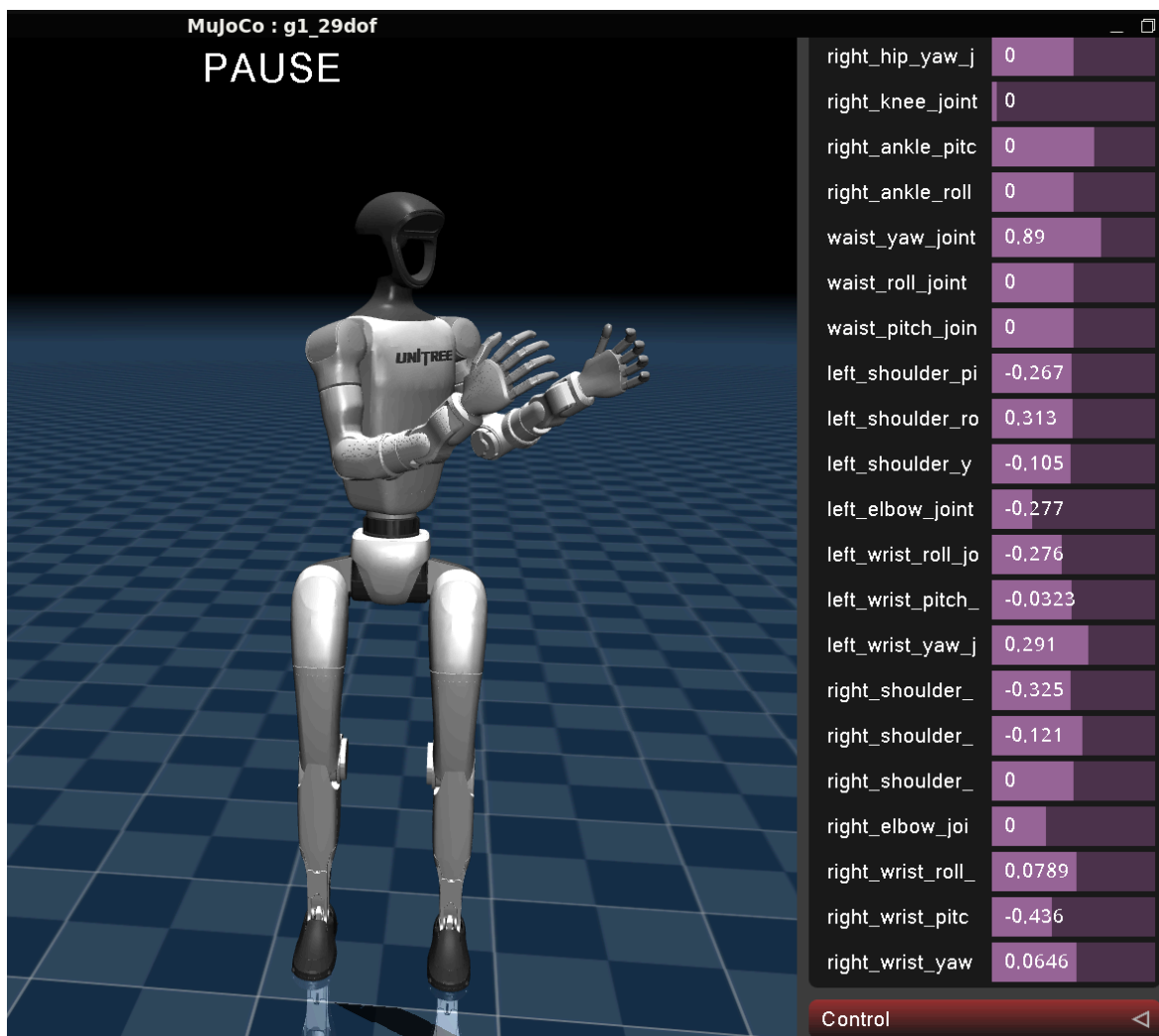
8. Segundo movimiento secuencia caja

ROBOTICS



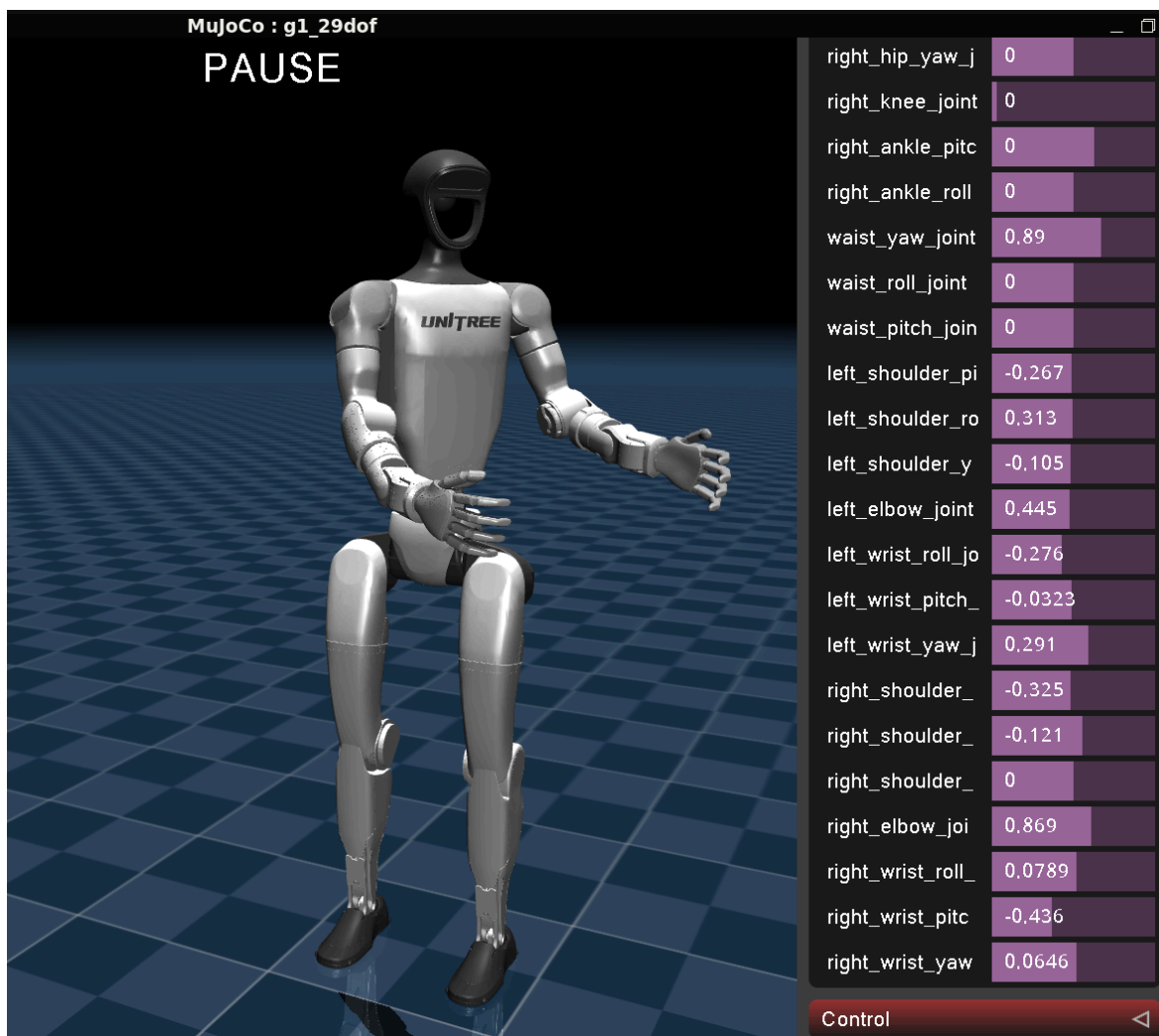
9. Tercer movimiento secuencia caja

ROBOTICS



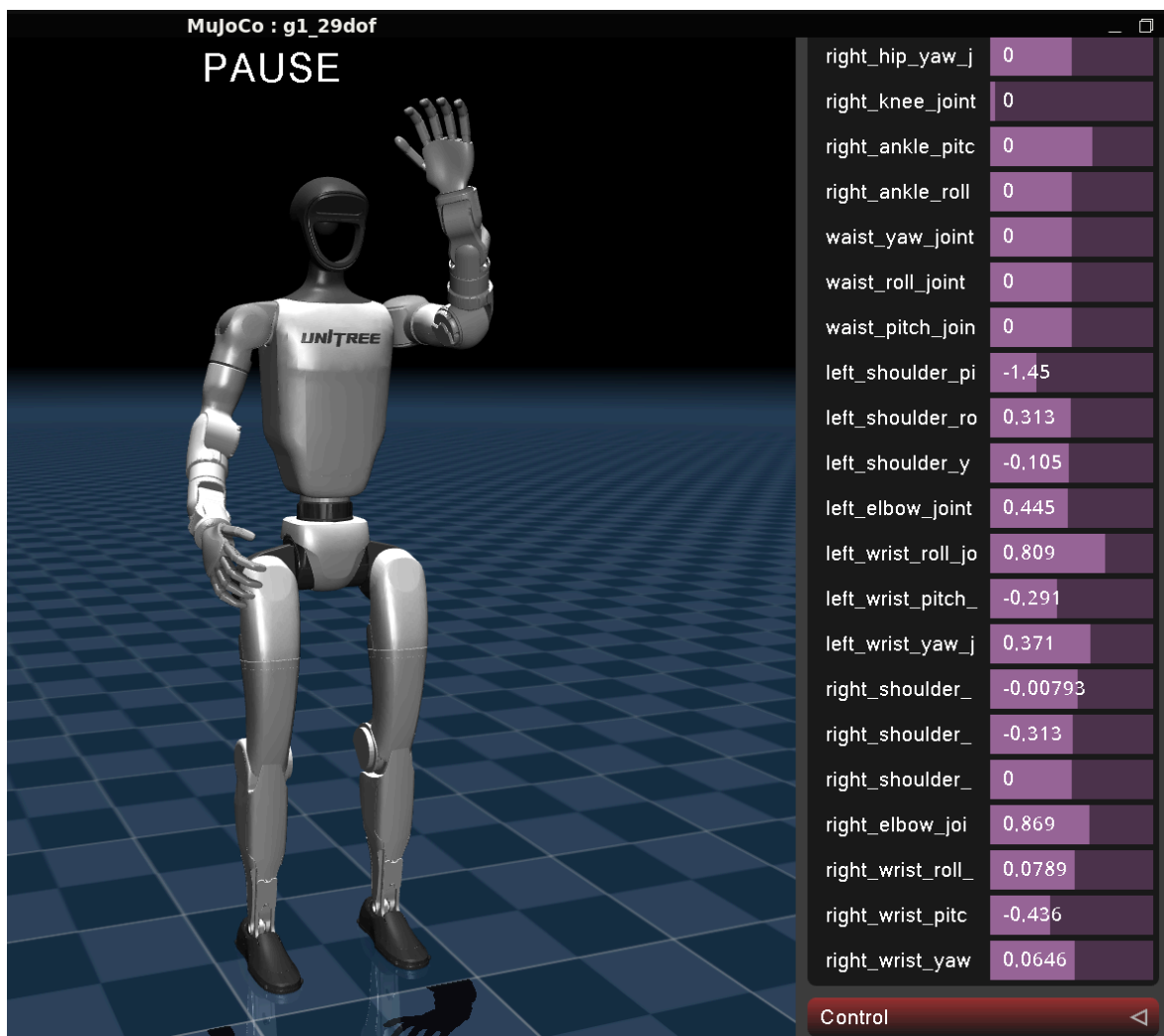
10. Cuarto movimiento secuencia caja

ROBOTICS



11. Movimiento final secuencia caja

ROBOTICS



ROBOTICS