Manual de Uso — Control del Kinova Gen3 en Docker (Teleoperación + Control Cartesiano)

---

1. Ejecutar el contenedor Docker

Para instalar las librerías necesarias (PyKinova + pygame) es importante **NO usar la bandera**

`--rm -it`, porque eliminaría el contenedor después de cerrarlo.

Por ello, el script ./docker_run.sh ya está configurado para ejecutar el contenedor **sin auto-borrarlo**.

Ejecuta en un terminal:

```
./docker_run.sh
```

Este script (`docker_run.sh`) inicia un contenedor Docker preparado para trabajar con la imagen kinova-phantom, configurando automáticamente el acceso a la interfaz X11, GPUs NVIDIA, dispositivos físicos como el puerto `/dev/ttyACM0` y la red interna con `--net=host`, además de montar la carpeta de trabajo del host (carpeta actual) dentro del contenedor (/catkin_ws/shared_folder).

Con esto logramos que, si el contenedor ya existe, simplemente lo inicia. Si no existe, lo crea con todos los permisos, volúmenes y configuraciones necesarios sin borrarse al cerrarlo.

2. Instalar librerías dentro del contenedor

Dentro del contenedor, ir al directorio del paquete:

```
cd /catkin_ws/shared_folder/pyKinovaGen3
```

Instalar dependencias necesarias:

```
./install.sh
```

Este script instala:

- **pykinova** → librería oficial para controlar el Kinova Gen3
- **pygame** → para la interfaz visual del teleop con teclado
- otros paquetes auxiliares para control cartesiano

---

3. Terminales necesarios

Para realizar las pruebas son necesarios **al menos 4 terminales**, todos dentro del contenedor.

Terminal 1 — Lanzar descripción del robot

```
cd /catkin_ws/kinova/
source install/setup.bash
ros2 launch kortex_description view_robot.launch.py \
   robot_type:=gen3 dof:=7 gripper:=robotiq_2f_140 \
   use_sim_time:=true launch_rviz:=true
```

> IMPORTANTE:
>
> En RViz aparecerá una ventana con *sliders* de control manual.
>
> **Ciérrala**, ya que controlaremos el robot mediante *topics*.

Terminal 2 — Abrir nueva terminal dentro del contenedor

Desde tu host, abre un nuevo shell dentro del contenedor:

```
sudo docker exec -it kinova-phanthom_container /bin/bash
```

Luego, ejecutar el controlador que espera la inicialización de los joints:

```
cd /catkin_ws/shared_folder/pyKinovaGen3/tests/
python3 ee_cartesian_velocity_controller.py
```

Este nodo:

- Espera a que el brazo esté en **HOME**
- Luego empieza a aceptar velocidades cartesianas (`/tool_twist_vel`)

Terminal 3 — Enviar el brazo a HOME

Abrir otra terminal del contenedor:

```
sudo docker exec -it kinova-phanthom_container /bin/bash
```

Ejecutar:

```
cd /catkin_ws/shared_folder/pyKinovaGen3/tests/
python3 home_kinova.py
```

Cuando termine:

- El brazo estará en **posición home**
- El controlador del Terminal 2 comenzará a aceptar comandos de velocidad

---

Terminal 4 — Teleoperación con teclado (pygame)

Abrir otro terminal dentro del contenedor:

```
sudo docker exec -it kinova-phanthom_container /bin/bash
```

Ejecutar el teleop:

```
cd /catkin_ws/shared_folder/pyKinovaGen3/tests/
python3 cartesian_key_teleop.py
```

Esto abrirá una ventana con **interfaz pygame**:

- Muestra las velocidades lineales y angulares
- Permite controlar el brazo con teclado
- Permite abrir/cerrar la pinza
- Permite ajustar la velocidad con (0.5 cm/s por paso)