

Práctica 2: Diseño 3D de un Robot en Blender

Modelado y Simulación de Robots

Los objetivos de esta práctica son:

- Manejo de Blender
- Diseño 3D de un robot
- Diseño de mecanismo de pick and place
- Análisis de fuerzas de las articulaciones.

Fase 1: Modelado del Robot

En esta primera fase se requiere modelar un robot tipo rover en Blender con las siguientes características/restricciones.

- Altura: 1.5 - 3 metros
- Anchura: 1.7 - 2.2 metros
- Longitud: 3-4 metros
- Número de ruedas: 4-8
- Tracción ruedas: a elegir

Define los links, joints y restricciones que consideres oportunas. Es recomendable ir comprobando el funcionamiento del modelo en pybullet. Puedes utilizar el script “base.py” que encontrarás en el repositorio Gitlab de la asignatura, para mostrar en pybullet el modelo URDF generado con Blender.

Fase 2: Extendiendo el modelo

En esta segunda fase vamos a extender el modelo 3D del robot para que soporte una nueva funcionalidad. Esta nueva funcionalidad debe permitir al robot coger cubos y depositarlos en un compartimento del propio robot. Esta nueva funcionalidad se denomina mecanismo de “pick and place” e incluye un manipulador y un gripper.

- Los cubos se encuentran siempre en el suelo ($z=0$) y su tamaño y dimensiones son fijas, largo_arista=0.5 metros y peso=4 kg.
- El compartimento para alojar los cubos (mínimo 3), puede estar en cualquier parte del chasis siempre y cuando permita al robot desplazarse sin perder los cubos durante su recorrido.

- La base del mecanismo “pick and place” para coger los cubos y depositarlos en el compartimento es un manipulador de tipo SCARA, como se visualiza en la Figura 1. Se puede utilizar diseño de brazos mecánicos (conjunción de joints y links), y end-effector que se considere oportuno.

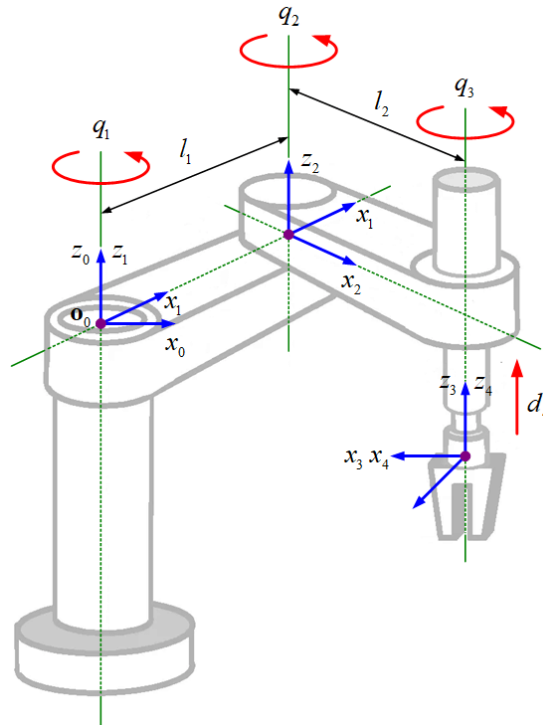


Figura 1. Esquema de manipulador de tipo SCARA. ([Lavín-delgado et al., 2020](#))

- El movimiento del mecanismo “pick and place” debe realizarse utilizando cinemática inversa.
- Las restricciones respecto al diseño del manipulador son las siguientes:
 - Las flechas son los ejes de libertad articular (Rotacional - Rotacional - Rotacional - Traslacional), en algunos libros podría ser visualizado como RRTR.
 - Las distancias L_1 y L_2 deben estar comprendidas entre 1 y 2 metros de longitud cada una.
- Respecto al gripper:
 - Gripper de 2 o más dedos
 - Se deberá localizar de forma fija al sistema coordenado $\{4\}$

Puntos de diseño a tener en cuenta:

- Las restricciones de tamaño definidas en la Fase 1 no están contempladas para el compartimento ni para el mecanismo pick and place. Aun así, utiliza el sentido común para establecer sus tamaños.
- JOINTS y grados de libertad del mecanismo “pick and place” (brazo + end effector)
- Durante el proceso de “pick and place” y hasta que el mecanismo vuelva a su posición original, el robot no puede realizar ningún movimiento.

- Durante el proceso de “pick and place”, el brazo + end effector no puede atravesar ninguna parte del robot. Si es necesario puedes configurar varios checkpoints intermedios para asegurar que realizar una trayectoria segura.
- El tiempo máximo será de 15 segundos para realizar:
 - Movimiento del brazo+end-effector hacia la caja
 - Coger la caja
 - Movimiento del brazo+end-effector hacia el contenedor
 - Soltar la caja
 - Movimiento del brazo+end-effector a su posición original

Fase 3: Análisis del coste del mecanismo “pick and place”

En esta fase integramos el robot diseñado en las fases anteriores en pybullet con el fin de evaluar el coste de nuestro mecanismo para coger cubos. Para ello se generará un escenario sencillo donde el robot estará colocado en el origen de coordenadas (0,0,0), y un cubo estará colocado en (0,4,0).

El robot debe empezar su recorrido hacia (0,4,0), al llegar a este punto debe parar, y recoger el cubo para introducir la pieza en su contenedor. **NO** se teleoperará en ningún momento el robot. Puedes asumir que el robot y el cubo siempre están en las mismas coordenadas. Una vez que el cubo esté en el contenedor y el mecanismo de “pick and place” vuelva a su posición original el programa finalizará.

En esta fase nos interesa estudiar el gasto (G) en términos de potencia que tiene nuestro mecanismo de “pick and place”. Para ello obtendremos las fuerzas aplicadas en cada JOINT del mecanismo “pick and place” (brazo + end effector) utilizando el método `getJointState()`. Se obtendrán medidas desde el momento que el mecanismo inicia el “pick” hasta que vuelve a su posición de origen.

El gasto (G -parcial) en cada momento se define como el sumatorio de las fuerzas aplicadas a cada JOINT involucrado en la cinemática inversa.

$$G = \sum_{i=0}^N |F_i|$$

Siendo:

- N , número de joints del mecanismo “pick and place” (involucrados en cinemática inv.)
- F , fuerzas de par motor aplicadas al joint

El **objetivo** es realizar la operación de pick and place con el menor coste posible, cumpliendo los requerimientos máximos de tiempo (15 segundos).

Para realizar la simulación correctamente, **no** debes utilizar *setRealTimeSimulation* y debes configurar el bucle principal para que cada paso de simulación se realice cada 0.005 segundos.

En esta fase debes obtener:

- A. Cálculo del G-parcial cada 0.01 segundos (muestreo medidas cada 0.01 seg). Generar un CSV con el formato siguiente:

Tiempo, NúmeroJoints, G_parcial

NumeroJoints: Es el número de joints implicados en el cálculo de la cinemática inversa

- B. El valor G-total de tu mecanismo “pick and place” durante el recorrido completo. El valor G-total se calculará como la suma de todos los G-parciales calculados en el apartado anterior.
- C. Generar un plot “tiempo vs G-parcial”, donde se muestran el gasto (G-parcial) de tu mecanismo pick and place. En el título de este plot debe aparecer el valor G-Total calculado en el apartado B) y la desviación estándar de G-parcial.