PONTIFICIA
UNIVERSIDAD
CATÓLICA DE
VALPARAÍSO

pucv.cl

EIE ___-_ Robótica e inteligencia artificial

Módulo 3
Programación y lógica de funcionamiento S16

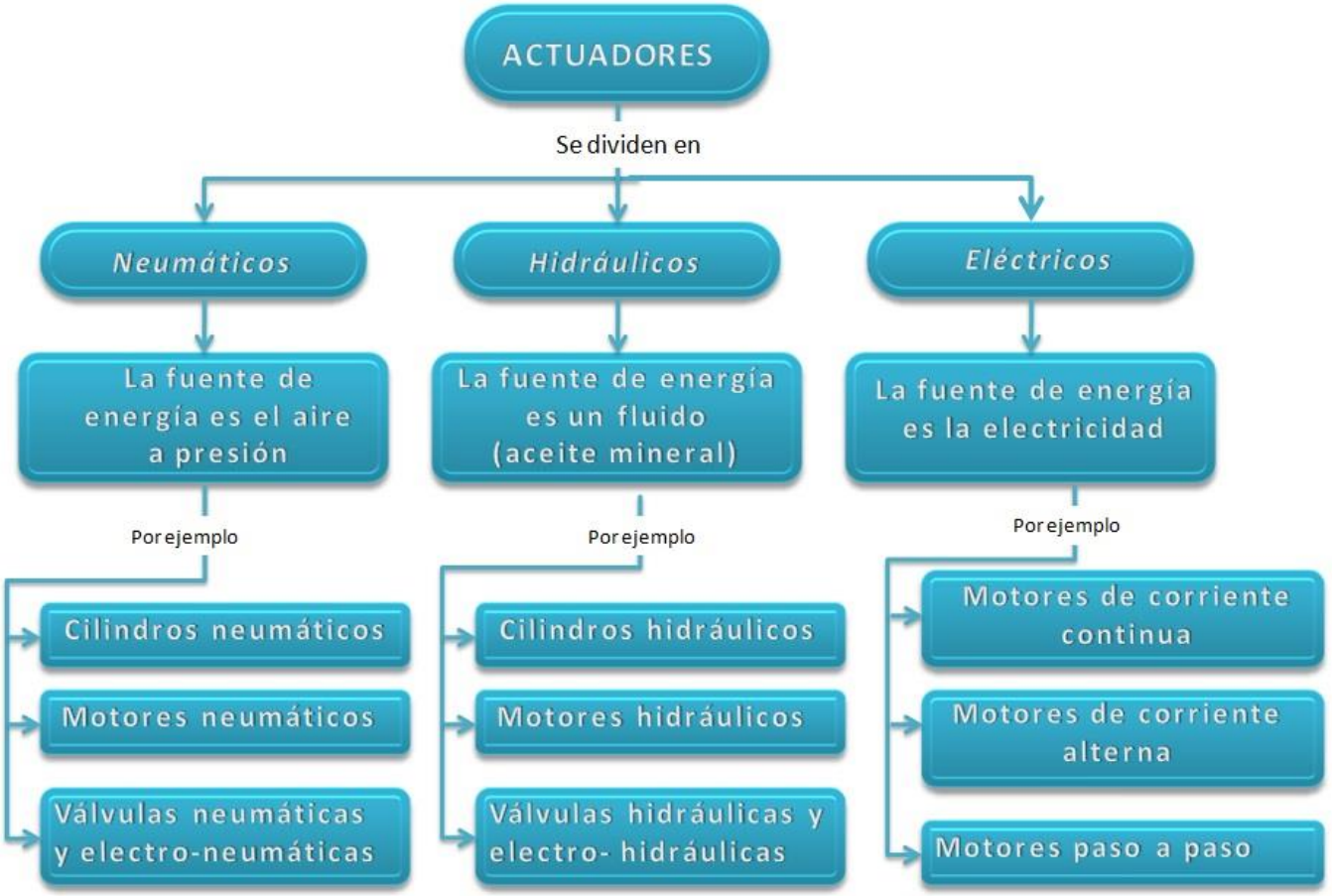
Valparaíso, 2022

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

PROGRAMACIÓN Y LÓGICA DE FUNCIONAMIENTO
SESIÓN 16

Programación y lógica de funcionamiento

Los actuadores componen una parte fundamental de la constitución del robot, estos permiten el desplazamiento o la operación de la máquina robótica, ejecutando las acciones informadas por la CPU.



Programación y lógica de funcionamiento

El tipo de actuadores más utilizado en la robótica de investigación es el electrónico, el cual presenta diversas configuraciones donde se puede obtener torque, precisión y velocidad, además de un control casi directo con la unidad de procesos centrales, obedeciendo señales eléctricas para su operación.

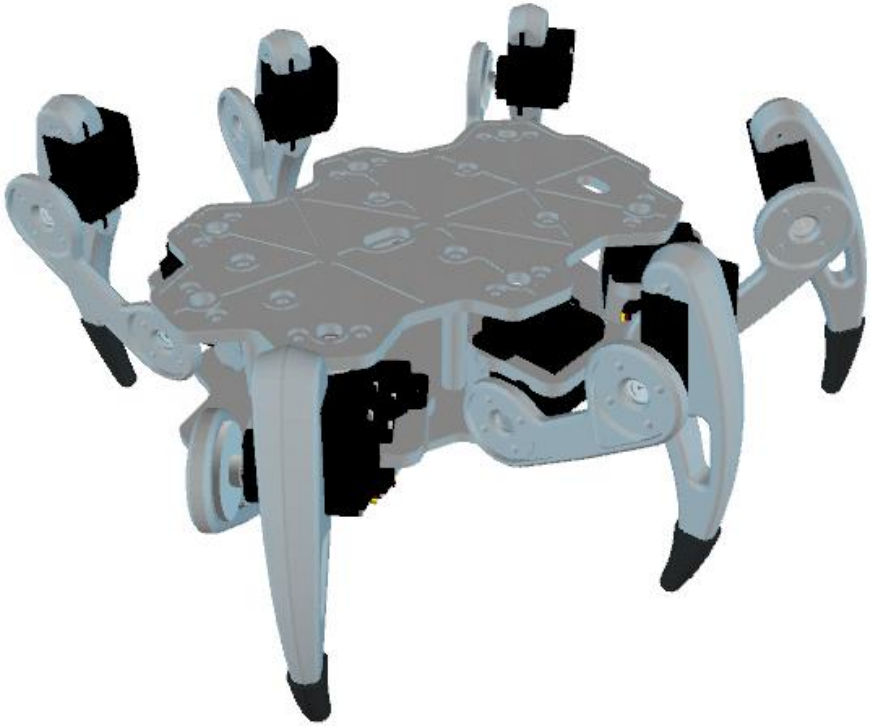
Cabe destacar que el actuador como tal no es programado, sino que este recibe una señal proveniente del control central del robot para posteriormente ejecutar la acción.

El efector final era un robot manipulador, también es considerado un actor. Para separar estos dos elementos en algunos análisis se hace la diferencia.

Programación y lógica de funcionamiento

En muchas ocasiones es posible identificar diagramas de flujos que se refieren simplemente al movimiento o desplazamiento del robot, donde se realiza un análisis más profundo se puede identificar una secuencia de cada actuador necesario para lograr tal movimiento.

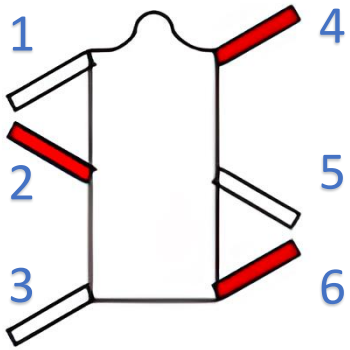
Si se analiza el movimiento de un robot hexápodo al detalle, es posible modificar la gran cantidad de movimientos secuenciales que se debe realizar para lograr el desplazamiento de este.

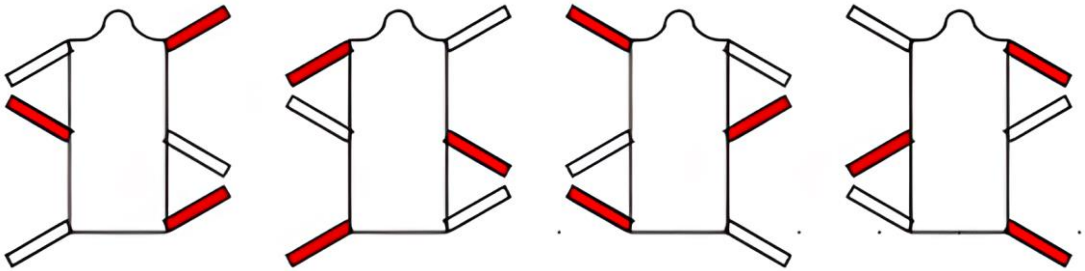


Programación y lógica de funcionamiento

Actuadores (actividad)

Analizar y anotar La secuencia de movimientos del esquema del robot hexápodo mostrado a continuación.

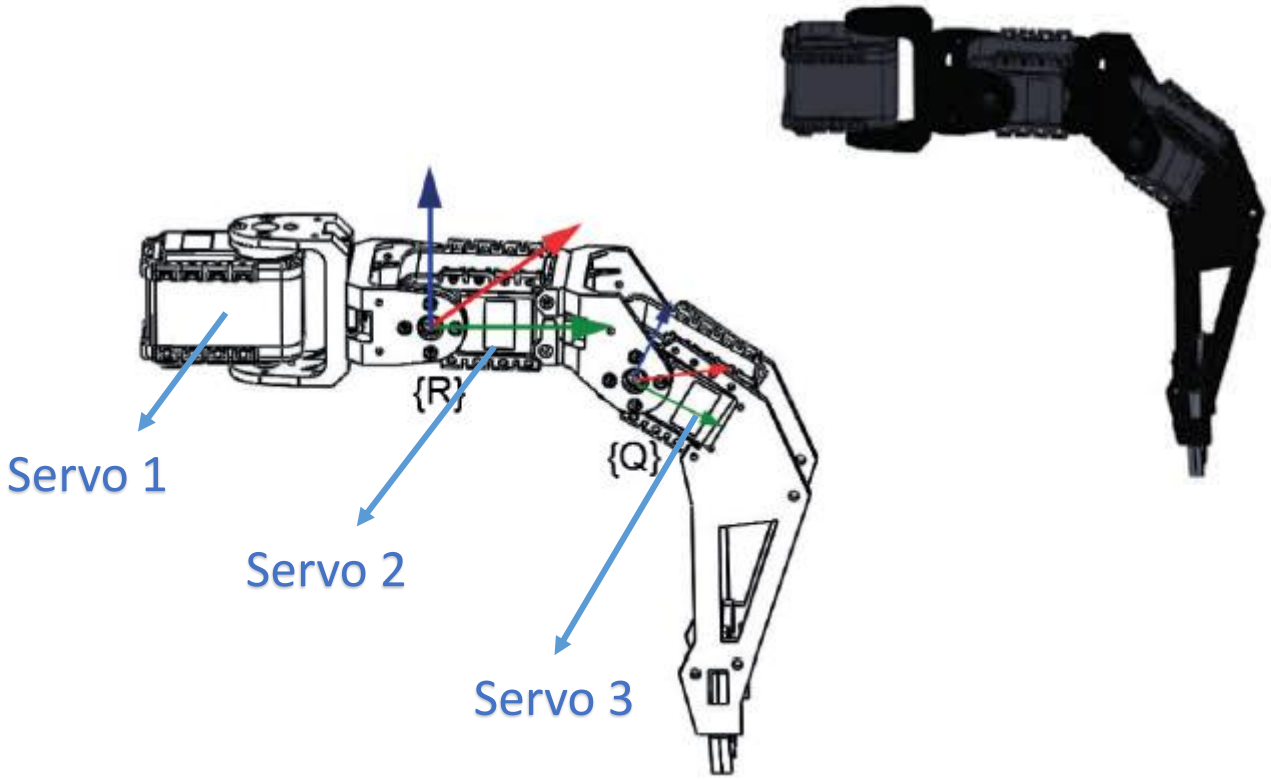




Programación y lógica de funcionamiento

Actuadores (actividad)

Analizar y anotar La secuencia de Acciones que debe hacer cada servomotor de una de las patas del hexápodo.

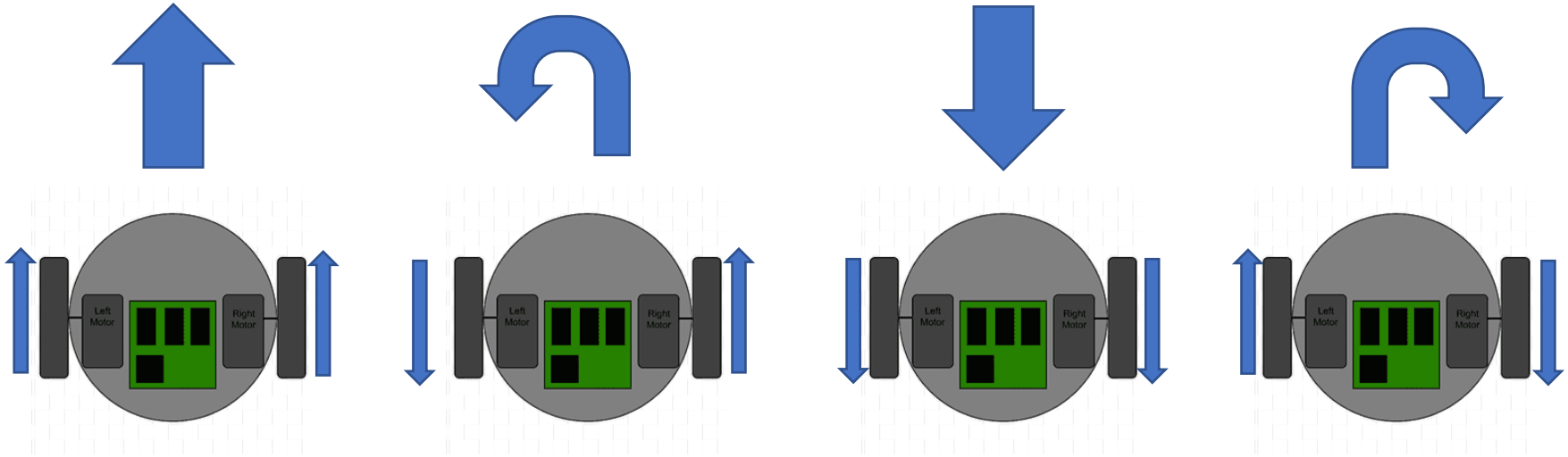


Programación y lógica de funcionamiento

Actuadores

La programación para la acción de un actuador tipo Rueda es bastante sencilla, ya que solo debe obedecer a parámetros como potencia o rapidez, sentido de giro y en algunos casos duración de la acción.

Realizar un algoritmo de movimiento para un robot diferencial de 2 ruedas puede ser sencillo, ya que únicamente se requieren cuatro tipos de movimientos para poder desplazar el equipo robótico.



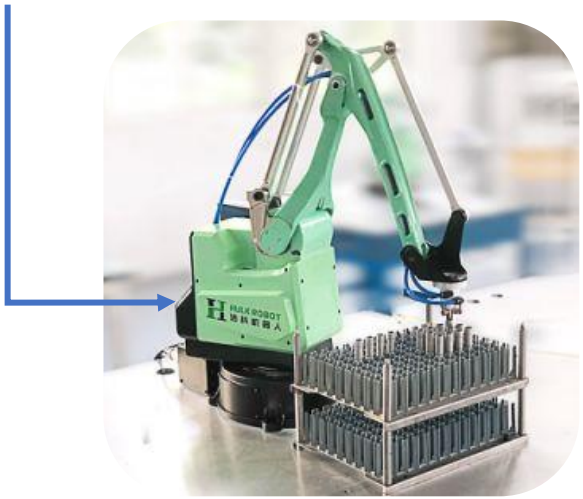
Programación y lógica de funcionamiento

Actuadores

De la mano de los actuadores es posible encontrar reductores y transmisiones, que se encargan de convertir la velocidad de giro de un motor a torque además de la transmisión de esta energía a algún punto necesario del robot.

Algunos robots sitúan su motor o actuador principal directamente conectado al eje funcional, sin utilizar transmisiones y reducciones intermedias que puedan sumar efectos no deseados a las cinemáticas, esto es llamado accionamiento directo.

Actuador principal



Actuador principal

Programación y lógica de funcionamiento

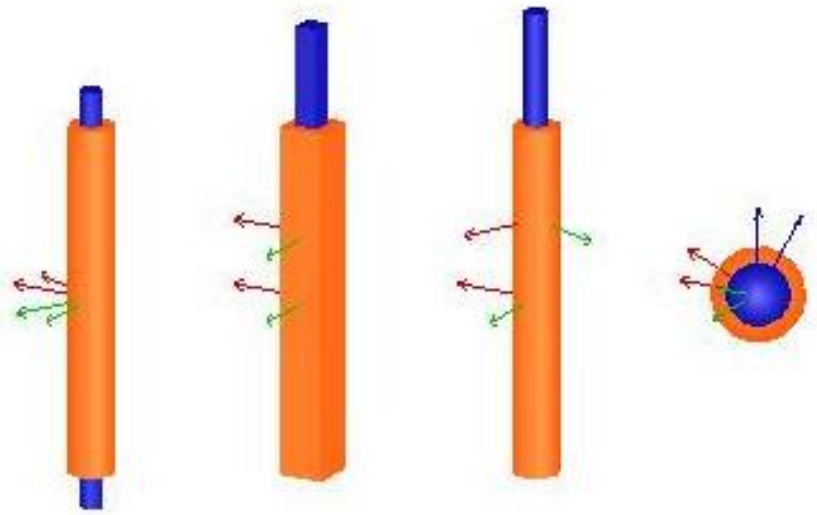
Simulador (Joint)

Para el caso del simulador CoppeliaSim El trabajo de los actores está representado por los denominados “Joint”, los cuales representan tipos de grados de libertad.

Estos elementos son configurados directamente con un determinado torque o potencia, sin necesidad de representar específicamente un sistema de reducción o transmisión.

Luego, podemos encontrar una clasificación sencilla de los tipos de “Joint” que CoppeliaSim ofrece:

Articulación giratoria, prismática, de tornillo y esférica.



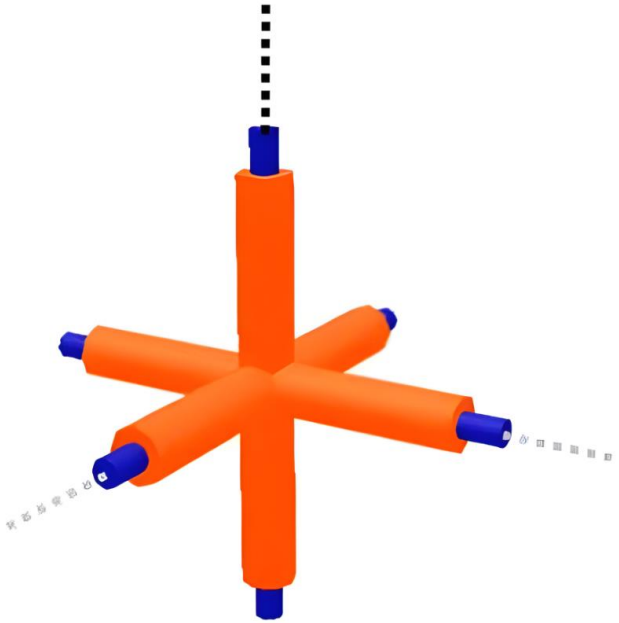
Programación y lógica de funcionamiento

Simulador (Joint)

Articulaciones giratorias:

Las articulaciones giratorias tienen un GDL y se utilizan para describir movimientos de rotación entre objetos. Su configuración está definida por un valor que representa la cantidad de rotación sobre el eje z de su primer marco de referencia. Se pueden utilizar como articulaciones pasivas o como articulaciones activas (motores).

Es posible configurar sencillamente la orientación de esta articulación para permitir un giro paralelo al plano X, o Y, o Z. inclusive planos inclinados.



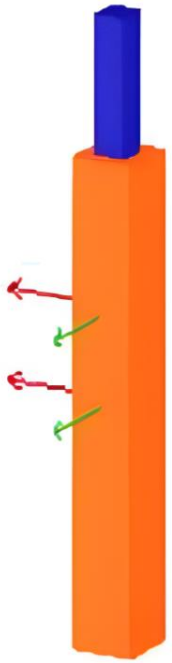
Programación y lógica de funcionamiento

Simulador (Joint)

Articulaciones prismáticas:

las articulaciones prismáticas tienen un GDL y se utilizan para describir movimientos de traslación entre objetos. Su configuración está definida por un valor que representa la cantidad de traslación a lo largo del eje z de su primer marco de referencia. Se pueden utilizar como articulaciones pasivas o como articulaciones activas (motores).

Es posible configurar sencillamente la orientación de esta articulación para permitir traslaciones paralelas al plano X, o Y, o Z. inclusive planos inclinados.



Programación y lógica de funcionamiento

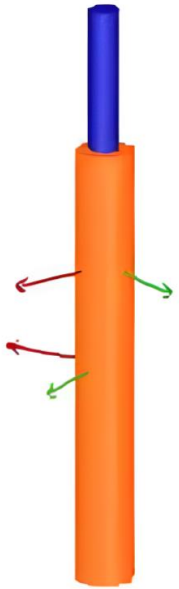
Simulador (Joint)

Tornillos:

Los tornillos, que pueden verse como una combinación de articulaciones giratorias y articulaciones prismáticas (con valores vinculados), tienen un GDL y se utilizan para describir un movimiento similar a un tornillo.

Un parámetro de tono define la cantidad de traducción para una cantidad dada de rotación.

Una configuración de tornillo se define por un valor que representa la cantidad de rotación sobre el eje z de su primer marco de referencia. Los tornillos se pueden utilizar como juntas pasivas o como juntas activas (motores).

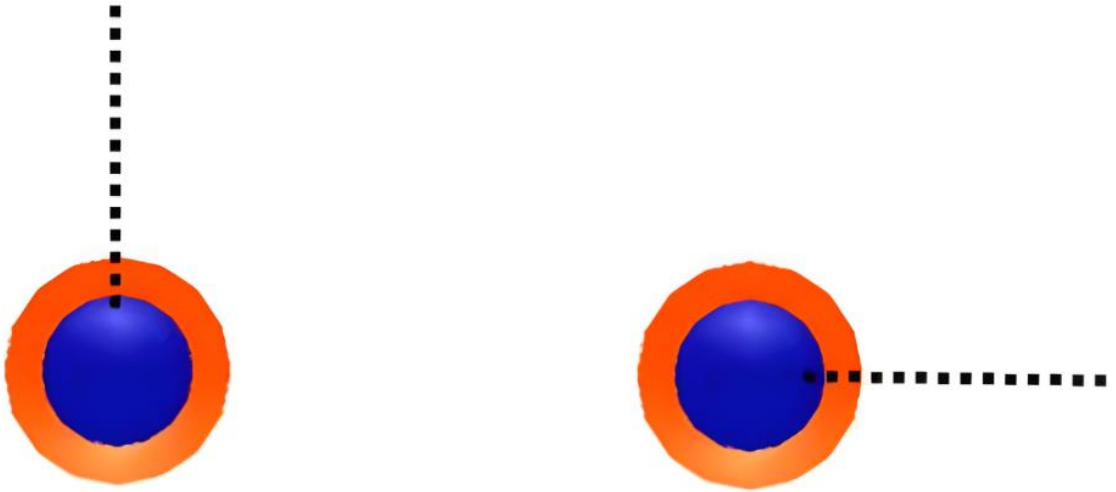


Programación y lógica de funcionamiento

Simulador (Joint)

Articulaciones esféricas:

las articulaciones esféricas tienen tres GDL y se utilizan para describir movimientos de rotación entre objetos. Su configuración está definida por tres valores que representan la cantidad de rotación alrededor de los ejes X, Y y Z de su primer marco de referencia. Los tres valores que definen la configuración de una junta esférica se especifican como ángulos de Euler. En algunas situaciones, una articulación esférica se puede considerar como 3 articulaciones concurrentes y ortogonales entre sí, que están agrupadas en una cadena jerárquica. Sin embargo, la analogía solo es válida mientras todas las articulaciones giratorias mantengan una orientación distinta de cualquiera de las otras dos: de hecho, si dos articulaciones se acercaran a coincidir, podría aparecer una situación singular y el mecanismo podría perder un GDL.



Programación y lógica de funcionamiento

CoppeliaSim (Joint)

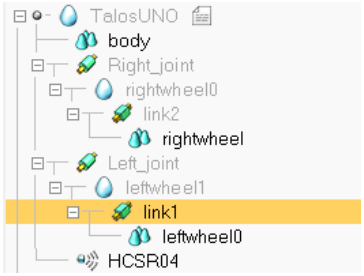
Agregar un objeto tipo “Joint” es bastante sencillo, solo se debe hacer clic derecho en el área de trabajo, luego agregar un objeto tipo joint y seleccionar el tipo.

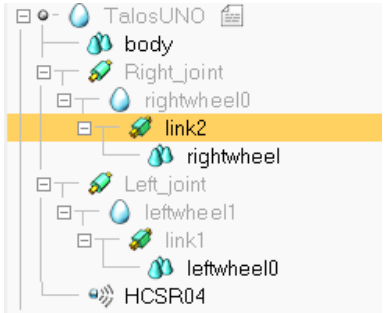
Algo importante es poder establecer la posición jerárquica que esté Joint ocupara, ya que son utilizados como uniones entre un objeto y otro. En efecto, un Joint puede unir la carrocería de un robot con su Rueda a través de una articulación tipo giratoria.

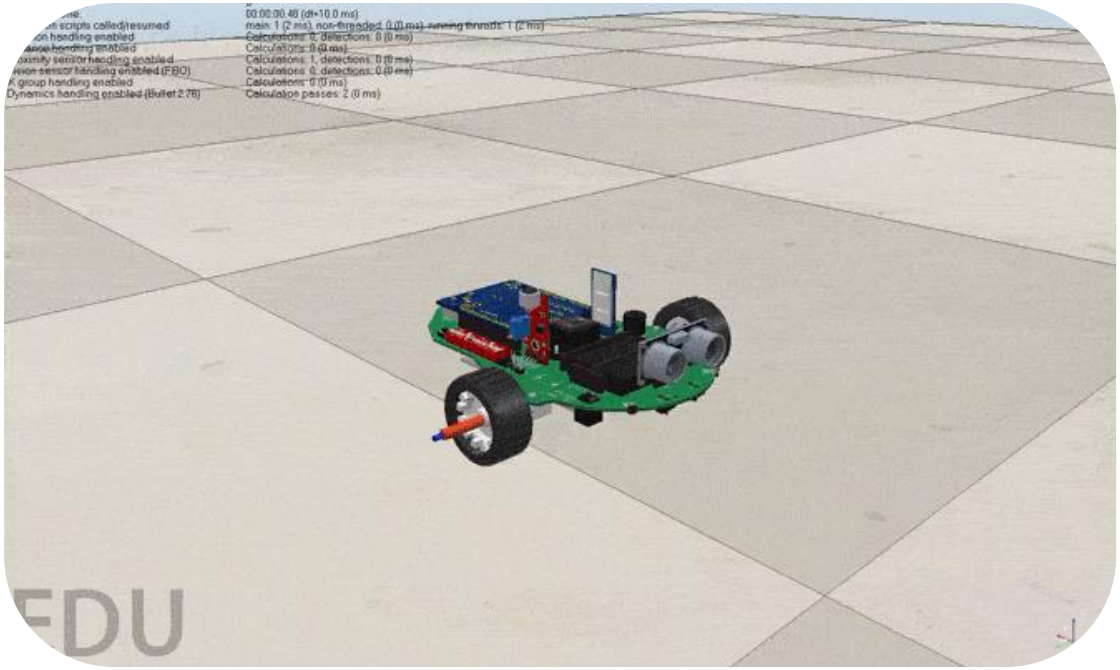
Programación y lógica de funcionamiento

CoppeliaSim (Joint)

En este ejemplo se puede observar articulación tipo giratoria uniendo las dos ruedas del robot diferencial. Notar dónde se debe ubicar el objeto tipo Joint dentro de la jerarquía de uniones del robot.





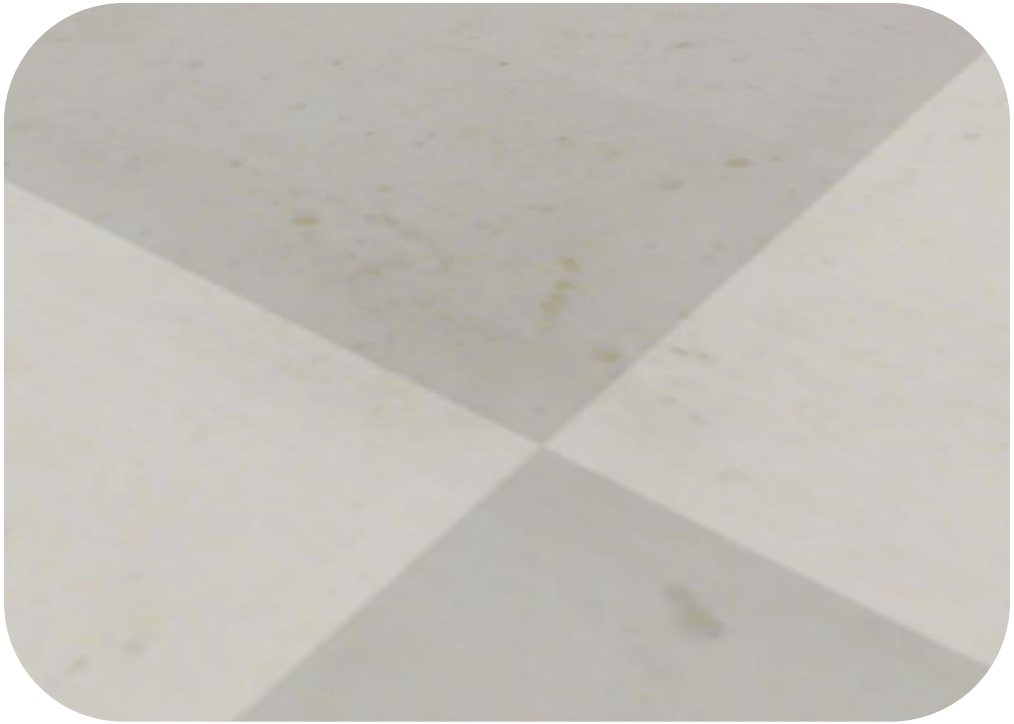


Programación y lógica de funcionamiento

CoppeliaSim – Objeto geométrico primitivo

La creación de un objeto geométrico primitivo es bastante sencillo, al igual que la creación de un objeto tipo joint, se debe hacer clic en el plano de trabajo para “Add” y luego “Primitive shape”.

Al crear el objeto, el software solicitará ingresar las dimensiones de este.

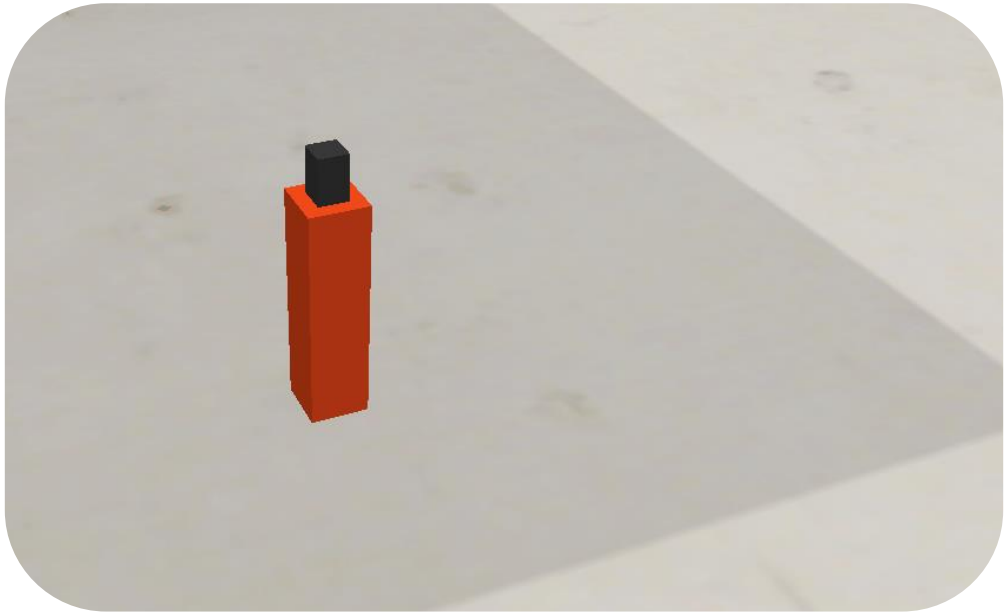


Programación y lógica de funcionamiento

CoppeliaSim (Joint – articulación prismática)

Luego de crear un objeto Joint tipo articulación primatica se debe configurar para que este pueda operar como actuador.

Al crear no es importante indicar que el actuador está habilitado.

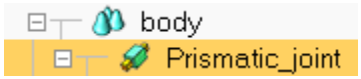


Programación y lógica de funcionamiento

CoppeliaSim (Joint – articulación prismática)

La programación del actuador es bastante sencilla, solo se debe asociar el nombre específico del actuador creado con una variable. Luego simplemente se debe indicar la posición que se quiere alcanzar.

La extensión del actuador prismático podrá ser establecida dentro de las propiedades de este objeto. Bajo la programación se podría alcanzar porcentualmente la extensión del actuador establecida.




```
piston=sim.getObject('./Prismatic_joint')
sim.setJointPosition(piston,1) --valor de 0 a 1 (porcentual)
```

Actividad de aplicación

Programación y lógica de funcionamiento

Actividad – Amazon Robot

En esta actividad se deberá implementar un actuador que pueda imitar el trabajo que realizan los robots en los almacenes de Amazon.

Los robots de Amazon permiten transportar racks o repisas con productos de un lugar del almacén a otro, a través de un pistón situado en la superficie del robot. Cuando el robot se sitúa abajo del rack, el pistón se eleva levantando el rack por completo y pudiendo transportarlo dentro de la planta.

Actividad de aplicación

Programación y lógica de funcionamiento

Actividad – Amazon Robot

Se busca modificar el robot presente en coppélia SIM “KUKA Omnirob”, el cual cuenta con cuatro ruedas independientes tipo omnidireccionales.

Se debe crear el objeto tipo Joint para situarlo en la superficie del robot jerárquicamente, de modo que si el robot se desplaza el objeto Joint también lo hará con la estructura.

Luego se debe crear un objeto geométrico primitivo tipo disco qué puede hacer adosado al otro extremo del Joint.

Finalmente se debe programar el robot para que pueda ubicarse debajo del rack, levantarlo y transportarlo hacia adelante.