

Práctica de Laboratorio N° 3

Objetivos principales

1. Implementar clases y objetos especiales para la representación gráfica de manipuladores robóticos.
2. Obtener parámetros para modelado de manipuladores robóticos utilizando el algoritmo de D-H.

Objetivos secundarios

1. Familiarizarse con el entorno de desarrollo de JupyterLab.
2. Aprender cómo acceder a la documentación de un paquete desde la línea de comandos de Python.
3. Inspeccionar el paquete robotics-toolbox para obtener información sobre su funcionamiento.
4. Desarrollar de manera exitosa los ejercicios propuestos.

Explorando el toolbox de robótica para Python

Introducción

Python es un lenguaje de programación de alto nivel conocido por su claridad y sencillez. Una de sus varias características destacables es la introspección^[1], que permite a los programas examinar el tipo o las propiedades de un objeto en tiempo de ejecución. Esto significa que un programa Python puede conocer la información sobre sí mismo y modificar su comportamiento en consecuencia.

Llevado a nuestro caso de uso en particular, si bien le daremos un uso muy acotado, esto nos permite averiguar sobre la marcha cómo está compuesto el toolbox, así como también cuáles son las funcionalidades disponibles. En esta práctica vamos a hacer un paso rápido por las funciones básicas utilizadas para fines introspectivos para poder averiguar información útil sobre el toolbox, para luego pasar a resolver los ejercicios con esta información que acabamos de adquirir.

Dado que es una estrategia que puede utilizarse de manera universal con otros paquetes, clases y funciones de Python, esto podría resultar de utilidad en otros campos de interés para el alumno, no necesariamente relacionados con el toolbox ni la robótica.

Uso de la función `help()`

Al invocarse sobre un paquete, esta función devuelve una guía integrada que proporciona información detallada sobre cómo utilizar los diferentes objetos, paquetes, funciones, clases y métodos disponibles en el mismo.

La función `help` se puede aplicar a paquetes completos, por ejemplo:

```
import roboticstoolbox as rtb
help(rtb)
```

Imprimirá algo similar a lo siguiente:

```
Help on package roboticstoolbox:
```

```
NAME
```

```
    roboticstoolbox
```

```
PACKAGE CONTENTS
```

```
    backends (package)
```

```
    bin (package)
```

```
    blocks (package)
```

```
    examples (package)
```

```
    fknm
```

```
    frne
```

```
    mobile (package)
```

```
    models (package)
```

```
    robot (package)
```

```
    tools (package)
```

Tener en cuenta que el contenido está truncado por brevedad, la salida real de este comando es mucho más extensa porque contiene *toda* la información de ayuda de todo el paquete, incluyendo cada uno de sus componentes.

Por este motivo, si bien existe una utilidad inmediata en invocar la función de ayuda para un paquete completo, es usual que uno vaya llamando a la ayuda para las clases y funciones específicas de interés.

En nuestro caso de uso de la práctica de hoy, nos interesa saber sobre los modelos de robots de ejemplo que se encuentran disponibles en el toolbox. El comando de arriba nos dio la información de que existe un sub-paquete llamado `models`, que parece ser lo que necesitamos. Para averiguarlo, invocamos la función de ayuda sobre dicho módulo:

```
help(rtb.models)
```

La cual imprimirá lo siguiente:

```
Help on package roboticstoolbox.models in roboticstoolbox:
```

```
NAME
```

```
    roboticstoolbox.models
```

```
PACKAGE CONTENTS
```

```
DH (package)
ETS (package)
URDF (package)
list
```

FUNCTIONS

```
list(keywords=None, dof=None, type=None, border='thin')
```

Display all robot models in summary form

```
:param keywords: keywords to filter on, defaults to None
:type keywords: tuple of str, optional
:param dof: number of DoF to filter on, defaults to None
:type dof: int, optional
:param type: model type "DH", "ETS", "URDF", defaults to all types
:type type: str, optional
```

- ``list()`` displays a list of all models provided by the Toolbox. It lists the name, manufacturer, model type, number of DoF, and keywords.
- ``list(type=MT)`` as above, but only displays models of type ``MT`` where ``MT`` is one of "DH", "ETS" or "URDF".
- ``list(keywords=KW)`` as above, but only displays models that have a keyword in the tuple ``KW``.
- ``list(dof=N)`` as above, but only display models that have ``N`` degrees of freedom.

The filters can be combined

- ``list(keywords=KW, dof=N)`` are those models that have a keyword in ``KW`` and have ``N`` degrees of freedom.

DATA

```
__all__ = ['list', 'ETS', 'DH']
```

FILE

```
c:\users\apoyo\anaconda3\envs\robotica\lib\site-
packages\roboticstoolbox\models\__init__.py
```

Ah , podemos llamar al m todo `list()`, as  sin argumentos y eso nos deber a devolver la lista de modelos de robots disponibles en el toolbox. Probemos:

```
rtb.models.list()
```

Ejercicios

Ejercicio 1

Utilizando las clases *Link* y *SerialLink* y sus respectivos métodos construya y represente en forma gráfica el manipulador cuyos parámetros de D-H se muestran en la tabla 1. Luego compare su resultado con la figura 1.

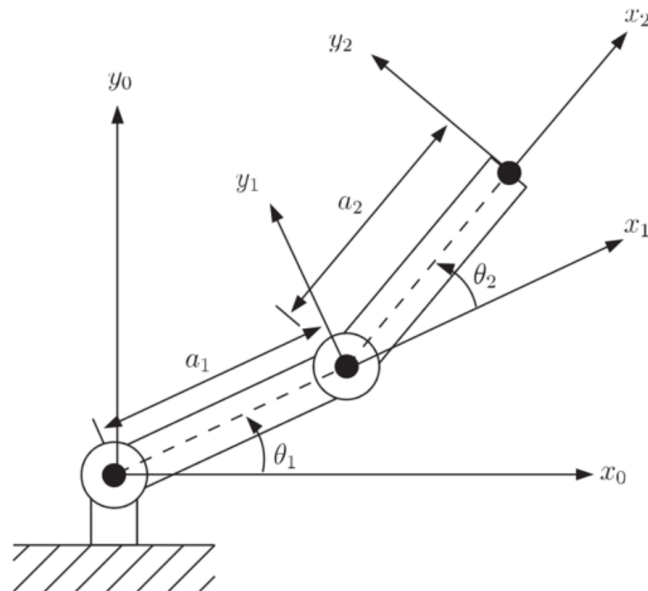


Figura 1. Robot de 2GDL

Tabla 1. Configuración de robot de 2GDL

Articulación	θ	d	a	α
1	q_1	0	1	0
2	q_2	0	1	0

Ejercicio 2

Utilizando las clases *Link* y *SerialLink* y sus respectivos métodos construya y represente en forma gráfica el manipulador cuyos parámetros de D-H se muestran en la tabla 2. Luego compare su resultado con la figura 2.

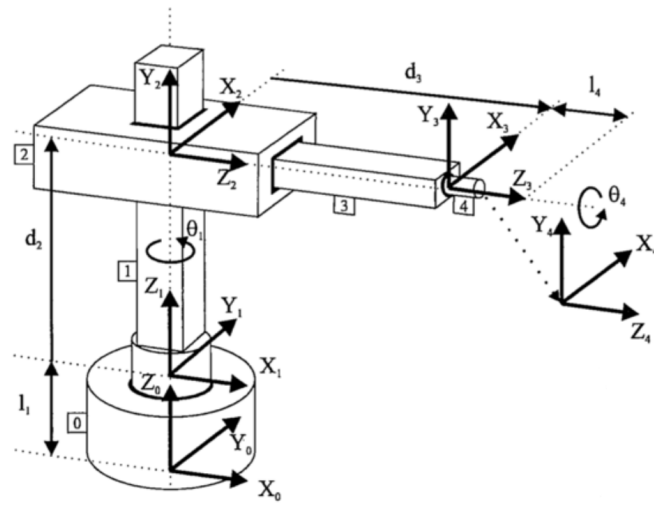


Figura 2. Robot de 4GDL

Tabla 2. Configuración de robot de 4GDL

Articulación	θ	d	a	α	Σ
1	q_1	l_1	0	0	0
2	90	d_2	0	90	0
3	0	d_3	0	0	0
4	q_4	l_4	0	0	0

Ejercicio 3

Siga las instrucciones:

- Invoque el modelo del manipulador FANUC10L utilizando el comando `mdl_fanuc10l`.
- Identifique la clase correspondiente al robot en el workspace y obtenga la lista de parámetros de D-H correspondiente al manipulador.
- Utilizando el método `plot` represente el robot en forma gráfica con todas las articulaciones en la posición cero.
- Experimente con el método `teach` para interactuar con la representación gráfica.

Ejercicio 4

Siga las instrucciones:

- Invoque el modelo del manipulador Cobra600 utilizando el comando `mdl_cobra600`.
- Identifique la clase correspondiente al robot en el workspace y obtenga la lista de parámetros de D-H correspondiente al manipulador.
- Utilizando el método `plot` represente el robot en forma gráfica con todas las articulaciones en la posición cero.
- Experimente con el método `teach` para interactuar con la representación gráfica.

[1] https://es.wikipedia.org/wiki/Introspecci%C3%B3n_de_tipos