

## Guía número 3: Cinemática de robots manipuladores.

Unidad de Aprendizaje 3:  
Robótica Industrial.

Aprendizaje Esperado  
3.1.- Verifica el funcionamiento de sistema robótico, de acuerdo con requerimiento. (Integrada Innovación).

### Actividades

1. Simulación de cinemática directa e inversa.

## I. Presentación

La siguiente guía de laboratorio, consta de una aplicación de cinemática directa e inversa de un robot manipulador de dos grados de libertad.

La simulación fue realizada en el software Matlab con el toolbox de robótica de Peter Corke.

### Criterios de evaluación:

Las actividades consideran los siguientes criterios de evaluación.

- 3.1.1.- Analizando el comportamiento de los sensores, de acuerdo a la estrategia de control de trayectoria del robot.
- 3.1.2.- Analizando actuadores apropiados a la aplicación del robot mediante la ejecución de una trayectoria.
- 3.1.3.- Aplicando métodos para obtención de cinemáticas directa e inversa, de acuerdo a morfología del robot.
- 3.1.4.- Identificando la información necesaria y las diversas variables involucradas.

## II. Instrucciones

1. En el equipo de trabajo, desarrollar las actividades, usando herramientas matemáticas, software de simulación y editor de ecuaciones de MS Office o similar.

## III. Ejemplo de aplicación

### 2. Simulación de cinemática de un robot.

#### Paso 2.1: Denavit y Hartenberg

- 2.1.1. Considere el robot planar de la figura, el cual corresponde a un robot planar de 2 GDL (grados de libertad), determine:

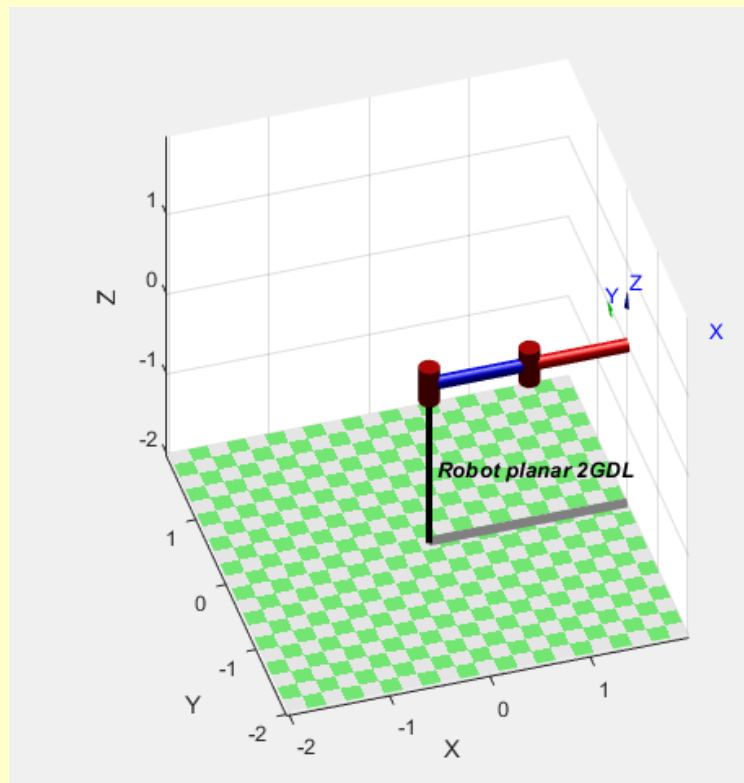


Figura 1: Robot planar de 2 grados de libertad.

- 2.1.2. Para poder simular un robot en Matlab, es necesario instalar el toolbox de robótica de Peter Corke, el cual es gratuito y no requiere licencia, se adjunta un enlace para más información:  
<https://petercorke.com/toolboxes/robotics-toolbox/>
- 2.1.3. Los parámetros de la matriz de Denavit y Hartenberg se entregan a continuación:

$i$	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
i=1	$\theta_1(t)$	0	L1	0
i=2	$\theta_2(t)$	0	L2	0

2.1.4. La línea de comandos para simular la matriz de D-H se presenta a continuación:

```
%% DH RR Robotics toolbox 2018a (ver 9.10)
clear all
clc
%% Valor de los eslabones
L1=1;
L2=1;
%% Formato para utilizar el toolbox
% Create Link using this code
% L = Link ( [ Th d a alph] )
% condiciones iniciales iguales a cero
% th1(0)=0;
% th2(0)=0;
%% DH del primer grado de libertad
L(1)= Link ( [0 0 L1 0] );
%% DH del segundo grado de libertad
L(2)= Link ( [0 0 L2 0] );
%% Union de los tres eslabones
R=SerialLink (L);
%% Nombre del robot
R.name = 'Robot planar 2GDL';
%% Simulacion del Robot RR
R.teach
```

Figura 2: Comandos DH robot RR.

Paso 2.2: Comprobación de la cinemática directa e inversa

2.2.1. Compruebe los cálculos de la cinemática directa e inversa, los scripts correspondientes se presentan a continuación:

```

%% Funcion embebida Cinematica Directa
function posiciones = fcn(angulosq)
%% Vector de entrada q
angth1=angulosq(1);
angth2=angulosq(2);
%% Largo de los eslabones
L1=1;
L2=2;
%% Ecuaciones de la cinematica
posx=L1*cos(angth1) +L2*cos(angth1+angth2);
posy=L1*sin(angth1) +L2*sin(angth1+angth2);
%% Funcion resultante
posiciones=[posx;posy];

```

Figura 3: Cinemática robot RR

```

%% Funcion embebida Cinematica Inversa
function angulos = fcn(pos)
%% Vector de entrada
xe=pos(1);
ye=pos(2);
%% Largo de los eslabones
L1=1;
L2=2;
%% Ecuaciones de la cinematica inversa
anguloq2=acos(((xe^2) + (ye^2) - (L1^2) - (L2^2)) / (2*L1*L2));
anguloq1=atan2((ye*(L1 +L2*cos(anguloq2)) -xe*L2*sin(anguloq2)), ((xe*(L1 +L2*cos(
(anguloq2)) +ye*L2*sin(anguloq2))));
%% Funcion de salida
angulos=[anguloq1;anguloq2];

```

Figura 4: Cinemática inversa RR

3. El diagrama de bloques de simulink se muestra a continuación.

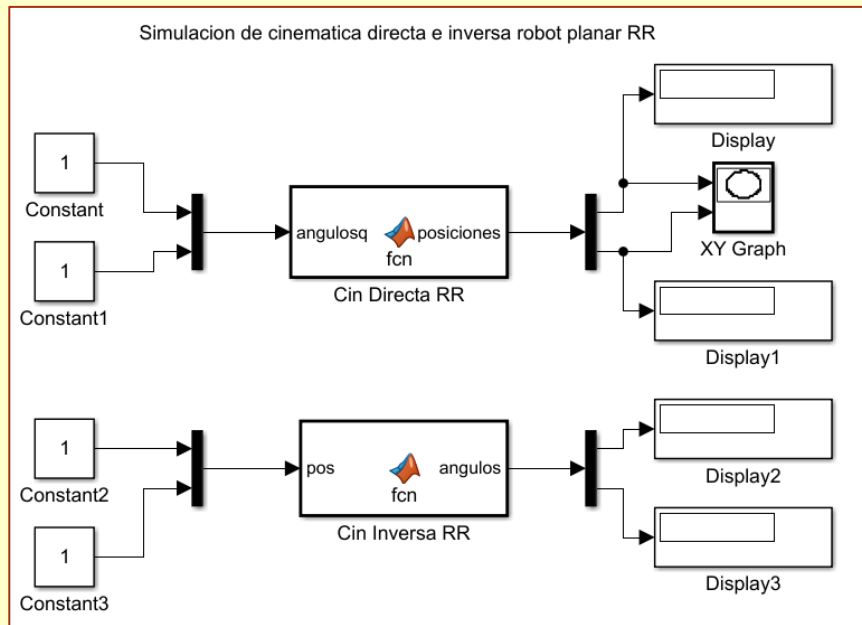


Figura 5: Diagrama de bloques

Verifique la cinemática directa e inversa del robot, utilice la simulación D-H y compare posición del efector final y los ángulos de cada articulación.