Marta Rosa Flores // Daniel Triano Moreno

ampliación de robótica

Memoria de prácticas: Manipuladores

Índice:

[Práctica 1: Planificación cartesiana - cuaternios. 2](#_Toc73778955)

[Apartado 1.1: 2](#_Toc73778956)

[Apartado 1.2: 4](#_Toc73778957)

[Apartado 1.3: 5](#_Toc73778958)

[Práctica 2: Control de fuerzas indirecto 7](#_Toc73778959)

[Apartado 2.1 7](#_Toc73778960)

[Apartado 2.2 10](#_Toc73778961)

[Apartado 2.3 12](#_Toc73778962)

[Práctica 3: Linealización por realimentación 15](#_Toc73778963)

[Apartado 3.1 15](#_Toc73778964)

[Apartado 3.2 16](#_Toc73778965)

[Práctica 4: Implantación de control de fuerzas indirecto 19](#_Toc73778966)

[Apartado 4.1 19](#_Toc73778967)

[Apartado 4.2 19](#_Toc73778968)

[Apartado 4.3 20](#_Toc73778969)

[Práctica 5: Ampliación de caminos (A\*) 21](#_Toc73778970)

[Apartado 5.1 21](#_Toc73778971)

[Apartado 5.2 21](#_Toc73778972)

[Apartado 5.3 21](#_Toc73778973)

# Práctica 1: Planificación cartesiana - cuaternios.

## Apartado 1.1:

**Definir la función de interpolación por cuaternios basada en el método de Taylor qr=qpinter(q1,q2,t) que realice el cálculo del cuaternio intermedio entre q1 (inicial) y q2 (final). El valor t debe cumplir 0≤t≤1, de modo que [q1]=qpinter(q1,q2,0) y [q2]=qpinter(q1,q2,1).**

Para realizar la planificación cartesiana, debemos realizar la interpolación de la matriz homogénea y en este caso se pide que la realicemos por el método de cuaternios basada en el método de Taylor. Este método es generalmente usado porque simplifica los cálculos y logra movimientos más uniformes. Para aplicarlo necesitamos de una localización inicial y una final, que serán P1 y P2 respectivamente, tal y como se observa en la fig X.

Se puede alcanzar P2 partiendo de P1 y viceversa ejecutando una traslación y una rotación del sistema, trataremos de implementar dichos conceptos de forma matemática.

Se define la traslación del sistema en función del tiempo desde el P1 hacia P2, tal y como se observa en la ecuación X.

Definimos la rotación del sistema partiendo de la orientación 1 hacia la orientación 2 en base a la ecuación X:

Donde el valor de viene definido por la siguiente ecuación:

**Para calcular el eje de giro de nuestro cuaternio, se calcula q para** , tal y como se muestra en la ecuación X.

Conociendo estas ecuaciones, se procede a su implementación en MATLAB tal y como se aprecia en la Figura 1.

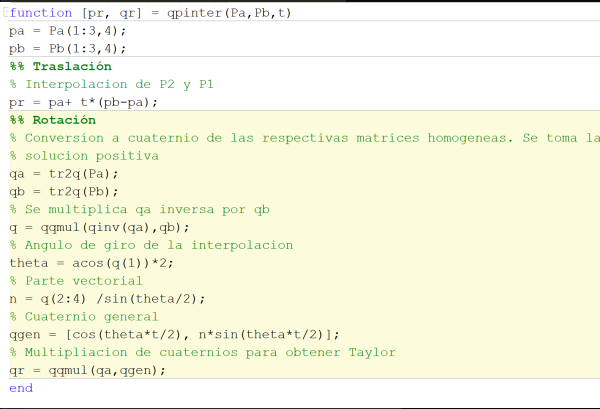


Figura : Función qpinter.

En primer lugar, se leen las orientaciones de ambos puntos, y se realiza la interpolación entre ambos puntos, utilizando la ecuación X, a continuación, se convierten a cuaternios ambas orientaciones, tomando la solución positiva por defecto. Seguidamente, se multiplican los cuaternios de cada punto y se calculan tanto el ángulo de giro como la parte vectorial.

Finalmente, se calcula el cuaternio general en función del tiempo y se multiplica por el cuaternio del punto de partida para obtener el cuaternio Taylor.

Se hace uso de la función qinv que calcula la inversa de un cuaternio tal y como se expone a continuación en la Figura 2.

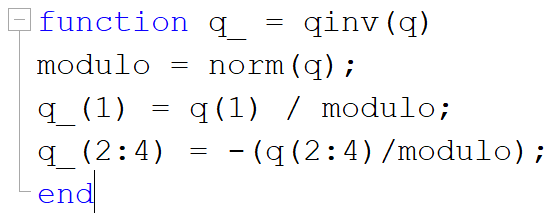
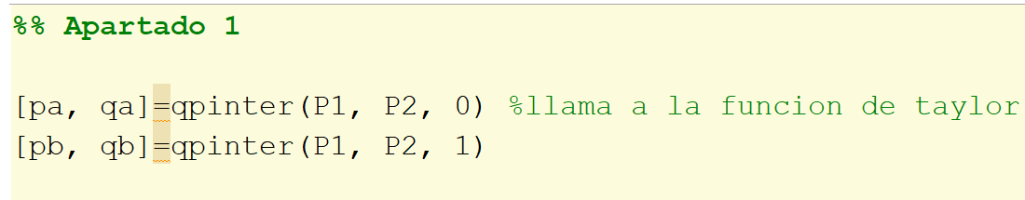


Figura - Función qinv que invierte un cuaternio

Se implementa la función qpinter en el script de la práctica y se ejecuta tal y como se observa en la Figura 3.



Figura

El resultado es el observado en la Figura 4.

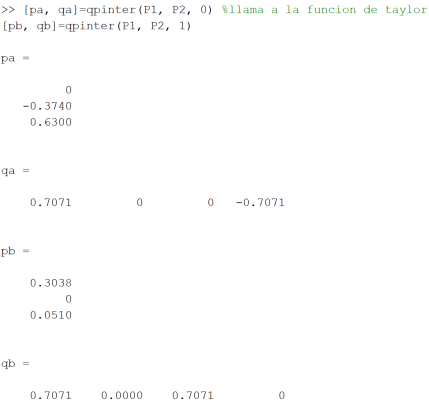


Figura : Resultado tras ejecutar la función Taylor.

## Apartado 1.2:

**Realizar una función MATLAB con el formato P=tramo(P0,P1,P2,tau,T,t) que calcula la transformación P correspondiente al movimiento de P0 a P2 vía P1 suavizado por el método de Taylor. Los parámetros tau y T, corresponden respectivamente al intervalo de transición y tiempo total empleado en recorrer el camino tal y como se muestra en la figura 1, y t indica el tiempo en el cual se alcanza la localización del camino P calculada.**

Para poder realizar este apartado, se hará uso de las ecuaciones X y X.

Se crea la función “tramo”, tal y como se observa en la Figura 5. En esta función se calculan los distintos cuaternios de Taylor iniciales dependiendo del tramo en el que nos encontramos gracias a las funciones condicionales que se detallan en el código.

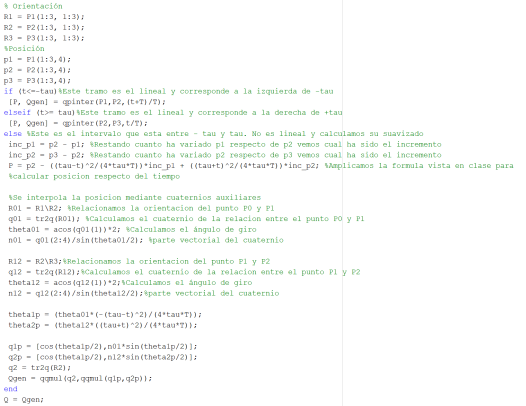


Figura – Script de la función tramo.

Una vez se han reconocen las distintas regiones de trabajo, se procede a implementar las ecuaciones X, X.

## Apartado 1.3:

**Representar gráficamente la evolución de la posición y la orientación (en ángulos de EulerZYZ) a lo largo de toda la trayectoria.**

Se ejecuta el script proporcionado por el enunciado para comprobar la correcta ejecución de las funciones implementadas, obteniendo los resultados mostrados en las siguientes figuras.

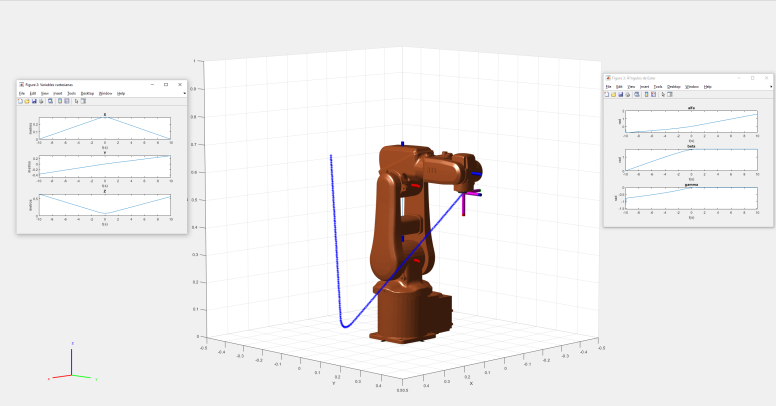


Figura : Trayectoria seguida por el robot

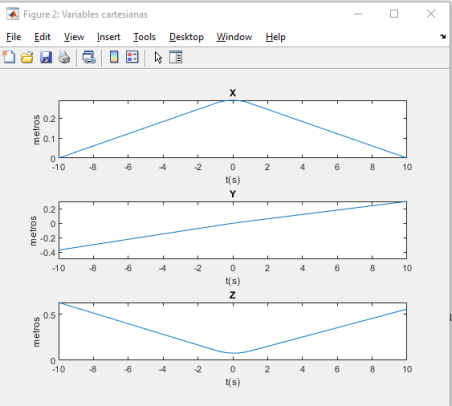


Figura : Variación de la trayectoria en variables cartesianas

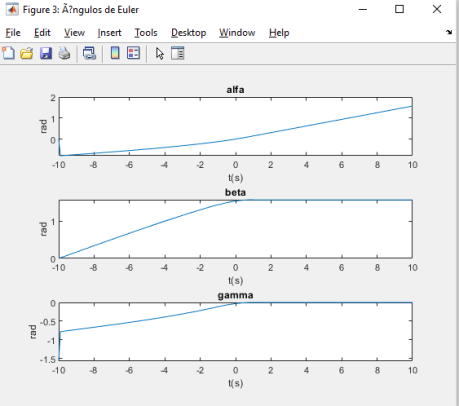


Figura : Variación de la trayectoria en ángulos de Euler ZYZ

# Práctica 2: Control de fuerzas indirecto

## Apartado 2.1

**Si se considera una elasticidad del entorno sobre el eje X igual a kx=103 N/m, calcular la frecuencia natural no amortiguada y el coeficiente de amortiguación del modelo dinámico equivalente del robot en contacto con el entorno. Al tomar kx=104 N/m, calcular los nuevos valores de los parámetros dinámicos.**

En este ejercicio, se realizará un control de fuerzas resistivo utilizando las ecuaciones dinámicas dadas en el enunciado para aplicar la ley de control sobre una superficie suave y elástica.

En primer lugar, se han considerado las ecuaciones presentes en el campus virtual y replicadas a continuación.

|  |
| --- |
|  |
|  |

Introducciendo dichas ecuaciones en MATLAB (ver fig) y utilizando los valores numéricos proporcionados en el enunciado, obtenemos los siguientes resultados.

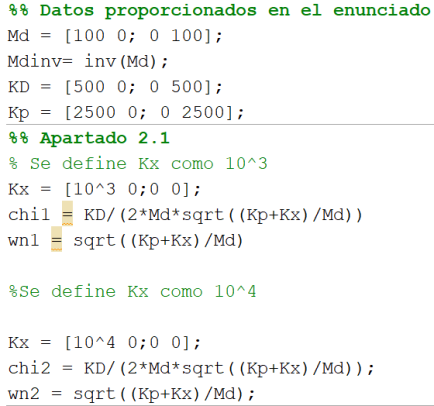


Figura 9: Cálculo

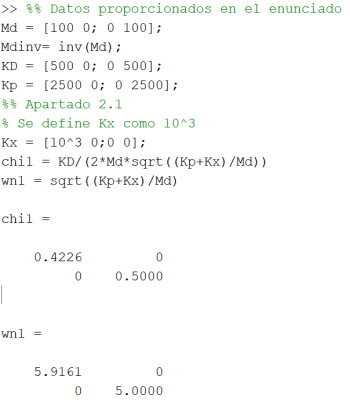


Figura 10: Resultados obtenidos para la frecuencia natural y el coeficiente de amortiguación para Kx=103

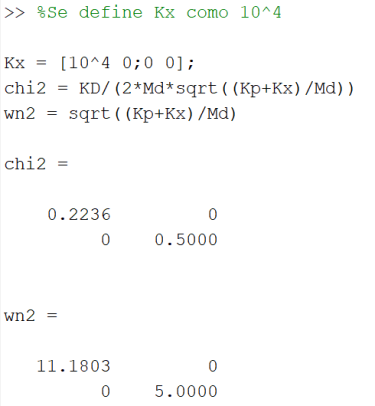


Figura 11: Resultados obtenidos para la frecuencia natural y el coeficiente de amortiguación para Kx=103

## Apartado 2.2

**Construir un modelo en Simulink para realizar la simulación dinámica del manipulador en contacto con el entorno, mediante el uso del modelo equivalente, si la posición de contacto es [1, 0], la deseada es [1’1, 0’1] con los dos valores de elasticidad del entorno considerados en el apartado primero.**

Para este apartado, despejamos la ecuación dada por el enunciado y se he reproducido en Simulink, tal y como se aprecia en Figura 12.

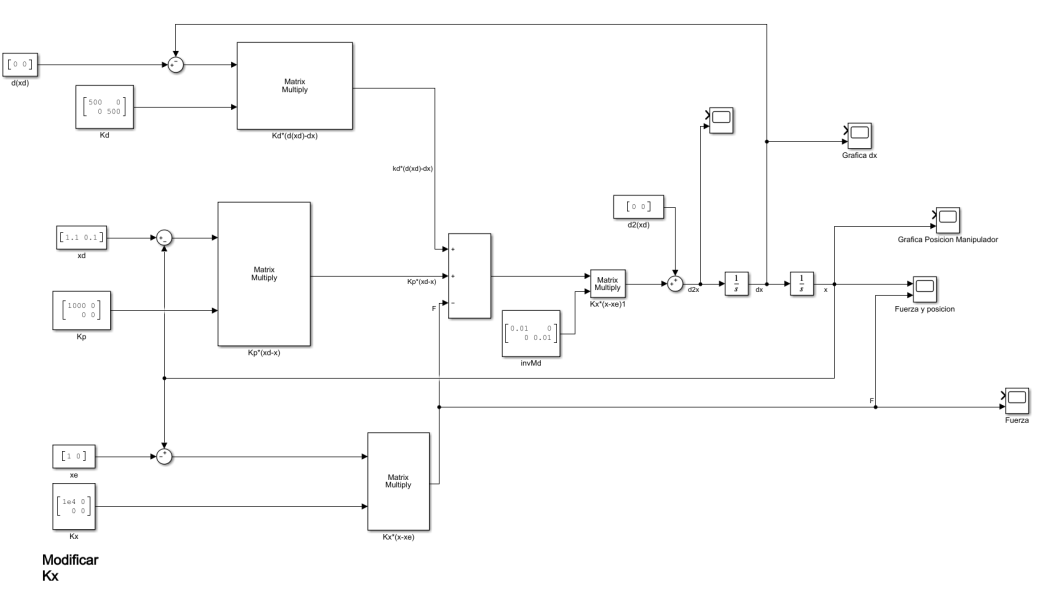


Figura 12: Simulación de la ecuación del manipulador.

Se han simulado las respuestas y estas se observan en las fig. X, X. Como podemos ver, se alcanza régimen estacionario en la salida tal y como deseábamos. Se han simulado las posiciones para Kx=103 y Kx=104.



Figura 13: Posición del manipulador con Kx=103



Figura 14: Posición del manipulador con Kx=104

Para el segundo caso, se observa una mayor sobreoscilación que en el primero y un mayor error en régimen permanente. También se aprecia cómo en el segundo caso se tarda más en lograr la estabilidad.

## Apartado 2.3

**Calcular un conjunto de parámetros del controlador que verifiquen que el comportamiento transitorio del sistema se encuentre críticamente amortiguado. ¿Existe un conjunto único de valores?**

Establecemos el valor de como condición para que el sistema sea críticamente amortiguado. Teniendo en cuenta las siguientes ecuaciones, despejamos los valores de Kp y Kd.

En el siguiente script de MATLAB, utilizamos la primera de las ecuaciones con un valor de .

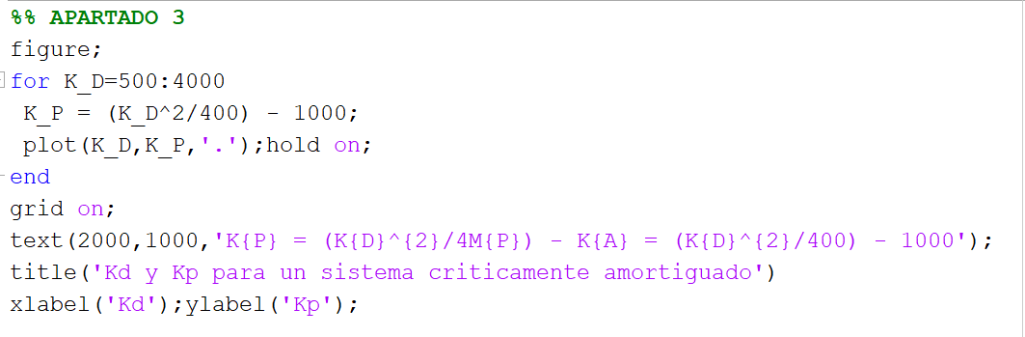


Figura 15

El resultado se aprecia en la Figura 16. Un rango notable de las soluciones obtenidas es negativo, por lo tanto, se ha decidido acotar el sistema para obtener valores de Kp y Kd positivos. Para ello se ha implementado el código mostrado en la fig X.

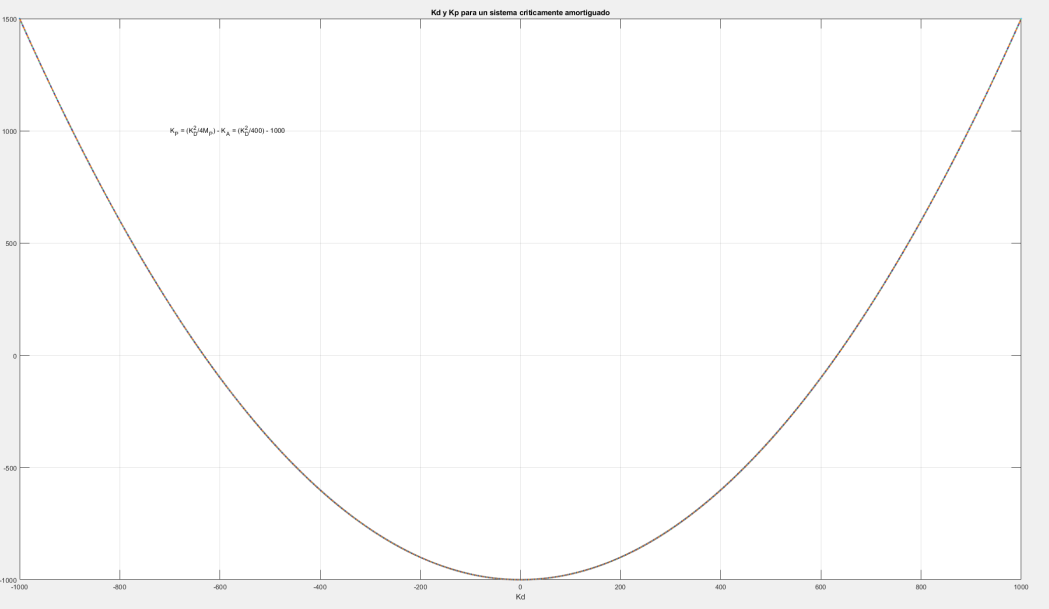


Figura 16

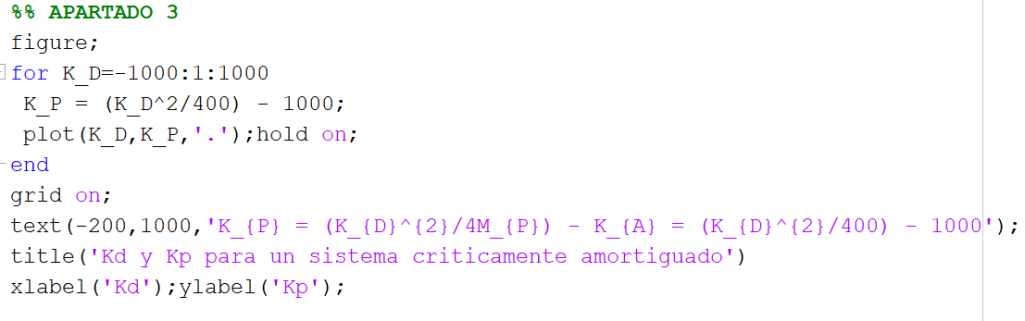


Figura 17

En la fig X, se observa el resultado de ejecutar dicho código, y se muestra un rango de soluciones positivas.

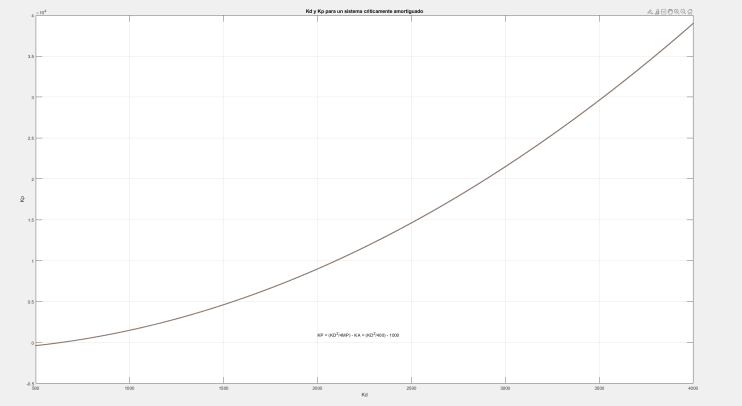


Figura 18

# Práctica 3: Linealización por realimentación

## Apartado 3.1

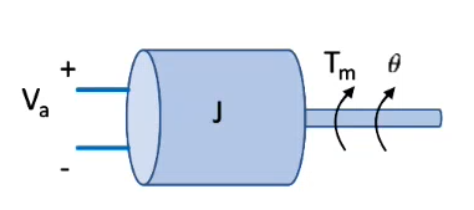
**Obtener las expresiones de los elementos 𝐵(𝜃) y 𝑁(𝜃, 𝜃̇) cuando el manipulador detallado en la figura 1 posee incorporado los motores y los reductores, de manera que la entrada 𝜏𝑢 coincidirá con la tensión Va de entrada a los actuadores del manipulador. Se empleará Kr como el parámetro de la reductora, Kv como la constante de fuerza contra-electromotriz, Kt como constante de conversión tensión-par y Ra la resistencia de la armadura.**

En este apartado, se utiliza el modelo de control resistivo basado en la linealización por realimentación según la expresión X. El objetivo de esta técnica consiste en transformar una dinámica no lineal en una que sí lo sea, permitiendo aplicar a posteriori herramientas reservadas a sistemas lineales.

𝜏𝑢 = 𝐵(𝜃)𝑦 + 𝑁(𝜃, 𝜃̇)

La matriz B es la matriz de masas y N el conjunto de los efectos no lineales del mecanismo, siendo τu el par de entrada del manipulador. El objetivo de esta t

En primer lugar, y basándonos en la Figura 19, vemos el esquema del motor junto con los parámetros que le afectar.



Figura

Obtenemos la ecuación que determina la tensión de entrada del motor , que depende de la tensión que cae en la resistencia Ra, la que cae en el bobinado del motor, y , que es la tensión inducida en el estator.

Despreciamos el valor de la tensión que cae en el bobinado puesto que es un valor muy pequeño, y teniendo en cuenta la EC X, se obtiene la EC X.

|  |
| --- |
|  |
|  |

Nuestro objetivo es relacionar el par de entrada con la tensión , para ello usamos la EC X, y despejando a partir de la EC X expresada arriba, obtenemos la expresión EC X.

Una vez tenemos la ecuación que relaciona el par de salida y la tensión de entrada, podemos aplicar la ecuación dada por el enunciado, teniendo en cuenta además los valores de la reductora Kr, que nos relaciona las posiciones de la forma indicada en las ec xxx.

|  |
| --- |
|  |
|  |
|  |

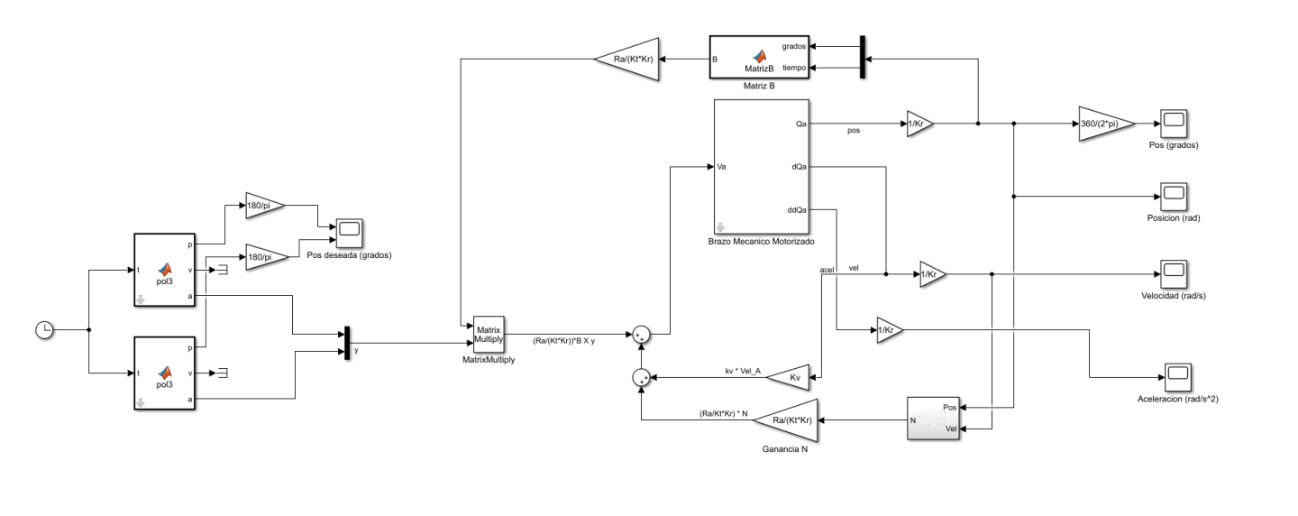
Despejando, haciendo uso de las ecuaciones anteriores, obtenemos la siguiente ecuación en la que se muestra el valor de la tensión de entrada.

En esta ecuación, se pueden resaltar los valores de B y , como nos pedían en el enunciado de la práctica.

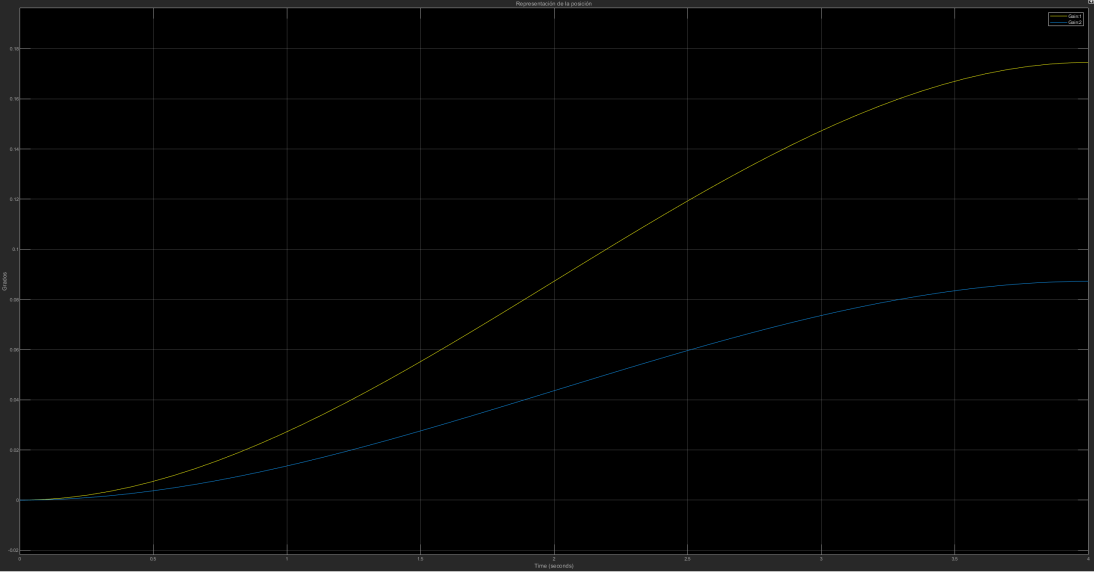
## Apartado 3.2

**Simular el esquema de control diseñado a partir de la Figura 1 para un manipulador RR con el objeto de llegar a la referencia articular (10,5), expresado en grados, en un tiempo de 4 segundos (utilizar una interpolación de la posición por polinomios cúbicos). Verificar que efectivamente se sigue la trayectoria deseada.**

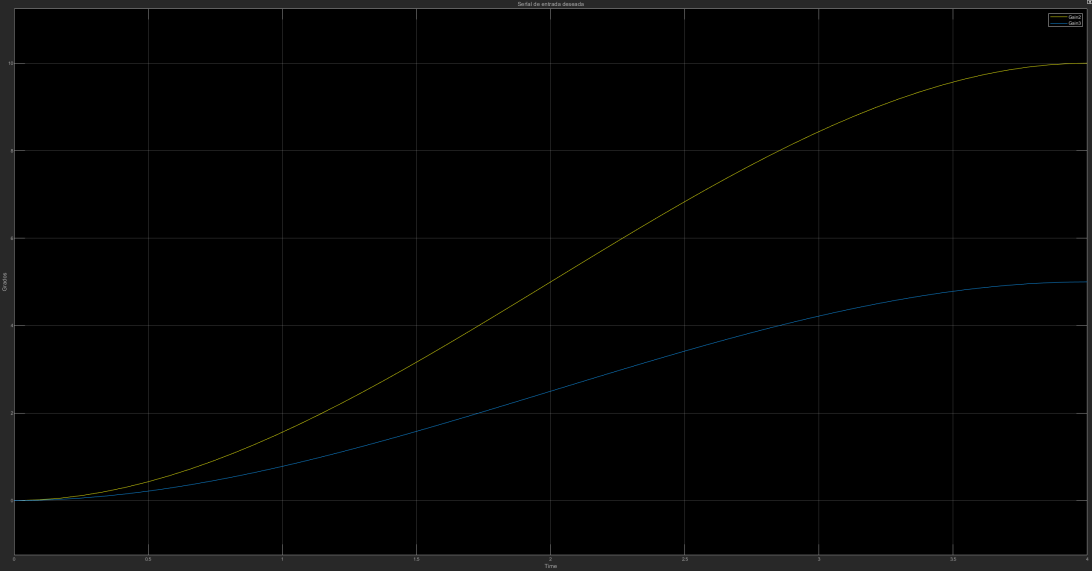
En la Figura 20 se muestra el diseño implementado en Simulink, que se simula durante 4 segundos para poder observar la trayectoria generada y la seguida por el manipulador, que se muestran en la Figura 21 y Figura 22. Como podemos ver, son iguales, por lo que nuestro modelo funciona correctamente.



Figura



Figura



Figura

# Práctica 4: Implantación de control de fuerzas indirecto

## Apartado 4.1

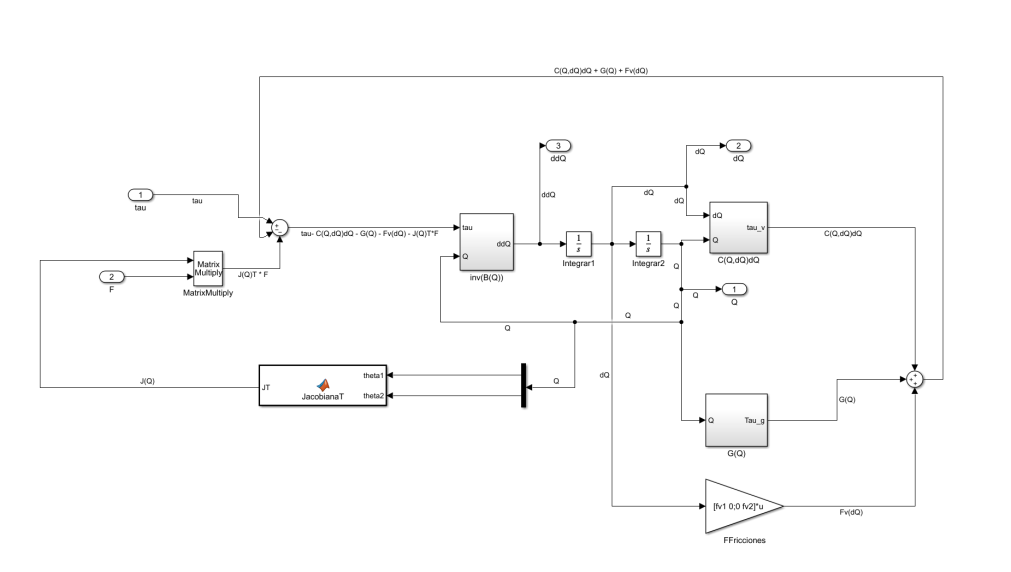
**Construir un subsistema que contenga el robot RR motorizado y compensado por la linealización por realimentación. El mencionado subsistema, se modificará para que contenga además la compensación de fuerzas, y para que sus entradas sean variables cartesianas en vez de articulares.**

Partiendo de ecuación vista en la práctica anterior, vamos a añadir el término , que es el producto de la matriz jacobiana traspuesta por la fuerza. Se puede observar en la siguiente ecuación.

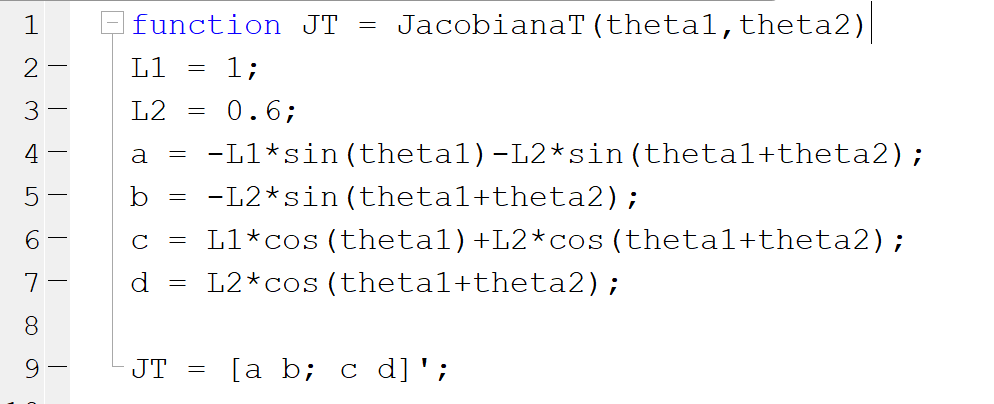
Esta ecuación es la que describe el comportamiento del robot compensado.

En primer lugar, y basándonos en el diseño Simulink previo, vamos a introducir la ecuación X, en la que se observa el comportamiento del manipulador tras añadir las fuerzas externas en el modelo dinámico.

La implementación de dicha ecuación se observa en la Figura 23, donde se hace uso de una función denominada JacobianaT, que realizará la traspuesta de la jacobiana, cuyo código se incluye a continuación (ver Figura 24).



Figura



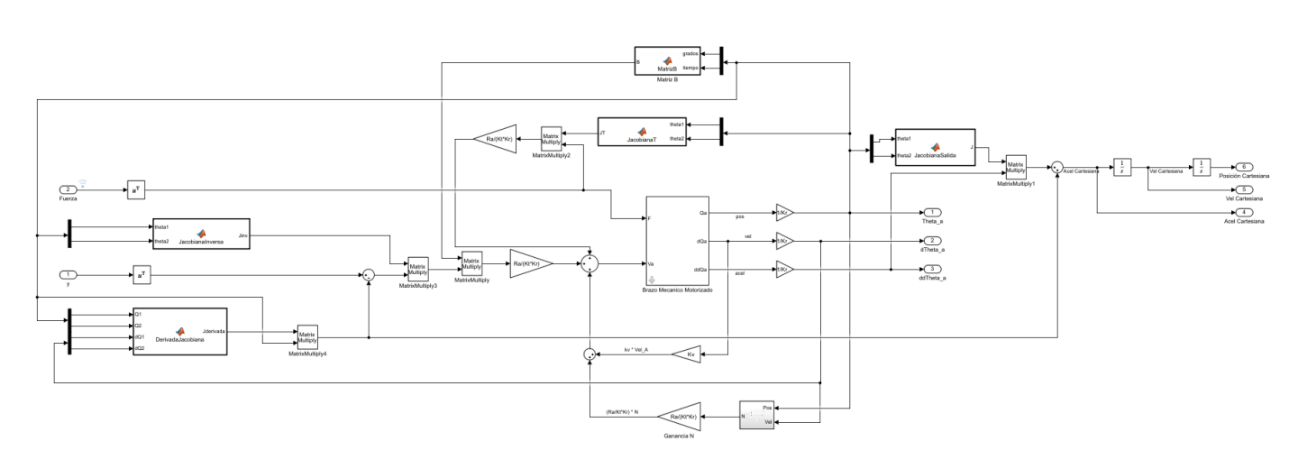
Figura

Para poder realizar la función JacobianaT, se ha seguido el siguiente procedimiento. En primer lugar, se ha calculado el modelo cinemático directo mediante las ecuaciones X, X

|  |
| --- |
|  |
|  |

Una vez conocido el valor de x e y, se calcula la jacobiana de la siguiente manera, esta ecuación es la utilizada para crear el código utilizado en la FIG X.

Una vez, hemos hecho el cálculo de la jacobiana, podemos implementar el esquema global del robot compensado siguiendo la EC ENUNCIADO, y obteniendo el esquema que se observa en la siguiente Figura 25.

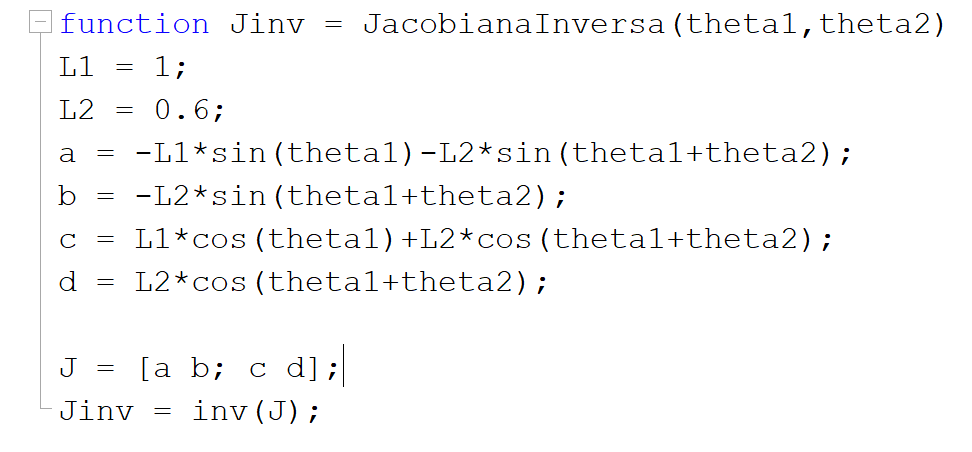


Figura

Tal y como se observa en la figura de arriba, no solo se ha introducido este bloque para realizar la suma, si no que se han añadido distintas funciones que realizan cambios entre velocidades articulares y cartesianas.

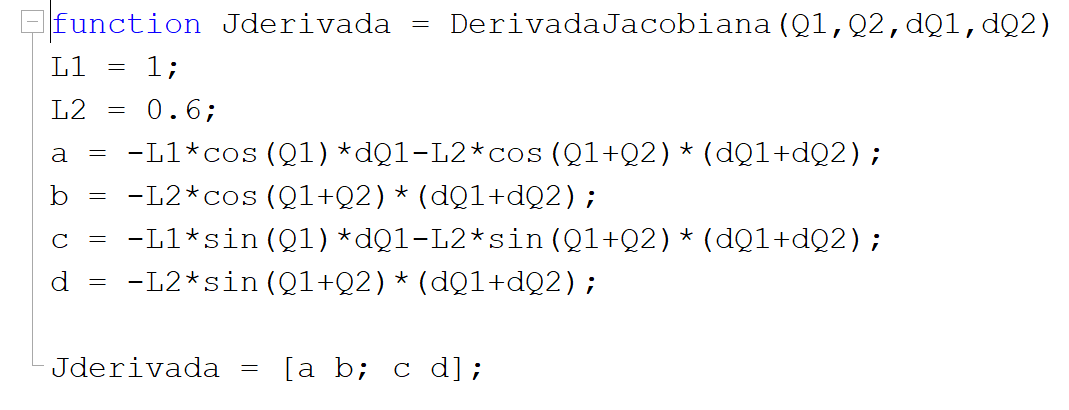
Para poder calcular la relación entre ambas velocidades, se han empleado las siguientes fórmulas, que describen la relación entre velocidades articulares y cartesianas.

Como se puede observar, para poder implementar estas fórmulas, se necesita tanto la derivada de la jacobiana como su inversa, cuyo cálculo se realiza con el código de la Figura 26, que será el introducido al inicio del sistema para poder trabajar en velocidades articulares.



Figura

Por último, justo a la salida final del sistema, tenemos que volver a convertir las velocidades de articulares a cartesianas, realizándose mediante el código de la Figura 27.



Figura

## Apartado 4.2

**Adecuar el esquema de la figura 1 para que contenga el subsistema desarrollado en el punto anterior. Se simulará su comportamiento según los parámetros de control Md, KD y KP dados en la práctica 2 y con un coeficiente de elasticidad del entorno Kx=103N/m. El manipulador parte de la posición inicial (y de contacto) dada por ϴ1=45˚ y ϴ2=-45˚, y como posición deseada sumar 0.1 a cada coordenada. Comparar los resultados obtenidos con los resultados de utilizar la ecuación equivalente.**

## Apartado 4.3

**Proponer una modificación al esquema de control presentado para que la entrada al sistema de control sea la fuerza deseada que se quiere ejercer sobre el entorno.**

# Práctica 5: Ampliación de caminos (A\*)

## Apartado 5.1

**A partir de la ecuación equivalente del robot en contacto con la superficie, calcular la frecuencia natural no amortiguada y el coeficiente de amortiguación si KA en el eje X toma el valor de 103 N/m y CF en el mismo eje el valor 0,0024.**

## Apartado 5.2

**Realizar en SIMULINK la simulación dinámica del manipulador en contacto con el entorno, mediante el uso del modelo equivalente, si la posición de contacto es la utilizada en la práctica anterior y se desea ejercer la fuerza FR=[10, 0] N. Simular el sistema en la mencionada situación.**

## Apartado 5.3

**Deducir el nuevo modelo equivalente del robot en contacto con la superficie, implementar su modelo en SIMULINK y simular el comportamiento dinámico cuando en los valores para el eje X de KF y KI son respectivamente 0,00064 y 0,0016. Estudiar si se ha introducido alguna mejora en el comportamiento del sistema.**