

CONTROL Y PROGRAMACIÓN DE ROBOTS

Proyecto de robots manipuladores

Grado en Ingeniería Electrónica, Mecatrónica y Robótica



Autores: Lozano Romero, Daniel
Mérida Floriano, Javier
Montes Grova, Marco Antonio

Índice

1. Introducción al proyecto	3
2. Análisis Cinemático del brazo	4
2.1. Modelo Cinemático Directo	4
2.1.1. Parámetros y estudio del MCD según Denavit-Hartenberg	4
2.1.2. Matrices de transformación homogéneas del robot	4
2.1.3. Ecuaciones simbólicas del MCD	5
2.2. Modelo Cinemático Inverso	5
2.3. Jacobiano del robot y análisis de puntos singulares	6
3. Control Cinemático del robot	8
3.1. Tipos de Trayectorias Generadas	8
3.2. Interpoladores	10
3.3. Pruebas y conclusiones	11
4. Análisis Dinámico del brazo	13
4.1. Modelado Dinámico e Incertidumbres	13
4.2. Obtención de los parámetros dinámicos del robot	16
4.2.1. Identificación y Linealización de parámetros dinámicos	16
4.2.2. Estimación estadística de los parámetros linealmente independientes	19
4.3. Parámetros y modelos obtenidos	21
4.3.1. Robot medidas ideales con reductoras	22
4.3.2. Robot medidas ideales sin reductoras	23
4.3.3. Robot medidas reales con reductoras	24
4.3.4. Robot medidas reales sin reductoras	25
4.4. Verificación de los modelos obtenidos	26
4.4.1. Robot medidas ideales con reductoras	26
4.4.2. Robot medidas ideales sin reductoras	27
4.4.3. Robot medidas reales con reductoras	29
4.4.4. Robot medidas reales sin reductoras	30
5. Control Dinámico del brazo	32
5.1. Obtención del modelo lineal de las articulaciones del brazo	32
5.2. Diseño de controladores	33
5.2.1. Controlador PD/PID	33
5.2.2. Controlador PD/PID con compensación de gravedad	33
5.2.3. Controlador PD/PID con compensación de dinámica (Feedforward)	34
5.2.4. Controlador PD/PID con par calculado	35
5.3. Análisis de experimentos para probar controladores	36
5.3.1. Comparativa controladores PD-PID ideales con reductoras	37
5.3.2. Comparativa controlador PID y PD con compensación gravedad	39
5.3.3. Comparativa del control con compensación de dinámica o <i>FeedForward</i>	40
5.3.4. Comparativa del control par calculado	43
6. Anexos	47
6.1. Códigos de programación	47
6.1.1. Implementación de controladores	47
6.2. Generadores de trayectorias	48
6.3. Estimación de parámetros	48

Resumen

En las páginas que siguen, se realizará un análisis cinemático y dinámico de un robot de 3 grados de libertad, cuyas articulaciones serán todas de rotación. Además de ello, debido a que únicamente se considerarán conocidas las longitudes del robot, se deberá realizar una estimación de ciertos parámetros dinámicos del robot, como pueden ser inercias o fricciones viscosas. Finalmente, se diseñará un control cinemático y dinámico del brazo robótico.

1. Introducción al proyecto

El robot que se va a estudiar, representado por la estructura de la portada de este trabajo, posee los siguientes parámetros estructurales:

$$L_0 = 0,60m \quad L_1 = 0,60m \quad L_2 = 1,00m \quad L_3 = 0,80m$$

Además, también vienen dados los valores de factores de reducción de velocidad y constantes de par de los motores:

$$R_1 = 50; R_2 = 30; R_3 = 15$$

$$K_{t1} = 0,5 \frac{Nm}{A}; K_{t2} = 0,4 \frac{Nm}{A}; K_{t3} = 0,35 \frac{Nm}{A}$$

En primer lugar, se desarrollará un análisis cinemático del brazo robótico, tanto directo como inverso, es decir, se hallará la posición cartesiana del efector final en función de las variables articulares del brazo y viceversa.

Tras ello, será necesario estimar los parámetros dinámicos del brazo, ya que únicamente se conocen las longitudes del mismo. Se obtendrán diferentes parámetros en función de las condiciones supuestas para el brazo, principalmente las condiciones que se emplearan serán:

- Robot con medidas ideales con reductoras.
- Robot con medidas ideales de accionamiento directo.
- Robot con medidas reales con reductoras.
- Robot con medidas reales de accionamiento directo.

Para el robot con medidas reales, se evaluará qué condición es mas favorable. Por ejemplo, si se obtiene un mejor modelo matemático del robot empleando únicamente encoders que conozcan las posiciones de las variables articulares y derivando numéricamente para conocer las velocidades y las aceleraciones de las mismas, o el caso en el que se empleen encoders que conozcan la posición y tacómetros que conozcan las velocidades pero derivando numéricamente para obtener las aceleraciones.

Una vez se hayan obtenido los modelos matemáticos de las articulaciones de los robots, se diseñará un control cinemático del mismo, el cual será un generador de trayectorias que, a partir de una posición inicial, una posición real, un número de puntos y el tiempo en que se desea que vaya de un punto a otro, generará una trayectoria de las variables articulares del robot.

Por último, se diseñarán una serie de controladores dinámicos para el robot, en los cuales se observe la mejoría o empeoramiento de los mismos en función del tipo de robot empleado frente a otros y del tipo de experimento realizado.

Además de ello, éste análisis dinámico servirá para evaluar la bondad de los modelos obtenidos para las diferentes condiciones.

2. Análisis Cinemático del brazo

2.1. Modelo Cinemático Directo

2.1.1. Parámetros y estudio del MCD según Denavit-Hartenberg

Uno de los modos de estudio del problema cinemático directo de un robot es el procedimiento de Denavit-Hartenberg, el cual se basa en la realización de cambios de sistema de referencia empleando las matrices de transformación homogéneas. Siguiendo una serie de pasos, se llegará a obtener los siguientes parámetros que definen la cinemática directa del robot.

Por tanto, los sistemas de referencia definidos se muestran a continuación:

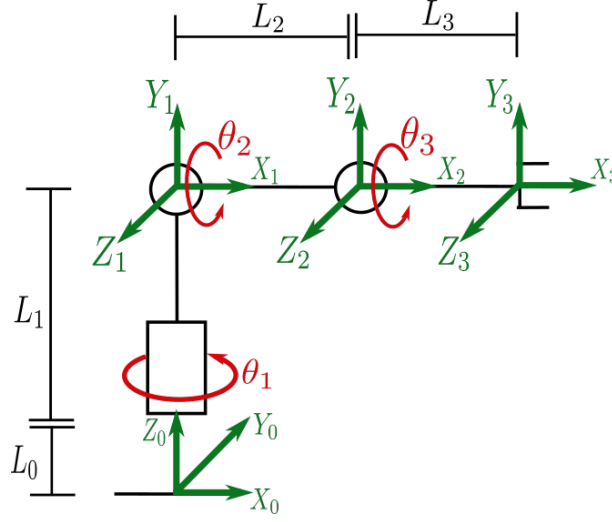


Figura 1: Sistemas de referencia definidos

En base a esos sistemas de referencia, los parámetros de Denavit-Hartenberg obtenidos serán:

Articulación	θ_i	d_i	a_i	α_i
1	θ_1	$L_0 + L_1$	0	$\frac{\pi}{2}$
2	θ_2	0	L_2	0
3	θ_3	L_3	L_2	0

2.1.2. Matrices de transformación homogéneas del robot

A continuación, se mostrarán las matrices de transformación homogéneas que definen los cambios de sistema de referencia que han hecho posible relacionar el sistema de referencia base con el del efector final. La matriz de transformación homogénea que relaciona un sistema de referencia con el siguiente se define como:

$${}^{i-1}A_i = Rot_z(\theta_i) * T(0, 0, d_i) * T(d_i, 0, 0) * Rot_x(\alpha_i) = \begin{pmatrix} \cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \sin(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\sin(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

Ahora que se ha definido la matriz general, se definirán las matrices de transformación homogéneas de cada cambio de sistema de referencia del robot:

$${}^0A_1 = \begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & L_0 + L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^1A_2 = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & L2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & L2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^2A_3 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & L3\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & L3\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2.1.3. Ecuaciones simbólicas del MCD

A partir de estos parámetros será posible obtener las matrices de transformación homogéneas asociadas a cada traslación y giro de sistema de referencia. Si se premultiplican las matrices desde la base hasta el punto final del brazo (efector final), la matriz de transformación obtenida será:

$${}^3T_0 = \begin{pmatrix} \cos(\theta_2 + \theta_3)\cos(\theta_1) & -\sin(\theta_2 + \theta_3)\cos(\theta_1) & -\sin(\theta_1) & \cos(\theta_1)[L3\cos(\theta_2 + \theta_3) + L2\cos(\theta_2)] \\ \cos(\theta_2 + \theta_3)\sin(\theta_1) & -\sin(\theta_2 + \theta_3)\sin(\theta_1) & \cos(\theta_1) & \sin(\theta_1)[L3\cos(\theta_2 + \theta_3) + L2\cos(\theta_2)] \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & L0 + L1 + L3\sin(\theta_2 + \theta_3) + L2\sin(\theta_2) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

Donde se puede extraer que la matriz de orientación del efector final y la posición del mismo respecto al sistema de referencia de la base son:

$$noa = \begin{pmatrix} \cos(\theta_2 + \theta_3)\cos(\theta_1) & -\sin(\theta_2 + \theta_3)\cos(\theta_1) & -\sin(\theta_1) \\ \cos(\theta_2 + \theta_3)\sin(\theta_1) & -\sin(\theta_2 + \theta_3)\sin(\theta_1) & \cos(\theta_1) \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 \end{pmatrix}$$

$$p = \begin{pmatrix} \cos(\theta_1)[L3\cos(\theta_2 + \theta_3) + L2\cos(\theta_2)] \\ \sin(\theta_1)[L3\cos(\theta_2 + \theta_3) + L2\cos(\theta_2)] \\ L0 + L1 + L3\sin(\theta_2 + \theta_3) + L2\sin(\theta_2) \end{pmatrix}$$

2.2. Modelo Cinemático Inverso

En lo que a la resolución del modelo cinemático inverso del robot respecta, se basará en obtener los valores de las variables articulares a partir de la posición del efector final. Por tanto, se partirá de las ecuaciones del vector p :

$$\begin{aligned} px &= \cos(\theta_1)[L3\cos(\theta_2 + \theta_3) + L2\cos(\theta_2)] \\ py &= \sin(\theta_1)[L3\cos(\theta_2 + \theta_3) + L2\cos(\theta_2)] \\ pz &= L0 + L1 + L3\sin(\theta_2 + \theta_3) + L2\sin(\theta_2) \end{aligned}$$

Para obtener las ecuaciones que definen el valor de θ_1, θ_2 y θ_3 , se trabajará con las funciones que definen px, py y pz . Será necesario definir los ángulos empleando la función atan2 de *Matlab*, ya que de ese modo se distinguirá en qué cuadrante se encuentra la tangente. Por tanto el objetivo es obtener el valor del seno y el coseno de cada ángulo.

Se comenzará obteniendo el valor de θ_1 . Para ello se define:

$$A = \sqrt{px^2 + py^2} = L3\cos(\theta_2 + \theta_3) + L2\cos(\theta_2)$$

de ese modo, se obtiene θ_1 como:

$$\theta_1 = \text{atan2}(\sin(\theta_1), \cos(\theta_1)) = \text{atan2}\left(\frac{py}{A}, \frac{px}{A}\right) \quad (3)$$

Para obtener el valor de θ_3 , se comenzará a partir de pz y, se definirá la siguiente variable auxiliar:

$$B = pz - L1 - L0 = L3\sin(\theta_2 + \theta_3) + L2\sin(\theta_2)$$

Si se trabaja con las dos variables auxiliares creadas, A y B, y se crea la nueva variable C, se conseguirá obtener el valor del ángulo deseado:

$$C = A^2 + B^2 = L2^2 + 2L2L3\cos(\theta_3) + L3^2$$

Cómo último paso para obtener el valor de θ_3 , se despejará el valor del coseno del mismo:

$$(\cos(\theta_3)) = \frac{C - L2^2 - L3^2}{2L2L3}$$

Por tanto, θ_3 se define como:

$$\theta_3 = \beta - \text{atan2}(\sin(\theta_3), \cos(\theta_3)) = \beta - \text{atan2}\left(\pm \sqrt{1 - \left(\frac{C - L2^2 - L3^2}{2L2L3}\right)^2}, \frac{C - L2^2 - L3^2}{2L2L3}\right) \quad (4)$$

Por último, se hallará el valor de la variable articular θ_2 . Para ello, se partirá de la ecuación A y se desarrollará empleando la fórmula del coseno del ángulo doble. La ecuación quedará de la forma:

$$A = L3\cos(\theta_2 + \theta_3) + L2\cos(\theta_2) = L3[\cos(\theta_2)\cos(\theta_3) - \sin(\theta_2)\sin(\theta_3)] + L2\cos(\theta_2) = \\ L3\cos(\theta_2)\cos(\theta_3) - L3\sin(\theta_2)\sin(\theta_3) + L2\cos(\theta_2)$$

donde será de ayuda despejar la función A en una suma de un término por el seno de θ_2 y otro término por el coseno del mismo ángulo. De ese modo será posible intentar obtener dicho ángulo empleando un cambio a coordenadas polares:

$$A = \cos(\theta_2)[L3\cos(\theta_3) + L2] - \sin(\theta_2)[L3\sin(\theta_3)]$$

$\begin{aligned} \rho \cos(\alpha) &= L3\cos(\theta_3) + L2 \\ \rho \sin(\alpha) &= L3\sin(\theta_3) \end{aligned}$	$\begin{aligned} \rho &= \sqrt{(L3\sin(\theta_3))^2 + (L3\cos(\theta_3) + L2)^2} \\ \alpha &= \text{atan2}(L3\sin(\theta_3), L3\cos(\theta_3) + L2) \end{aligned}$
---	--

Sustituyendo en la ecuación A y empleando la fórmula del coseno del ángulo suma, se podría definir:

$$A = \rho \cos(\theta_2)\cos(\alpha) - \rho \sin(\theta_2)\sin(\alpha) \rightarrow A = \rho \cos(\theta_2 + \alpha)$$

Despejando el valor del coseno θ_3 se obtendrá:

$$\theta_2 = \text{atan2}\left(\pm \sqrt{1 - \left(\frac{A}{\rho}\right)^2}, \frac{A}{\rho}\right) - \alpha \quad (5)$$

De ese modo, la solución del problema cinemático inverso está formada por siguientes ecuaciones:

$$\begin{aligned} \theta_1 &= \text{atan2}(\sin(\theta_1), \cos(\theta_1)) = \text{atan2}\left(\frac{py}{A}, \frac{px}{A}\right) \\ \theta_2 &= \text{atan2}\left(\pm \sqrt{1 - \left(\frac{A}{\rho}\right)^2}, \frac{A}{\rho}\right) - \alpha \\ \theta_3 &= \beta - \text{atan2}(\sin(\theta_3), \cos(\theta_3)) = \beta - \text{atan2}\left(\pm \sqrt{1 - \left(\frac{C - L2^2 - L3^2}{2L2L3}\right)^2}, \frac{C - L2^2 - L3^2}{2L2L3}\right) \end{aligned}$$

2.3. Jacobiano del robot y análisis de puntos singulares

Aunque éste apartado no cabe en este proyecto, es interesante llevar a cabo el análisis de los puntos singulares del robot, los cuales se obtendrían hayando qué valores de las variables articulares del robot hacen nulo el jacobiano de velocidades, ya que esos valores de las variables articulares serán los que generen velocidades infinitas en las articulaciones del robot.

El jacobiano analítico de velocidades se define del siguiente modo:

$$J_a = \begin{pmatrix} \frac{\partial px}{\partial \theta_1} & \cdots & \frac{\partial px}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \psi}{\partial \theta_1} & \cdots & \frac{\partial \psi}{\partial \theta_n} \end{pmatrix} \quad (6)$$

sin embargo, debido a que no se empleará efector final y el robot posee 3 grados de libertad, no se tendrán en cuenta los ángulos de Euler que especifican la orientación del mismo, resultando la matriz jacobiana cuadrada.

La relación entre velocidades angulares y velocidades en el efector final se relacionan mediante el jacobiano inverso.

$$\begin{pmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{pmatrix} = J^{-1} \begin{pmatrix} \dot{px} \\ \vdots \\ \dot{\psi} \end{pmatrix} \quad (7)$$

Por definición, la inversa de una matriz se define como:

$$J^{-1} = \frac{(adj(J))^T}{det(J)}$$

Por ello, si el determinante es nulo, el jacobiano inverso valdrá infinito y, por extensión, las velocidades en las articulaciones también lo será.

3. Control Cinemático del robot

Una vez estudiado el análisis cinemático del modelo de brazo manipulador, y obtenidas las ecuaciones cinemáticas inversa y directa de este, se terminará de abordar el problema cinemático, al ser capaces de desarrollar el control sobre esta; para ello, se requiere poder generar trayectorias dentro del espacio articular, con tal de que el brazo manipulador pueda cumplir una ordenes de movimiento conocidas.

Las ordenes de movimiento, en primer lugar conocidas, tendrán que especificar aquellas posiciones del espacio de trabajo, que cumplan con la trayectoria del movimiento exigido, este primer apartado del control cinemático serán los generadores de trayectorias

3.1. Tipos de Trayectorias Generadas

■ GENERADOR DE TRAYECTORIAS LINEAL

Este primer generador de trayectoria trabaja con los puntos del espacio cartesiado, y con los tiempos de ejecución del movimiento; como datos iniciales de la orden de movimiento. El generador de trayectorias lineal trabaja con las ordenes de movimiento de entrada en el espacio cartesiano, y el numero de puntos intermedios en una trayectoria rectilinea. Con estos datos de entrada y el calculo matemático de la trayectoria se puede entregar un muestreo del número de puntos indicados, equidistantes en la trayectoria deseada.

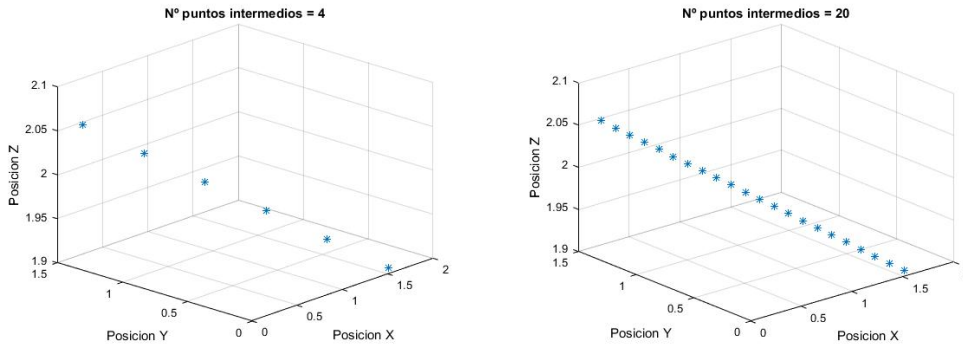


Figura 2: Generador Trayectorias lineal

■ GENERADOR DE TRAYECTORIAS CIRCULAR

El generador de trayectorias circulares es similar al generador de trayectorias lineal, excepto por el cálculo de la trayectoria en sí, es decir, una vez obtenidos los puntos que definen la trayectoria, se realizan los mismos cálculos para la obtención de las posiciones, velocidades y aceleraciones articulares. Se empleará también el método heurístico para definir las velocidades de paso.

Para empezar, se deben introducir 3 puntos, a diferencia de los generadores lineales, que definirán la curva como punto inicial (P_1), punto intermedio (P_2) y punto final (P_3). Tras ello, se calculara el centro de la circunferencia.

Primero se calcula el vector unitario perpendicular al plano definido por la circunferencia:

$$v_1 = (P_2 - P_1) \wedge (P_3 - P_1)$$

$$v_1 = \frac{v_1}{||v_1||}$$

Una vez obtenido dicho vector, se pasará al cálculo del centro de la circunferencia(P_0), lo cual se realiza con un sistema de 3 ecuaciones que se resuelve de forma matricial e imponiendo que se cumplan las siguientes relaciones entre vectores (puede haber más de un sistema de ecuaciones que lleve a la resolución del problema):

- El producto escalar entre el vector v_1 y el vector definido por $P_0 - P_1$ ha de ser nulo.
- El producto escalar entre el vector $P_0 - \frac{P_2+P_1}{2}$ y el vector $P_2 - P_1$ ha de ser nulo.
- El producto escalar entre el vector $P_0 - \frac{P_3+P_1}{2}$ y el vector $P_3 - P_1$ ha de ser nulo.

Una vez definidas las ecuaciones se ha de obtener el sistema de ecuaciones que verifique que:

$$P_0 = A|B \quad (8)$$

Realizando las operaciones pertinentes y despejando se pueden obtener las matrices A y B como:

$$A = \begin{bmatrix} v_1(1) & v_1(2) & v_1(3) \\ P_2(1) - P_1(1) & P_2(2) - P_1(2) & P_2(3) - P_1(3) \\ P_3(1) - P_1(1) & P_3(2) - P_1(2) & P_3(3) - P_1(3) \end{bmatrix} \quad (9)$$

$$B = \begin{bmatrix} P_1(1)v_1(1) + P_1(2)v_1(2) + P_1(3)v_1(3) \\ ((P_2(1) + P_1(1))\frac{P_2(1)-P_1(1)}{2} + ((P_2(2) + P_1(2))\frac{P_2(2)-P_1(2)}{2} + ((P_2(3) + P_1(3))\frac{P_2(3)-P_1(3)}{2} \\ ((P_3(1) + P_1(1))\frac{P_3(1)-P_1(1)}{2} + ((P_3(2) + P_1(2))\frac{P_3(2)-P_1(2)}{2} + ((P_3(3) + P_1(3))\frac{P_3(3)-P_1(3)}{2} \end{bmatrix} \quad (10)$$

Una vez hallado el centro de la circunferencia, P_0 , se debe proceder a calcular los puntos solicitados al principio del algoritmo.

Para ello se necesitará definir los siguientes vectores:

$$v_2 = \frac{P_1 - P_0}{\|(P_1 - P_0)\|} \quad v_3 = -\frac{v_2 \wedge v_1}{\|(v_2 \wedge v_1)\|} \quad (11)$$

que serán, respectivamente, el vector unitario que va desde el centro de la circunferencia al punto P_1 y el vector unitario perpendicular al plano definido por v_1 y v_2 con origen en P_1 y sentido de la trayectoria. Con estos dos vectores, el pseudocódigo implementado para calcular los puntos que componen la trayectoria se muestra a continuación:

$$\begin{aligned} g &= \frac{P_3 - P_0}{\|(P_3 - P_0)\|} \\ \cos_g &= (v_2 \cdot g) \\ \sin_g &= \|v_2 \wedge g\| \\ \rho_{fin} &= 2\pi - \text{atan2}(\sin_g, \cos_g) \\ \rho &= \text{linspace}(0, \rho_{fin}, \text{num_ptos}) \\ P &= \text{repmat}(P_0, 1, \text{prod}(\text{size}(\rho))) + R(v_2 \cos(\rho) + v_3 \sin(\rho)) \end{aligned}$$

Siendo ρ_{fin} el ángulo de la posición final de la trayectoria con respecto a la posición inicial, ρ el vector de N puntos de valores de ángulos desde 0 hasta el ángulo de la posición final, R el radio del arco de circunferencia calculado como $R = \|P_1 - P_0\|$, y P la matriz de posiciones correspondientes a los puntos equidistantes a lo largo de la trayectoria.

Una vez realizados estos pasos ya se puede pasar al cálculo de los valores de las variables articulares para las posiciones calculadas como se ha hecho anteriormente en el generador de trayectorias lineal.

Los resultados obtenidos para una trayectoria específica y un valor de 100 puntos a calcular se mostrarán a continuación.

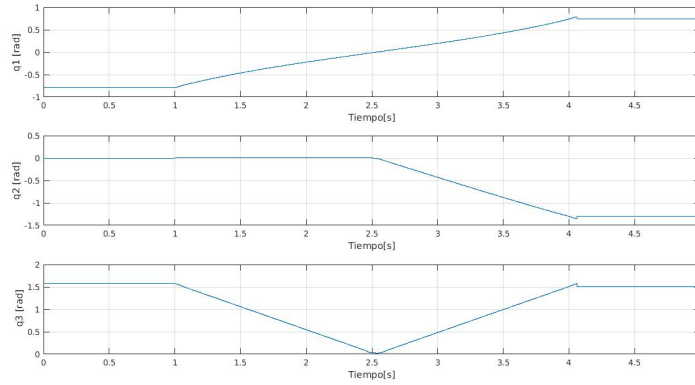


Figura 3: Valores de las variables articulares

Tras ello, la trayectoria ejemplo que podría seguirse en el plano XYZ será:

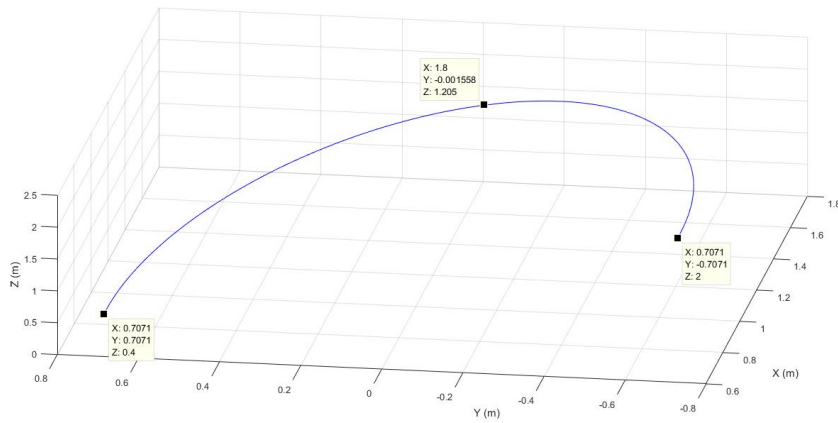


Figura 4: Posición del efector final

3.2. Interpoladores

Una vez obtenidos estos puntos de referencia, echaremos mano de las herramientas cedidas del análisis cinemático, ya que, se hace uso del *Modelo Cinemático Inverso*, para la obtención de los puntos de la trayectoria, representadas en el espacio de trabajo articular del robot.

Una vez llegadas aquí, ya tenemos un numero de referencias articulares , que puede conllevar a un control punto a punto; en nuestro caso, deberemos realizar un linealización de los puntos ya que debemos ser capaces de generar referencias a nuestro sistema de forma continua. A continuación se tratara el interpolador utilizado.

- **SPLINE CÚBICO.**

Este interpolador implementa un polinomio de 3º para dar continuidad de la trayectoria de puntos que se les entregue del generador pertinente, de la forma:

$$q(t) = a + b(t - t_i) + c(t - t_i)^2 + d(t - t_i)^3 \quad ; \quad t_i < t < t_{i+1} \quad (12)$$

- $a = q_i$
- $b = \dot{q}_i$
- $c = \frac{3}{T^2}(q_{i+1} - q_i) - \frac{1}{T}(\dot{q}_{i+1} + 2\dot{q}_i)$
- $d = -\frac{2}{T^3}(q_{i+1} - q_i) - \frac{1}{T^2}(\dot{q}_{i+1} + \dot{q}_i)$
- $T = t_{i+1} - t_1$

El trabajo matricial de esta ecuación en el espacio articular permite obtener tantos polinomios como tramos en la trayectoria sean exigidas por el número de puntos intermedios; pero, para ello debemos asignar valores a los parámetros a , b , c y d ; para estos que contengan parámetros de velocidad, se utilizará el método heurístico para la asignación de velocidades articulares.

Le método Heurístico se basta con el análisis de la variación de las posiciones articulares de las muestras para asignar una velocidad de la articulación igual a cero, en el caso de que se varíe la dirección del movimiento; y una media entre ambas velocidades, si se aportan a la misma dirección de giro. Véase:

$$\dot{q}_i = \begin{cases} 0 & \text{si } \text{signo}(q_i - q_{i-1}) \neq \text{signo}(q_{i+1} - q_i) \\ \frac{(q_{i+1} - q_i) + (q_i - q_{i-1})}{2T} & \text{si } \text{signo}(q_i - q_{i-1}) = \text{signo}(q_{i+1} - q_i) \end{cases}$$

3.3. Pruebas y conclusiones

- **Objetivos:**
Ser capaces de que nuestro modelo de brazo manipulador ejecute ordenes de movimientos basados en trayectorias.
- **Necesidad para el trabajo:**
Generar trayectorias de referencia en el espacio articular de nuestro modelo brazo manipulador.
- ¿Como lo hacemos?
Herramientas matemáticas. que traduzcan las funcionalidades de la Orden de movimiento.
Análisis Cinemático previo con el que poder trasladar las ordenes del espacio cartesiano, al articular del robot. Generar Señal de referencia: Aqui asignamos las características de la señal de referencia; asegurar condiciones especificas de velocidad y aceleraciones, además de la continuidad en las consignas de referencia, apoyados en algoritmos de interpolación.

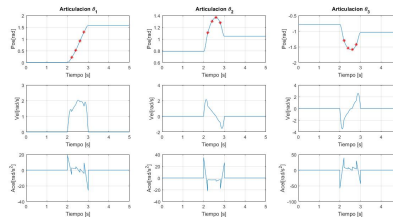


Figura 5: Salida del Control Cinemático con Generador Lineal, e Interpolador cúbico

Una vez conseguidas estas referencias, y al no poder sen entregadas, como intensidades que varien de forma continua a los motores de nuestro modelo de robot, es necesaria la adición de Mantenedores de Orden Cero, *ZoH*, para ajustar la señal a escalones de referencias, que puedan ser referencias reales a nuestro modelo de robot en intensidades.

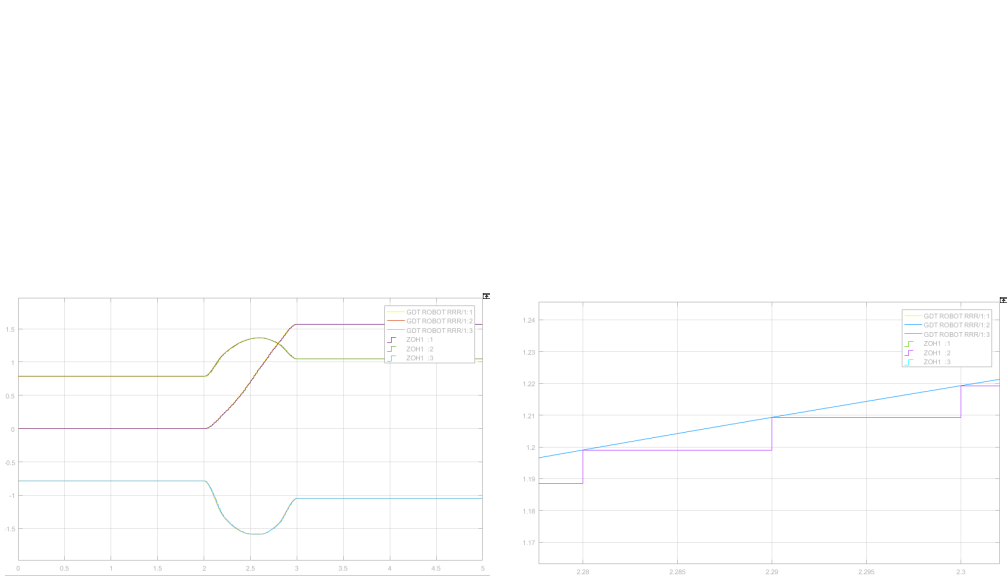


Figura 6: Referencia Generada Vs Referencia Entregada al Modelo

4. Análisis Dinámico del brazo

Una vez resuelto el problema cinemático del brazo, se debe pasar a realizar un análisis dinámico del mismo. Comenzando por un análisis que nos muestre la relación que existe entre las intensidades aplicadas a los motores de las articulaciones del robot y las posiciones, velocidades y aceleraciones de dichas articulaciones.

Obteniendo así, un modelo dinámico del brazo manipulador; que nos permitirá desarrollar posteriormente todas las técnicas de control propuestas, por tanto, la fiabilidad, y la exactitud del modelo obtenido, determinaran en gran medida al control que se puedan desarrollar en este proyecto.

4.1. Modelado Dinámico e Incertidumbres

Como se ha comenzado diciendo, se ha de ser precavido en el desarrollo de este modelo dinámico, y mas aún, de cuanto podremos fiarnos de este.

En primer lugar, nos basamos en el modelo estructural que permite obtener dicha relación dinámica del motor de corriente continua:

$$K_t R I_m = (M(q) + J_m R^2) \ddot{q} + (C(q, \dot{q}) + B_m R^2) \dot{q} + G(q) + F(\dot{q}) \quad (13)$$

En la ecuación anterior, se tiene; en el término de la izquierda, las matrices de constantes de par de cada motor (K_t) y de reductoras (R) e intensidades (I_m) aplicadas a cada motor; por otro lado, en el término de la derecha encontramos las matrices de inercia de los eslabones ($M(q)$) y de los motores (J_m), la matriz de términos de Coriolis ($C(q, \dot{q})$) y la de términos viscosos de los motores (B_m), y las matrices de términos gravitatorios ($G(q)$) y las fuerzas externas que puedan ser aplicadas sobre el robot, ($F(\dot{q})$), donde ésta última no se tendrá en cuenta para la estimación del modelo.

Aparecen también los vectores columna q , \dot{q} y \ddot{q} que corresponden, respectivamente, a los valores de posición, velocidad y aceleración de las articulaciones.

Teniendo la ecuación matricial, se prosigue definiendo el contenido interno de estas matrices, donde entramos a considerar las incertidumbres dinámicas del robot. Estas son, en base a la estructura tomada:

- Momentos de Inercia orden 0:

$$m_0 \quad m_1 \quad m_2 \quad m_3$$

- Momentos de Inercia orden 1:

$$s_{11x} \quad s_{11y} \quad s_{11z} \quad s_{22x} \quad s_{22y} \quad s_{22z} \quad s_{33x} \quad s_{33y} \quad s_{33z}$$

- Momentos de Inercia orden 2:

$$I_{11xx} \quad I_{11yy} \quad I_{11zz} \quad I_{22xx} \quad I_{22yy} \quad I_{22zz} \quad I_{33xx} \quad I_{33yy} \quad I_{33zz}$$

- Inercia y Fricciones de Motores:

$$Jm_1 \quad Jm_2 \quad Jm_3 \quad Bm_1 \quad Bm_2 \quad Bm_3$$

Tras definir las incógnitas inerciales, y el conocimiento geométrico del robot, se procede a la descripción interna de las matrices que definen nuestro modelo. Para ello, se va a realizar un método recursivo que se basa en la segunda ley de Newton denominado algoritmo de Newton-Euler, el cual obtiene los esfuerzos/pares aplicados en cada articulación.

Dicho algoritmo ha sido proporcionado en clase por lo que no se explicará aquí; sí decir, que del robot sólo se conocen las longitudes de los eslabones y algunos parámetros dinámicos (reductoras y constantes de par), por tanto, se realizará el cálculo con las incertidumbres dinámicas definidas como variables simbólicas que serán estimadas y sustituidas más adelante.

El resultado que se desea obtener con dicho algoritmo es el siguiente:

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + G(q), \quad (14)$$

donde el término $V(q, \dot{q})$ hace referencia al producto $C(q, \dot{q})\dot{q}$.

Como el algoritmo anterior obtiene el resultado total, se debe derivar el mismo para poder hallar las matrices aisladas. Por ello, el resultado τ se deriva respecto a \ddot{q} para obtener la matriz $M(q)$. Acto seguido restar la matriz obtenida multiplicada por \ddot{q} al valor de τ para eliminarlo y obtener las otras dos matrices.

$$M(q) = \frac{\partial \tau}{\partial \ddot{q}}$$

El siguiente paso consiste en derivar la nueva τ respecto a la constante de gravedad g , puesto que aparece únicamente en los términos gravitatorios. Así, y multiplicando por g a posteriori, se obtiene la matriz $G(q)$. Para hallar la matriz $V(q, \dot{q})$ basta con restar a la τ resultante de extraer la matriz de inercia la $G(q)$ anterior.

$$\tau_{nueva} = \tau - \frac{\partial \tau}{\partial \ddot{q}}\ddot{q} \rightarrow G(q) = \frac{\partial \tau_{nueva}}{\partial g} \rightarrow V(q, \dot{q}) = \tau_{nueva} - G(q)$$

Una de las características que hay que tener en cuenta del algoritmo de Newton-Euler es que no tiene en cuenta las viscosidades e inercias de los motores, por lo que hay que añadirlas a posteriori obteniendo las matrices:

$$Ma(q) = M(q) + J_m R^2 \quad (15)$$

$$Va(q, \dot{q}) = V(q, \dot{q}) + B_m R^2 \dot{q} \quad (16)$$

$$Ga(q) = G(q) \quad (17)$$

Al fin de éste desarrollo matemático, se habrá obtenido el modelo dinámico simbólico del robot. Es un proceso sistemático, ya que únicamente consiste en seguir unos pasos descritos en clase, pero que, si se deseara, se podría comparar con un robot diseñado en Robotics Toolbox diseñado por *Peter Corke* para Matlab asignando valores a los parámetros y realizando un mismo experimento para ambos modelos.

Las ecuaciones dinámicas simbólicas que definen las articulaciones del robot se muestran a continuación:

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} Ma_{1,1} & 0 & 0 \\ 0 & Ma_{2,2} & Ma_{2,3} \\ 0 & Ma_{3,2} & Ma_{3,3} \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix} + \begin{pmatrix} Va_{1,1} \\ Va_{2,1} \\ Va_{3,1} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} + \begin{pmatrix} 0 \\ Ga_2 \\ Ga_3 \end{pmatrix}$$

A continuación se definirán las matrices que conforman las ecuaciones dinámicas del robot por separado:

La matriz asociada a los terminos inerciales del robot será:

$$Ma = \begin{bmatrix} Ma_{1,1} & 0 & 0 \\ 0 & Ma_{2,2} & Ma_{2,3} \\ 0 & Ma_{3,2} & Ma_{3,3} \end{bmatrix}$$

- $Ma_{1,1} = I_{11yy} + 0,5I_{22xx} + 0,5I_{22yy} + 0,5I_{33xx} + 0,5I_{33yy} - 0,5I_{22xx}\cos(2q_2) + 0,5I_{22yy}\cos(2q_2) + Jm_1R_1^2 + 0,5L_2^2m_2 + 0,5L_2^2m_3 + 0,5L_3^2m_3 - 0,5I_{33xx}\cos(2q_2 + 2q_3) + 0,5I_{33yy}\cos(2q_2 + 2q_3) + m_1s_{11}z^2 + 0,5m_2s_{22x}^2 + 0,5m_3s_{33x}^2 + 0,5m_2s_{22x}^2\cos(2q_2) + 0,5L_3^2m_3\cos(2q_2 + 2q_3) + 0,5m_3s_{33x}^2\cos(2q_2 + 2q_3) - L_2m_2s_{22x} - L_3m_3s_{33x} + 0,5L_2^2m_2\cos(2q_2) + 0,5L_2^2m_3\cos(2q_2) + L_2L_3m_3\cos(q_3) - L_2m_3s_{33x}\cos(q_3) - L_3m_3s_{33x}\cos(2q_2 + 2q_3) + L_2L_3m_3\cos(2q_2 + q_3) - L_2m_3s_{33x}\cos(2q_2 + q_3) - L_2m_2s_{22x}\cos(2q_2)$
- $Ma_{2,2} = I_{22zz} + I_{33zz} + Jm_2R_2^2 + L_2^2m_2 + L_2^2m_3 + L_3^2m_3 + m_2s_{22x}^2 + m_3s_{33x}^2 - 2L_2m_2s_{22x} - 2L_3m_3s_{33x} + 2L_2L_3m_3\cos(q_3) - 2L_2m_3s_{33x}\cos(q_3)$
- $Ma_{2,3} = m_3L_3^2 - 2L_3m_3s_{33x} + L_2L_3m_3\cos(q_3) + m_3s_{33x}^2 - L_2m_3s_{33x}\cos(q_3) + I_{33zz}$
- $Ma_{3,2} = I_{33zz} + m_3(L_3 - s_{33x})(L_3 - s_{33x} + L_2\cos(q_3))$
- $Ma_{3,3} = I_{33zz} + Jm_3R_3^2 + m_3(L_3 - s_{33x})^2$

La matriz asociada a los terminos de Coirolis del robot será:

$$Va = \begin{pmatrix} Va_1 \\ Va_2 \\ Va_3 \end{pmatrix}$$

- $Va_1 = Bm_1\dot{q}_1R_1^2 - \dot{q}_1(I_{33yy}\dot{q}_2\sin(2q_2 + 2q_3)) - I_{33xx}\dot{q}_3\sin(2q_2 + 2q_3) + I_{33yy}\dot{q}_3\sin(2q_2 + 2q_3) - I_{22xx}\dot{q}_2\sin(2q_2) + I_{22yy}\dot{q}_2\sin(2q_2) + m_3\dot{q}_2s_{33x}^2\sin(2q_2 + 2q_3) + m_3\dot{q}_3s_{33x}^2\sin(2q_2 + 2q_3) + L_2^2m_2\dot{q}_2\sin(2q_2) + L_2^2m_3\dot{q}_2\sin(2q_2) + m_2\dot{q}_2s_{22x}^2\sin(2q_2) + L_3^2m_3\dot{q}_2\sin(2q_2 + 2q_3) + L_3^2m_3\dot{q}_3\sin(2q_2 + 2q_3) + 2L_3L_2m_3\dot{q}_2\sin(2q_2 + q_3) + L_3L_2m_3\dot{q}_3\sin(2q_2 + q_3) - 2L_2m_3\dot{q}_2s_{33x}\sin(2q_2 + q_3) - L_2m_3\dot{q}_3s_{33x}\sin(2q_2 + q_3) - 2L_2m_2\dot{q}_2s_{22x}\sin(2q_2) + L_3L_2m_3\dot{q}_3\sin(q_3) - L_2m_3\dot{q}_3s_{33x}\sin(q_3) - 2L_3m_3\dot{q}_2s_{33x}^2\sin(2q_2 + 2q_3) - 2L_3m_3\dot{q}_3s_{33x}^2\sin(2q_2 + 2q_3)$
- $Va_2 = 0,5I_{33yy}\dot{q}_1^2\sin(2q_2 + 2q_3) - 0,5I_{33xx}\dot{q}_1^2\sin(2q_2 + 2q_3) + Bm_2R_2^2\dot{q}_2 - 0,5I_{22xx}\dot{q}_1^2\sin(2q_2) + 0,5I_{22yy}\dot{q}_1^2\sin(2q_2) + 0,5L_3^2m_3\dot{q}_1^2\sin(2q_2 + 2q_3) + 0,5m_3\dot{q}_1^2s_{33x}^2\sin(2q_2 + 2q_3) + 0,5L_2^2m_2\dot{q}_1^2\sin(2q_2) + 0,5L_2^2m_3\dot{q}_1^2\sin(2q_2) + 0,5m_2\dot{q}_1^2s_{22x}^2\sin(2q_2) - L_3L_2m_3\dot{q}_3^2\sin(q_3) + L_2m_3\dot{q}_3^2s_{33x}\sin(q_3) - L_3m_3\dot{q}_1^2s_{33x}\sin(2q_2 + 2q_3) + L_3L_2m_3\dot{q}_1^2\sin(2q_2 + q_3) - L_2m_3\dot{q}_1^2s_{33x}\sin(2q_2 + q_3) - L_2m_2\dot{q}_1^2s_{22x}\sin(2q_2) - 2L_3L_2m_3\dot{q}_3\dot{q}_2\sin(q_3) + 2L_2m_3\dot{q}_2\dot{q}_3s_{33x}\sin(q_3)$

$$\begin{aligned}
\blacksquare \quad V a_3 = & 0,5I_{33yy}\dot{q}_1^2 \sin(2q_2 + 2q_3) - 0,5I_{33xx}\dot{q}_1^2 \sin(2q_2 + 2q_3) + Bm_3R_3^2\dot{q}_3 \\
& + 0,5L_3^2m_3\dot{q}_1^2 \sin(2q_2 + 2q_3) + 0,5m_3\dot{q}_1^2 s_{33x}^2 \sin(2q_2 + 2q_3) + 0,5L_3L_2m_3\dot{q}_1^2 \sin(q_3) \\
& + L_3L_2m_3\dot{q}_2^2 \sin(q_3) - 0,5m_3L_2\dot{q}_1^2 s_{33x} \sin(q_3) - m_3L_2\dot{q}_2^2 s_{33x} \sin(q_3) \\
& - L_3m_3\dot{q}_1^2 s_{33x} \sin(2q_2 + 2q_3) + 0,5L_2L_3m_3\dot{q}_1^2 \sin(2q_2 + q_3) - 0,5L_2m_3\dot{q}_1^2 s_{33x} \sin(2q_2 + q_3)
\end{aligned}$$

La matriz asociada a los terminos gravitatorios del robot será:

$$Ga = g \begin{pmatrix} 0 \\ (L_3m_3\cos(q_2 + q_3) - m_2s_{22x}\cos(q_2) - m_3s_{33x}\cos(q_2 + q_3) + L_2m_2\cos(q_2)) \\ m_3\cos(q_2 + q_3)(L_3 - s_{33x}) \end{pmatrix}$$

4.2. Obtención de los parámetros dinámicos del robot

Al obtener las matrices del modelo dinámico, ya seremos capaces de modelar el comportamiento del robot de trabajo; una vez se determinen las incertidumbres paramétricas que se especificaron al inicio del apartado anterior. Osea se, que aunque tengamos estructuradas las matrices, no seremos capaces de determinar el comportamiento del robot, ya que, tenemos un compendio de incertidumbres, que deberán ser determinadas numéricamente.

Por tanto, este apartado esclarecerá como resolver las incertidumbres paramétricas presentes, mediante técnicas de regresión lineal y la realización de experimentos sobre el propio robot, para determinar las siguientes incógnitas definidas en Matlab como variables simbólicas:

4.2.1. Identificación y Linealización de parámetros dinámicos

Primero reestructuraremos nuestro modelo dinámico en dos matrices en las que, por un lado, se agrupen las incertidumbres paramétricas en una matriz denominada *theta*, y por otro, el resto de nuestro modelo dinámico en una matriz denominada *phi* que dependerá de las variables articulares del robot.

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + G(q) \quad \rightarrow \quad \tau = \phi(x) \quad \theta \quad (18)$$

Agrupando en *theta* el conjunto de las incertidumbres obtenemos el vector de términos resultante.

$$\theta = \begin{pmatrix} I_{11xx} \\ I_{11yy} \\ I_{11zz} \\ Jm_1 \\ Bm_1 \\ I_{22xx} \\ I_{22yy} \\ I_{22zz} \\ Jm_2 \\ Bm_2 \\ I_{33xx} \\ I_{33yy} \\ I_{33zz} \\ Jm_3 \\ Bm_3 \\ m_1s_{11z}^2 \\ m_1s_{11z} \\ m_1 \\ m_2s_{22x}^2 \\ m_2s_{22x} \\ m_2 \\ m_3s_{33x}^2 \\ m_3s_{33x} \\ m_3 \end{pmatrix} \quad (19)$$

Si usamos este vector θ como base para re-ordenar nuestro modelo dinámico, vemos, en la matriz resultante ϕ , que existen columnas repletas de ceros, o relaciones lineales entre las columnas, por lo que, se deberán agrupar aquellos términos que no sean linealmente independientes y eliminar los términos asociados a las columnas de ceros, ya que significará que no tienen relevancia en nuestro modelo dinámico.

Con el fin de agrupar los parámetros linealmente independientes, se empleará Matlab para poder gestionar estas matrices de gran tamaño.

Se irán asignando valores aleatorios a los parámetros articulares que aparecerán en la matriz ϕ y se sustituirán estos valores dentro de la matriz; se irán asignando valores aleatorios a las variables articulares, y se concatenarán con la matriz anterior. Este proceso se repetirá hasta obtener una matriz concatenada de iguales dimensiones de ancho y largo, es decir, hasta que la matriz sea cuadrada.

Obteniendo esta matriz cuadrada podremos ejecutar el comando $rref()$ que nos devolverá las relaciones entre las columnas de la matriz ϕ , las cuales serán las mismas relaciones dentro de las filas del vector de incertidumbres θ .

Por tanto, la matriz θ linealmente independiente obtenida será:

$$\theta = \begin{pmatrix} m_1 s_{11z}^2 + m_2 s_{22x}^2 + m_3 s_{33x}^2 + I_{11yy} + I_{22yy} + I_{33yy} + R_1^2 J m_1 - m_2 - 1,64 m_3 \\ B m_1 \\ -m_2 s_{22x}^2 + I_{22xx} - I_{22yy} + m_2 + m_3 \\ m_2 s_{22x}^2 + I_{22zz} + R_2^2 J m_2 - m_2 - m_3 \\ B m_2 \\ -m_3 s_{33x}^2 + I_{33xx} - I_{33yy} + 0,64 m_3 \\ m_3 s_{33x}^2 + I_{33zz} - 0,64 m_3 \\ J m_3 \\ B m_3 \\ -m_2 - m_3 + m_2 s_{22x} \\ m_3 s_{33x} - 0,8 m_3 \end{pmatrix}$$

Por otro lado, la matriz γ obtenida, la cuál únicamente también posee términos linealmente independientes, se muestra a continuación:

$$\gamma = \begin{pmatrix} \ddot{q}_1 & R_1^2 \dot{q}_1 & \gamma_{1,3} & 0 & 0 & \gamma_{1,6} & 0 & 0 & 0 & \gamma_{1,10} & \gamma_{1,11} \\ 0 & 0 & \gamma_{2,3} & \ddot{q}_2 & R_2^2 \dot{q}_2 & \gamma_{2,6} & \ddot{q}_2 + \ddot{q}_3 & 0 & 0 & \gamma_{2,10} & \gamma_{2,11} \\ 0 & 0 & 0 & 0 & 0 & \gamma_{3,6} & \ddot{q}_2 + \ddot{q}_3 & R_3^2 \ddot{q}_3 & R_3^2 \ddot{q}_3 & 0 & \gamma_{3,11} \end{pmatrix}$$

$$\blacksquare \gamma_{1,3} = \frac{\ddot{q}_1}{2} - \frac{\ddot{q}_1 \cos(2q_2)}{2} + \dot{q}_1 \dot{q}_2 \sin(2q_2)$$

$$\blacksquare \gamma_{1,6} = \frac{\ddot{q}_1}{2} - \frac{\ddot{q}_1 \cos(2q_2 + 2q_3)}{2} + \dot{q}_1 \dot{q}_2 \sin(2q_2 + 2q_3) + \dot{q}_1 \dot{q}_3 \sin(2q_2 + 2q_3)$$

$$\blacksquare \gamma_{1,10} = -L_2(\ddot{q}_1 + \ddot{q}_1 \cos(2q_2) - 2\dot{q}_1 \dot{q}_2 \sin(2q_2))$$

$$\blacksquare \gamma_{1,11} = 2L_3 \dot{q}_1 \dot{q}_2 \sin(2q_2 + 2q_3) - L_2 \ddot{q}_1 \cos(2q_2 + q_3) - L_3 \ddot{q}_1 \cos(2q_2 + 2q_3) - L_2 \ddot{q}_1 \cos(q_3) - L_3 \ddot{q}_1 + 2L_3 \dot{q}_1 \dot{q}_3 \sin(2q_2 + 2q_3) + L_2 \dot{q}_1 \dot{q}_3 \sin(q_3) + 2L_2 \dot{q}_1 \dot{q}_2 \sin(2q_2 + q_3) + L_2 \dot{q}_1 \dot{q}_3 \sin(2q_2 + q_3)$$

$$\blacksquare \gamma_{2,3} = -\frac{\dot{q}_1^2 \sin(2q_2)}{2}$$

- $\gamma_{2,6} = -\frac{\dot{q}_1^2 \sin(2q_2+2q_3)}{2}$
- $\gamma_{2,10} = -L_2 \sin(2q_2) \dot{q}_1^2 - 2L_2 \ddot{q}_2 - g * \cos(q_2)$
- $\gamma_{2,11} = L_2 q_3^2 \sin(q_3) - 2L_3 \ddot{q}_3 - g * \cos(q_2 + q_3) - 2L_3 \ddot{q}_2 - L_2 \dot{q}_1^2 \sin(2q_2 + q_3) -$
 $-L_3 \dot{q}_1^2 \sin(2q_2 + 2q_3) - 2L_2 \ddot{q}_2 \cos(q_3) - L_2 \ddot{q}_3 \cos(q_3) + 2L_2 \dot{q}_2 \dot{q}_3 \sin(q_3)$
- $\gamma_{3,6} = -\frac{\dot{q}_1^2 \sin(2q_2+2q_3)}{2}$
- $\gamma_{3,11} = -2L_3 \ddot{q}_2 - 2L_3 \ddot{q}_3 - g * \cos(q_2 + q_3) - \frac{L_2 \dot{q}_1^2 \sin(q_3)}{2} - L_2 \dot{q}_2^2 \sin(q_3) -$
 $-\frac{L_2 \dot{q}_1^2 \sin(2q_2+q_3)}{2} - L_3 \dot{q}_1^2 \sin(2q_2 + 2q_3) - L_2 \ddot{q}_2 \cos(q_3)$

Para verificar que no se ha cometido errores en este procedimiento basta con multiplicar las dos matrices obtenidas, y restarlo al modelo dinámico que se obtuvo con el algoritmo de Newton-Euler, ya que, al ser una reorientación de parámetros, este resultado debe ser 0; por lo que si este resultado es diferente se debe comprobar el proceso en busca de errores.

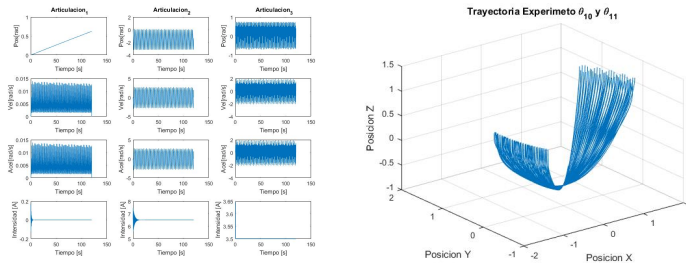
4.2.2. Estimación estadística de los parámetros linealmente independientes

Una vez que agrupados correctamente los términos linealmente independientes de nuestra estructuradas *phi-theta*, se procederá a la realización de los experimentos sobre el robot en cuestión, y la determinación de estas incertidumbres dinámicas.

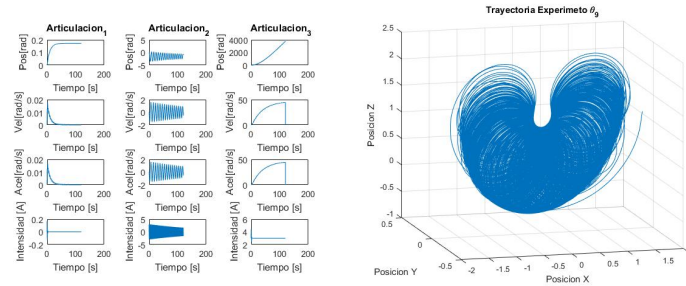
Como se explicó en clases, esta identificación se basa en la regresión lineal con la minimización de los errores cuadráticos, para ello, necesitaremos realizar un numero de experimentos mucho mas elevado que el número de incertidumbres que tenemos en la matriz *theta* linealmente independiente. Además, se deberá hacer un mínimo diseño de los experimentos a los que se va a someter el robot para la búsqueda de estos parámetros.

A continuación, se tratará obre los experimentos y la metodología seguida. Se tienen unos parámetros que se encuentran fuertemente asociados a determinados valores de las variables articulares; se puede ver esta dependencia al analizar el modelo dinámico de Newton-Euler. Ante esto, podemos deducir a priori, que ciertos parámetros estarán mas representados ante ciertas condiciones de funcionamiento; esto será lo que determinara la validez del experimento, y de donde sacaremos las consignas para el diseño de estos.

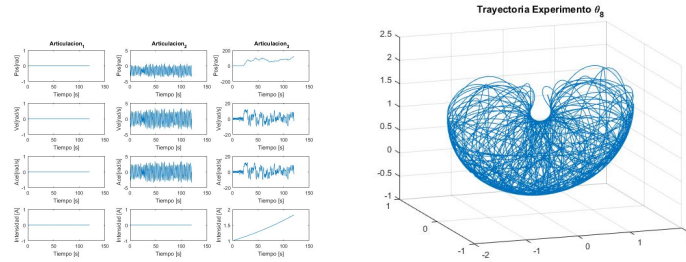
- Términos Gravitatorios → representados por velocidades bajas y constantes



- Términos Viscosos → valores de velocidad altas, y aceleraciones constantes



- Términos Inerciales → consignas de aceleraciones altas; movimientos breves y bruscos en las articulaciones



Para implementar las condiciones de funcionamiento han sido necesarias dos metodologías diferentes:

Por un lado, la propuesta en clase, que basa el diseño de las intensidades de entrada al robot, mediante la suma de dos señales senoidales, pudiendo variar de estas su amplitud, frecuencia y atenuación. Por otro, se han empleado señales exponenciales, en la que se han podido variar tanto la pendiente como la curvatura de la señal. El uso de estas dos maneras de diseño de las señales de intensidad, es justificado debido a las problemáticas de los modelos a identificar.

Se han identificado los modelos reales e ideales, con y sin reductoras, del robot propuesto. En el caso del robot real, solo se han tomado los datos procedentes de las posiciones, a las que se le ha introducido, el efecto del encoder, para asemejar estos resultados a los obtenidos en la realidad. Tras ello, se aplicaron sucesivas derivadas, para la obtención de las variables de velocidad y aceleración articulares.

Ha sido necesario introducir filtros Butterworth, un filtro paso bajo, que eliminen las componentes de alta frecuencia del ruido asociadas al efecto del encoder; es por ello, que en ciertos experimentos, donde la variación de las posiciones era fundamental para obtener las condiciones de funcionamiento mencionadas anteriormente, eran filtradas como ruido.

Es por esto, que se buscaron nuevas posibilidades, y se optó por la implementación de señales exponenciales, para someter al robot real a condiciones de altas aceleraciones sin necesidad de realizar una variación senoidal cuya frecuencia esté fuera de la banda de corte del filtrado.

Aun así, las dificultades a la hora de la identificación de los parámetros dinámicos, se han dado principalmente, por el robot real sin reductoras.

En este, se ha observado, que no solo se tiene que aumentar las intensidades de entrada de los experimentos, ya que, desde el punto de vista de la articulación, la ausencia de las reductoras se traduce en un par del motor menor; sino que, además de ello, los efectos externos que ocurren en las otras articulaciones o eslabones, cobran mucha más relevancia, al no tener la amortiguación de las reductoras.

Es decir, al no emplear reductoras el robot no se encontrará desacoplado y, una acción en una articulación, sea del tipo que sea, afectará a la siguiente.

En conclusión, la ausencia de las reductoras dificultan en gran medida la búsqueda exclusiva del parámetro dinámico a identificar numéricamente.

Finalmente, una vez definidas las consignas para la identificación de los parámetros, las formas de ondas de las señales aplicadas al robot para ello, y los modelos sobre los que se han realizado dichos experimentos, se abordará el cálculo estadístico detrás de la validación de estos experimentos, dando como resultado la calidad de estos.

Partiendo de la metodología expuesta en clase sobre el cálculo de la covarianza, donde se asume que el error cometido en la medida de las señales es análogo a un ruido blanco, y que el algoritmo para calcular la desviación estándar relativa del parámetro, el cual se encuentra en las diapositivas de la asignatura, se ha implementado correctamente en el código de programación de Matlab, se pasa a exponer las condiciones que aseguren la veracidad de este.

Por un lado, y como se explicó al inicio de este punto, el principio seguido para la identificación ha sido por caso redundante, es decir, la toma de datos del orden de mil veces superior, al número de parámetros dinámicos a identificar.

Por otro lado, cabe destacar que no se puede tomar las medidas consecutivas en el tiempo, sino que se debe realizar una espera entre la toma de los datos de salida del robot, evitando así los errores del sensorizado producidos por conservar propiedades de la medida una vez los estímulos iniciales han desaparecido. Esto viene a decir, que para evitar los errores por histéresis de los sensores, se han tomado los datos cada 15-20 ud de tiempo de simulación.

Finalmente, también se ha impuesto un umbral entorno al 5 % del error sobre el valor real. En el momento en el que la desviación típica entre las medidas se encuentre por debajo de este, se asumirá que el parámetro se ha obtenido correctamente.

A modo de consideraciones finales, cabe destacar que a la hora de la realización del experimento, se debe tener muy en cuenta que parámetros se están buscando, ya que, el algoritmo del cálculo de la covarianza, puede dar falsos positivos, en parámetros que ni siquiera se están acentuando.

Es por ello, que es recomendable iniciar la identificación, por los modelos ideales, y posteriormente, con los datos del modelo real, teniendo al menos una base de apoyo a la hora de determinar si un parámetro ha sido hallado correctamente o ha sido un falso válido.

4.3. Parámetros y modelos obtenidos

A continuación, se mostrarán los parámetros obtenidos para cada configuración del robot, así como la covarianza con la que se han obtenido los mismos.

Para evitar repetir siempre los parámetros, se irán definiendo componente a componente, es decir, a continuación se definirá θ con todos los parámetros y se dirá en cada caso concreto la posición del parámetro obtenido en el vector, el valor de dicho parámetro y la covarianza del mismo.

Además de ello, se obtendrán las ecuaciones dinámicas que definen el comportamiento dinámico de las articulaciones del robot. Para obtener estas ecuaciones, será necesario multiplicar la matriz γ que posteriormente se definirá por la matriz θ con valores numéricos.

Una vez se tengan las ecuaciones dinámicas del robot, al igual que se hizo cuando se aplicó el algoritmo de *Newton-Euler*, se irán derivando las expresiones respecto las posiciones, velocidades y aceleraciones articulares para obtener las matrices de inercias, matriz de términos de Coriolis y la matriz formada por los términos gravitatorios.

Por último, cabe destacar que los modelos con medidas reales se han obtenido a partir de la posición del modelo real cuantizada, es decir, para obtener la velocidad y la aceleración fue necesario la aplicación de filtros no causales y la aplicación de filtros que limpien los ruidos.

La matriz γ obtenida que, multiplicada por los valores numéricos de θ , resultarán las ecuaciones dinámicas del robot estudiado, se muestra a continuación:

4.3.1. Robot medidas ideales con reductoras

Los parametros estimados, es decir, la matriz tetha linealmente independiente de 11 parámetros, obtenida para desarrollar el modelo del robot con medidas ideales con reductoras en los motores será:

Parametro estimado	Valor obtenido	Covarianza obtenida
$\theta(1)$	15.6322	0.0338
$\theta(2)$	0.0012	0.0218
$\theta(3)$	7.389	0.0481
$\theta(4)$	55.1139	0.00081
$\theta(5)$	0.00085	0.02744
$\theta(6)$	2.0841	0.047868
$\theta(7)$	-2.0414	0.00623
$\theta(8)$	0.051	0.00113
$\theta(9)$	0.0015	0.033
$\theta(10)$	-6.665	0.00054
$\theta(11)$	-2.222	0.00113

Por lo tanto, tras multiplicar γ y θ y derivar respecto a las variables articulares, se definirán las ecuaciones dinámicas del robot ideal con reductoras cómo:

$$\begin{pmatrix} Kt_1 R_1 I m_1 \\ Kt_2 R_2 I m_2 \\ Kt_3 R_3 I m_3 \end{pmatrix} = \begin{pmatrix} Ma_{1,1} & 0 & 0 \\ 0 & Ma_{2,2} & Ma_{2,3} \\ 0 & Ma_{3,2} & 2,463 \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix} + \begin{pmatrix} Va_{1,1} \\ Va_{2,1} \\ Va_{3,1} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 1,81\cos(q_2 + q_3) + 5,44\cos(q_2) \\ 4,15\cos(q_2 + q_3) \end{pmatrix}$$

- $Ma_{1,1} = 0,0889\cos(2q_2 + q_3) + 0,119\cos(2q_2) + 0,0889\cos(q_3) + 0,0294\cos(2q_2 + 2q_3) + 1,15$
- $Ma_{2,2} = 0,3703\cos(q_3) + 5,83$
- $Ma_{2,3} = 0,1851\cos(q_3) + 0,126$
- $Ma_{3,2} = 0,4232\cos(q_3) + 0,288$
- $Va_{1,1} = 0,178\dot{q}_1\dot{q}_2\sin(2q_2 + q_3) + 0,0887\dot{q}_1\dot{q}_3\sin(2q_2 + q_3) + 0,235\dot{q}_1\dot{q}_2\sin(2q_2) + 0,0887\dot{q}_3\dot{q}_1\sin(q_3) + 0,059\dot{q}_1\dot{q}_2\sin(2q_2 + 2q_3) + 0,059\dot{q}_1\dot{q}_3\sin(2q_2 + 2q_3) - 0,12\dot{q}_1$
- $Va_{2,1} = 0,0638\dot{q}_2 - 0,185\dot{q}_3^2\sin(q_3) + 0,0613\dot{q}_1^2\sin(2q_2 + 2q_3) + 0,185\dot{q}_1^2\sin(2q_2 + q_3) + 0,248\dot{q}_1^2\sin(2q_2) - 0,37\dot{q}_2\dot{q}_3\sin(q_3)$
- $Va_{3,1} = 0,0643\dot{q}_3 + 0,212\dot{q}_1^2\sin(q_3) + 0,423\dot{q}_2^2\sin(q_3) + 0,14\dot{q}_1^2\sin(2q_2 + 2q_3) + 0,212\dot{q}_1^2\sin(2q_2 + q_3)$

4.3.2. Robot medidas ideales sin reductoras

Los parametros estimados, es decir, la matriz tetha linealmente independiente de 11 parámetros, obtenida para desarrollar el modelo del robot con medidas ideales sin reductoras en los motores será:

Parametro estimado	Valor obtenido	Covarianza obtenida
$\theta(1)$	-9.31476	0.00364
$\theta(2)$	0.001193	2.868
$\theta(3)$	7.3803	0.00036
$\theta(4)$	-7.2341	0.00369
$\theta(5)$	0.00121	5.890
$\theta(6)$	2.078	0.00358
$\theta(7)$	-2.0335	0.00359
$\theta(8)$	0.051	0.0148
$\theta(9)$	0.00146	2.19
$\theta(10)$	-6.6585	0.003621
$\theta(11)$	-2.222	0.00356

Por lo tanto, tras multiplicar γ y θ y derivar respecto a las variables articulares, se definirán las ecuaciones dinámicas del robot ideal con reductoras cómo:

$$\begin{pmatrix} Kt_1 R_1 I m_1 \\ Kt_2 R_2 I m_2 \\ Kt_3 R_3 I m_3 \end{pmatrix} = \begin{pmatrix} Ma_{1,1} & 0 & 0 \\ 0 & Ma_{2,2} & Ma_{2,3} \\ 0 & Ma_{3,2} & 4,47 \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix} + \begin{pmatrix} Va_{1,1} \\ Va_{2,1} \\ Va_{3,1} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 54,3\cos(q_2 + q_3) + 163\cos(q_2) \\ 62,1\cos(q_2 + q_3) \end{pmatrix}$$

- $Ma_{1,1} = 4,434\cos(2q_2 + q_3) + 5,937\cos(2q_2) + 4,434\cos(q_3) + 1,469\cos(2q_2 + 2q_3) + 7,693$
- $Ma_{2,2} = 11,08\cos(q_3) + 18,99$
- $Ma_{2,3} = 5,542\cos(q_3) + 3,784$
- $Ma_{3,2} = 6,334\cos(q_3) + 4,325$
- $Va_{1,1} = -8,84\dot{q}_1\dot{q}_2\sin(2q_2 + q_3) - 4,429\dot{q}_1\dot{q}_3\sin(2q_2 + q_3) - 11,85\dot{q}_1\dot{q}_2\sin(2q_2) - 4,429\dot{q}_1\dot{q}_3\sin(q_3) - 2,924\dot{q}_1\dot{q}_2\sin(2q_2 + 2q_3) - 2,924\dot{q}_1\dot{q}_3\sin(2q_2 + 2q_3)0,002387\dot{q}_1$
- $Va_{2,1} = 0,00304\dot{q}_2 - 5,54\dot{q}_3^2\sin(q_3) + 1,84\dot{q}_1^2\sin(2q_2 + 2q_3) + 5,54\dot{q}_1^2\sin(2q_2 + q_3) + 7,42\dot{q}_1^2\sin(2q_2) - 11,1\dot{q}_2\dot{q}_3\sin(q_3)$
- $Va_{3,1} = 0,00418\dot{q}_3 + 3,17\dot{q}_1^2\sin(q_3) + 6,33\dot{q}_2^2\sin(q_3) + 2,1\dot{q}_1^2\sin(2q_2 + 2q_3) + 3,17\dot{q}_1^2\sin(2q_2 + q_3)$

4.3.3. Robot medidas reales con reductoras

Los parametros estimados, es decir, la matriz tetha linealmente independiente de 11 parámetros, obtenida para desarrollar el modelo del robot con medidas reales con reductoras en los motores será:

Parametro estimado	Valor obtenido	Covarianza obtenida
$\theta(1)$	16.995	4.0573
$\theta(2)$	0.00122	0.2538
$\theta(3)$	12.393	1.291
$\theta(4)$	38.28	1.9472
$\theta(5)$	0.00129	0.941
$\theta(6)$	1.434	0.917
$\theta(7)$	4.0372	4.545
$\theta(8)$	0.0491	1.234
$\theta(9)$	0.00151	1.468
$\theta(10)$	-6.6722	0.003858
$\theta(11)$	-2.199	0.008916

Por lo tanto, tras multiplicar γ y θ y derivar respecto a las variables articulares, se definirán las ecuaciones dinámicas del robot ideal con reductoras cómo:

$$\begin{pmatrix} Kt_1 R_1 I m_1 \\ Kt_2 R_2 I m_2 \\ Kt_3 R_3 I m_3 \end{pmatrix} = \begin{pmatrix} Ma_{1,1} & 0 & 0 \\ 0 & Ma_{2,2} & Ma_{2,3} \\ 0 & Ma_{3,2} & 2,78 \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix} + \begin{pmatrix} Va_{1,1} \\ Va_{2,1} \\ Va_{3,1} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 1,796\cos(q_2 + q_3) + 5,449\cos(q_2) \\ 4,105\cos(q_2 + q_3) \end{pmatrix}$$

- $Ma_{1,1} = 0,088\cos(2q_2 + q_3) + 0,019\cos(2q_2) + 0,088\cos(q_3) + 0,0417\cos(2q_2 + 2q_3) + 1,29$
- $Ma_{2,2} = 0,366\cos(q_3) + 4,6$
- $Ma_{2,3} = 0,183\cos(q_3) + 0,293$
- $Ma_{3,2} = 0,419\cos(q_3) + 0,67$
- $Va_{1,1} = -0,1758\dot{q}_1\dot{q}_2\sin(2q_2 + q_3) - 0,08791\dot{q}_1\dot{q}_3\sin(2q_2 + q_3) - 0,03796\dot{q}_1\dot{q}_2\sin(2q_2) - 0,08791\dot{q}_1\dot{q}_3\sin(q_3) - 0,08325\dot{q}_1\dot{q}_2\sin(2q_2 + 2q_3) - 0,08325\dot{q}_1\dot{q}_3\sin(2q_2 + 2q_3) + 0,1223\dot{q}_1$
- $Va_{2,1} = 0,0969\dot{q}_2 - 0,183\dot{q}_3^2\sin(q_3) + 0,0868\dot{q}_1^2\sin(2q_2 + 2q_3) + 0,183\dot{q}_1^2\sin(2q_2 + q_3) + 0,0396\dot{q}_1^2\sin(2q_2) - 0,366\dot{q}_2\dot{q}_3\sin(q_3)$
- $Va_{3,1} = 0,0648\dot{q}_3 + 0,209\dot{q}_1^2\sin(q_3) + 0,419\dot{q}_2^2\sin(q_3) + 0,199\dot{q}_1^2\sin(2q_2 + 2q_3) + 0,209\dot{q}_1^2\sin(2q_2 + q_3)$

4.3.4. Robot medidas reales sin reductoras

Los parametros estimados, es decir, la matriz tetha linealmente independiente de 11 parámetros, obtenida para desarrollar el modelo del robot con medidas reales sin reductoras en los motores será:

Parametro estimado	Valor obtenido	Covarianza obtenida
$\theta(1)$	-11.07265	2.715
$\theta(2)$	0.018174	0.54845
$\theta(3)$	7.753	0.32937
$\theta(4)$	-7.3756	1.1879
$\theta(5)$	0.060807	3.259
$\theta(6)$	-2.19164	1.3296
$\theta(7)$	-1.0052	2.889
$\theta(8)$	0.10404	4.1499
$\theta(9)$	0.006486	4.0307
$\theta(10)$	-6.7508	1.2175
$\theta(11)$	0.00374	4.972

Sin embargo, cabe destacar que el ultimo parametro, debido a que durante la simulacion se observó que el modelo era malo, se ha optado por realizar un reajuste del mismo a posteriori. Debido a que es un parámetro al que solo le afectan los terminos gravitatorios, se pudo reajustar dandole un valor de -2.17 aproximadamente. Por lo tanto, para la defensa de éste proyecto, no en el controlador que semostrará en el analisis de los controladores, se ha realizado éste ajuste al último parámetro asociado a términos gravitatorios.

Por lo tanto, tras multiplicar γ y θ y derivar respecto a las variables articulares, se definirán las ecuaciones dinámicas del robot ideal con reductoras cómo:

$$\begin{pmatrix} Kt_1 R_1 I m_1 \\ Kt_2 R_2 I m_2 \\ Kt_3 R_3 I m_3 \end{pmatrix} = \begin{pmatrix} Ma_{1,1} & 0 & 0 \\ 0 & Ma_{2,2} & Ma_{2,3} \\ 0 & Ma_{3,2} & 7,56 \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix} + \begin{pmatrix} Va_{1,1} \\ Va_{2,1} \\ Va_{3,1} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 54,317\cos(q_2 + q_3) + 165,4\cos(q_2) \\ 62,076\cos(q_2 + q_3) \end{pmatrix}$$

- $Ma_{1,1} = 4,434\cos(2q_2 + q_3) + 5,7477\cos(2q_2) + 4,434\cos(q_3) + 5,7389\cos(2q_2 + 2q_3) + 0,46597$
- $Ma_{2,2} = 11,085\cos(q_3) + 21,67$
- $Ma_{2,3} = 5,5425\cos(q_3) + 6,3549$
- $Ma_{3,2} = 6,3343\cos(q_3) + 7,2628$
- $Va_{1,1} = -8,8682\dot{q}_1\dot{q}_2\sin(2q_2+q_3) - 4,4341\dot{q}_1\dot{q}_3\sin(2q_2+q_3) - 11,4956\dot{q}_1\dot{q}_2\sin(2q_2) - 4,4340\dot{q}_1\dot{q}_3\sin(q_3) - 11,4779\dot{q}_1\dot{q}_2\sin(2q_2 + 2q_3) - 11,4779\dot{q}_1\dot{q}_3\sin(2q_2 + 2q_3) + 0,0363$
- $Va_{2,1} = 0,15202\dot{q}_2 - 5,5425\dot{q}_3^2\sin(q_3) + 7,1736\dot{q}_1^2\sin(2q_2+2q_3) + 5,5425\dot{q}_1^2\sin(2q_2+q_3) + 7,1847\dot{q}_1^2\sin(2q_2) - 11,085\dot{q}_2\dot{q}_3\sin(q_3)$
- $Va_{3,1} = 0,018533\dot{q}_3 + 3,1671\dot{q}_1^2\sin(q_3) + 6,3343\dot{q}_2^2\sin(q_3) + 8,1984\dot{q}_1^2\sin(2q_2+2q_3) + 3,1671\dot{q}_1^2\sin(2q_2 + q_3)$

4.4. Verificación de los modelos obtenidos

Tras obtener éstos modelos, se hará una simple prueba antes de comenzar a controlar el robot para poder analizar la bondad del modelo obtenido.

Este experimento consistirá en darle un valor de unitario a la intensidad, es decir, valores constantes unitarios a todas las articulaciones.

Se ha optado por, para comprobar dicha bondad, introducirle al robot dado por el profesor, es decir, el robot real valores de intensidades unitarios y lo mismo al modelo, de ese modo se podrá observar y comparar la respuesta del robot real y el modelo obtenido.

Cabe destacar que, debido a que no se controlará el robot durante más de 5 segundos aproximadamente, durante ese tiempo es dónde interesa observar la bondad del modelo.

4.4.1. Robot medidas ideales con reductoras

En primer lugar, se mostrará cómo responden las variables articulares del modelo obtenido frente a las del robot real. La gráfica que se muestra es la obtenida empleando las medidas ideales de éstas variables articulares tanto del robot real cómo del modelo:

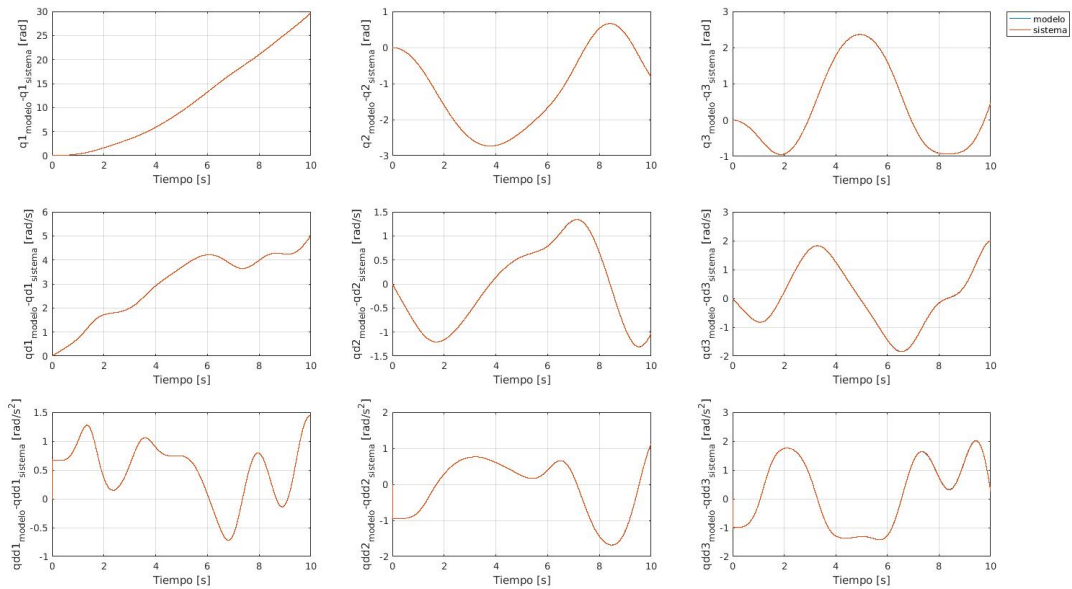


Figura 7: Comparativa Variables articulares del modelo obtenido con medidas ideales y reductoras

Cómo se puede observar, la magnitud del error será muy baja, por tanto, eso conllevará que se han estimado los parámetros dinámicos del robot con un bajo error entre los parámetros reales.

A continuación, se mostrará la gráfica del error en las variables articulares:

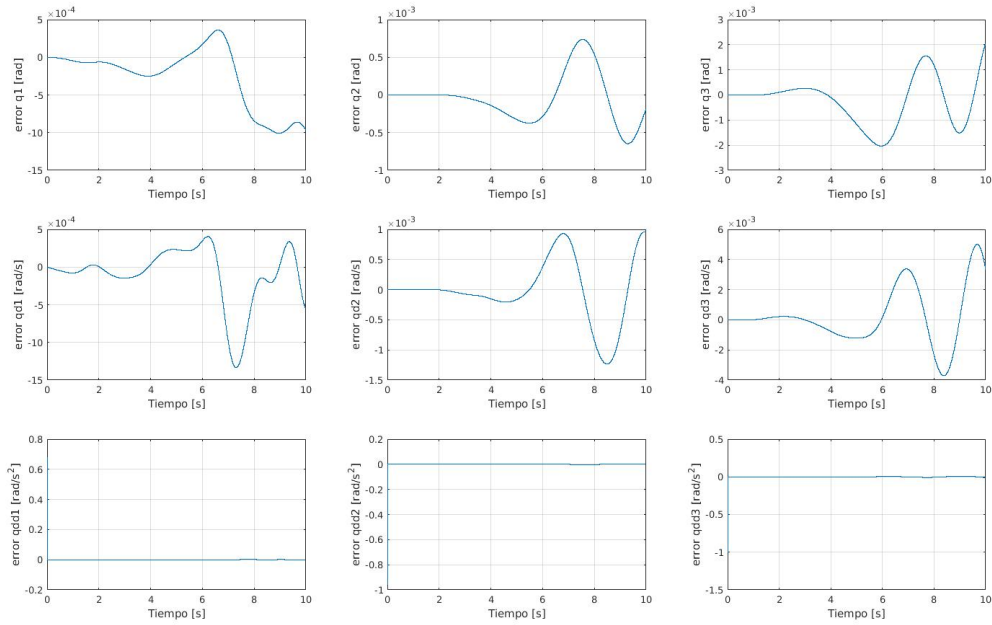


Figura 8: Error del modelo obtenido con medidas ideales y reductoras

4.4.2. Robot medidas ideales sin reductoras

En éste caso, se realizará la misma comparativa que en el caso anterior, con la salvedad es que en el modelo obtenido ahora, se suponen que no se tienen reductoras.

Cabe destacar que nos interesa el modelo del robot a bjos tiempos, es decir, los primeros 5 segundos de la simulación, pues no se va a querer controlar el robot en trayectorias que duren más tiempo. Se ha graficado más tiempo para que se observe que, cuando empieza a pasar más del tiempo que se desea controlar, los errores comienzan a incrementarse.

A continuación, se mostrará la gráfica resultando al excitar el modelo del sistema y el robot real con intensidades constantes y unitarias.

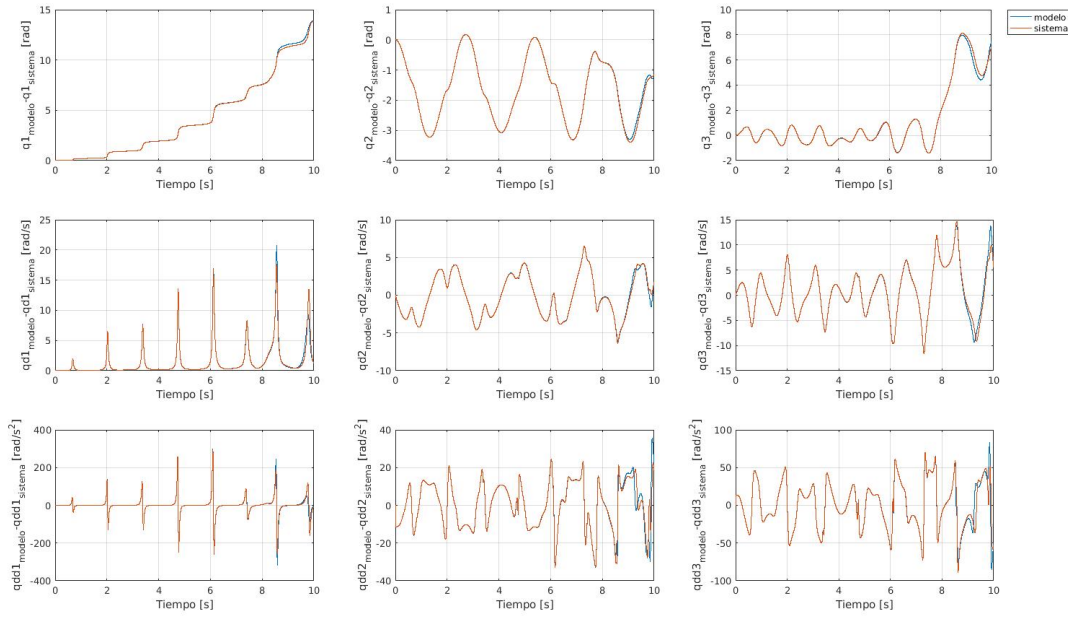


Figura 9: Comparativa Variables articulares del modelo obtenido con medidas ideales y sin reductoras

Cómo se puede observar, al igual que antes, cuándo se emplean medidas ideales para estimar los parámetros dinámicos del robot y obtener un modelo del mismo, se obtendrán buenos modelos, debido a que cómo se indico antes, las medidas son ideales.

Por tanto, a continuación se mostrará la gráfica de los errores de las variables articulares:

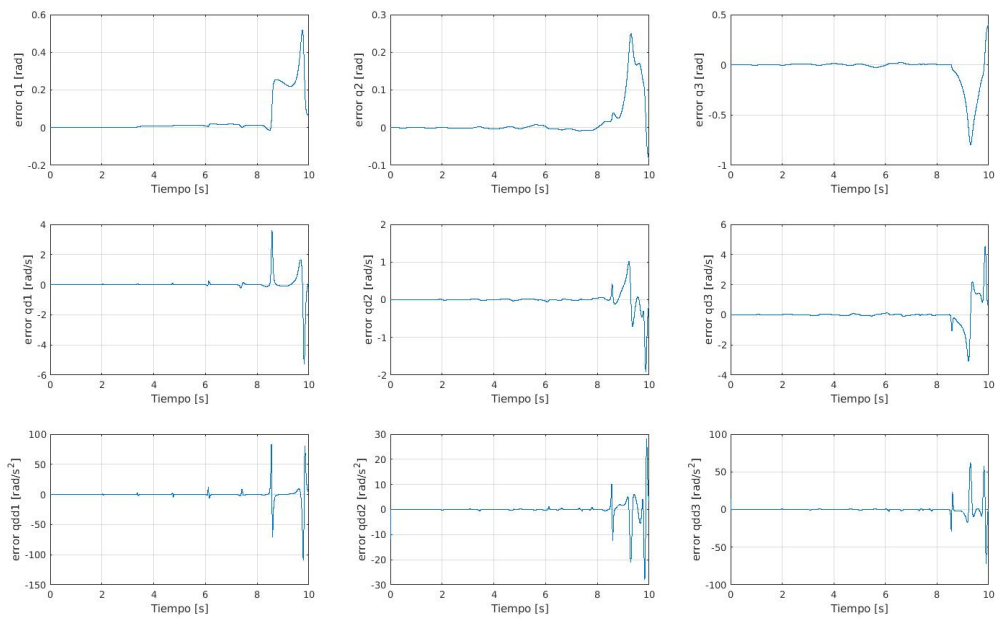


Figura 10: Error del modelo obtenido con medidas ideales y sin reductoras

4.4.3. Robot medidas reales con reductoras

Aunque se haya obtenido el modelo con medidas reales suponiendo que no se tienen tacómetros y que, por tanto, no se puede conocer la velocidad articular del robot, para realizar análisis de control y para analizar el modelo obtenido sí se conocerá la velocidad articular, es decir, se supondrá que se tienen tacómetros. De ese modo se evitará la necesidad de implementar un filtro no causal.

Los filtros no causales se caracterizan por el hecho de se emplean valores futuros de la señal, algo que sólo se podrá hacer de manera computacional una vez se hayan tomado todos los datos.

Por tanto, se compararán las medidas reales de velocidad y posición del robot con las medidas ideales del modelo obtenido. La grafica comparativa se muestra a continuación:

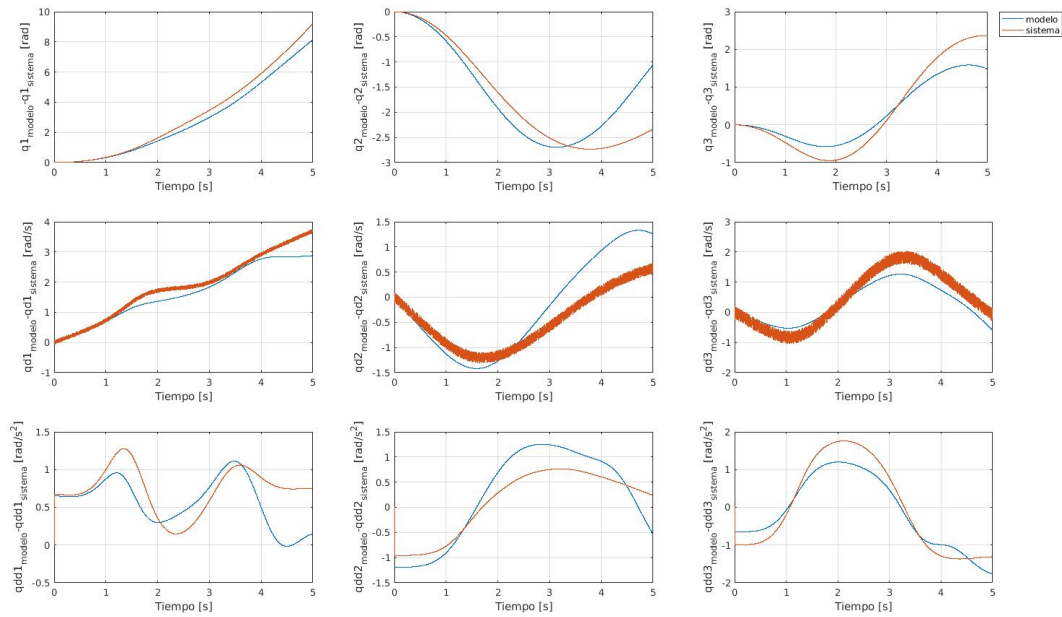


Figura 11: Comparativa Variables articulares del modelo obtenido con medidas reales y reductoras

En éste caso, se observa cómo al emplear medidas reales para obtener el modelo dinámico del robot y al haber tenido que emplear filtros computacionales para conocer velocidades y aceleraciones del robot, el modelo obtenido no será tan bueno como en el caso en el que se usaron medidas ideales. Sin embargo, debido a que no posee un elevado orden de magnitud, se asumirá dicho error y se tomará el modelo como válido.

Ademas de ello, se observa que aparece cierto desfase entre el modelo obtenido y el robot real, éste desfase es debido a la aplicación de filtros a la señal. Cuando se emplean filtros hay que tener en cuenta que éstos introducen un cierto desfase, por tanto, será necesario encontrar un compomiso entre cuán bueno se quiere que sea el filtro y cuánto desfase se está dispuesto a asumir. En éste caso, debido a que nos encontramos en un proyecto basado en la simulación, se aceptará un cierto desfase a cambio de haber diseñado buenos filtros para las señales.

Por tanto, la magnitud de los errores obtenidos se muestra a continuación:

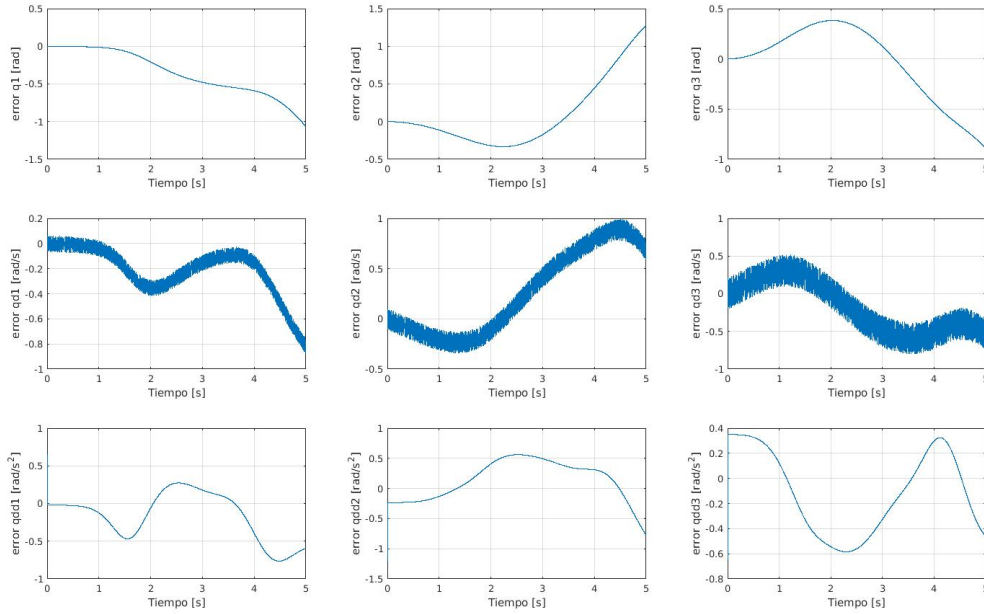


Figura 12: Error del modelo obtenido con medidas reales y reductoras

4.4.4. Robot medidas reales sin reductoras

Aunque se haya obtenido el modelo con medidas reales suponiendo que no se tienen tacómetros y que, por tanto, no se puede conocer la velocidad articular del robot, para realizar análisis de control y para analizar el modelo obtenido sí se conocerá la velocidad articular, es decir, se supondrá que se tienen tacómetros. De ese modo se evitará la necesidad de implementar un filtro no causal.

Los filtros no causales se caracterizan por el hecho de se emplean valores futuros de la señal, algo que sólo se podrá hacer de manera computacional una vez se hayan tomado todos los datos.

Por tanto, se compararán las medidas reales de velocidad y posición del robot con las medidas ideales del modelo obtenido. La grafica comparativa se muestra a continuación:

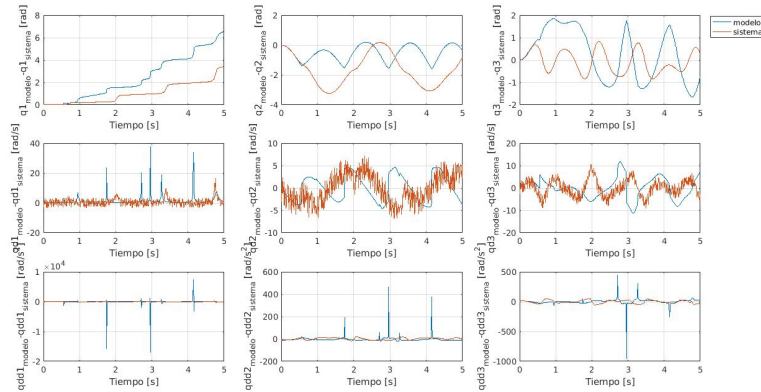


Figura 13: Comparativa Variables articulares del modelo obtenido con medidas reales sin reductoras

debido a que este modelo es el que se encuentra más acoplado y, ha sido necesario la aplicación de filtros se observa que el modelo no será un buen modelo, ya que aunque en algunos valores pueda parecer que se asemeje, el error entre las medidas del robot real y el modelo obtenido es significativo, cómo se muestra a continuación:

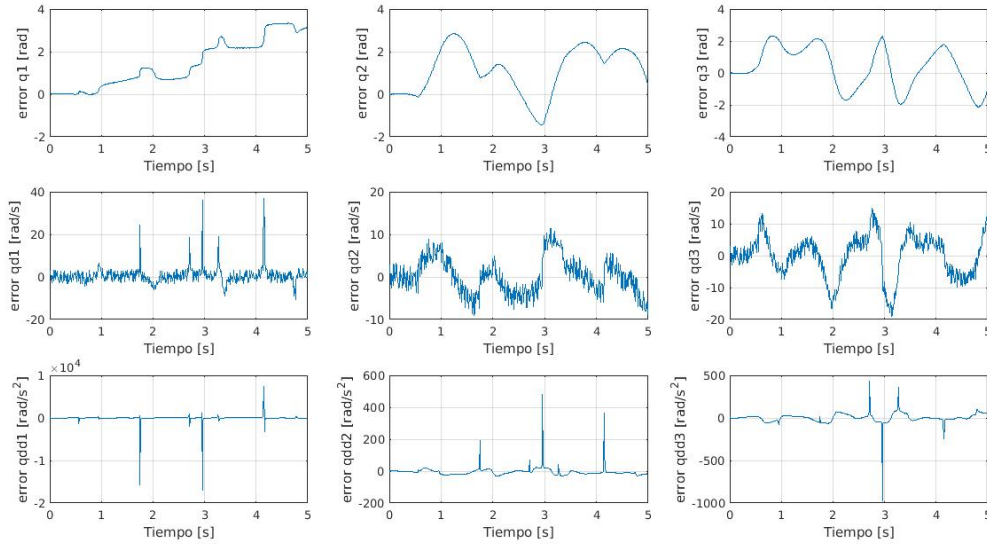


Figura 14: Error del modelo obtenido con medidas reales y sin reductoras

Es por ello que, aunque se diseñarán controladores para éste modelo, ya que de ese modo se verá que se obtienen malos resultados, no será posible asumir éste modelo cómo válido. Finalmente, una posible opción, sería realizar un reajuste del modelo a posteriori, es decir, a partir de las gráficas en las que se compara la respues de éste modelo y el robot real, intentar estimar qué parametros del robot están peor estimados en el modelo e intentar compensarlos. Sin embargo, debido a a la falta de tiempo ésta técnica no ha sido aplicada.

5. Control Dinámico del brazo

En esta última parte del proyecto, una vez se conoce el modelo del robot en las diferentes configuraciones, se pasará a buscar implementar un control dinámico sobre el mismo. Para poder implementar controladores sobre nuestro robot, será necesario obtener una función de transferencia matemática a modo de modelo que se asemeje al robot real.

Una vez se tenga un modelo lineal de cada articulación del robot, junto con el generador de trayectorias creado anteriormente, se buscará que el robot siga una trayectoria predefinida. Básicamente, se busca que el robot se desplace de un punto a otro minimizando el error y del modo y a la velocidad que el usuario desee.

5.1. Obtención del modelo lineal de las articulaciones del brazo

Para obtener la función de transferencia de cada articulación del robot, se linealizará la ecuación dinámica que define cada motor en un punto de equilibrio en torno a velocidades nulas. Por lo tanto, las consideraciones que se tendrán en cuenta para linealizar la ecuación dinámica que define el comportamiento de cada articulación del robot son:

- Velocidades de equilibrio

$$\begin{aligned} \dot{q}_{eq} &= 0 \text{ rad/s} \\ \dot{q} &= \dot{q}_{eq} + \Delta \dot{q} \end{aligned}$$

- Aceleraciones de equilibrio

$$\begin{aligned} \ddot{q}_{eq} &= 0 \text{ rad/s} \\ \ddot{q} &= \ddot{q}_{eq} + \Delta \ddot{q} \end{aligned}$$

Además de ello, se aplicarán una serie de simplificaciones a la ecuación dinámica. A continuación, se mostrarán las ecuaciones dinámicas de los motores:

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} Kt_1 R_1 I m_1 \\ Kt_2 R_2 I m_2 \\ Kt_3 R_3 I m_3 \end{pmatrix} = \begin{pmatrix} Ma_{11} & Ma_{12} & Ma_{13} \\ Ma_{21} & Ma_{22} & Ma_{23} \\ Ma_{31} & Ma_{32} & Ma_{33} \end{pmatrix} \ddot{q} + \begin{pmatrix} Va_1 \\ Va_2 \\ Va_3 \end{pmatrix} \dot{q} + \begin{pmatrix} Ga_1 \\ Ga_2 \\ Ga_3 \end{pmatrix}$$

donde se asume que dentro de los términos de inercia y de Coriolis se han tenido en cuenta las inercias y fricciones viscosas de los motores.

La primera simplificación del modelo que se hará para poder linealizar el modelo en torno a un punto de operación, será suponer la matriz de inercias diagonal y, además de ello, se cogerá el valor medio de todos los senos y cosenos de tal modo que únicamente se tomen los valores de inercias medios. De manera práctica, se harán cero todas las variables articulares del modelo. De éste modo, se desacoplará el sistema. En cuando a la matriz de términos de Coriolis, únicamente aportarán a la linealización la fricción viscosa de los motores, es decir, se despreciarán todos los términos exceptuando el que acompañe al valor de la velocidad articular.

Por último, la gravedad se despreciará para obtener un modelo, de tal modo que, se emplearán las siguientes ecuaciones para obtener los modelos de las articulaciones del robot:

$$\begin{pmatrix} Kt_1 R_1 I m_1 \\ Kt_2 R_2 I m_2 \\ Kt_3 R_3 I m_3 \end{pmatrix} = \begin{pmatrix} Ma_{11} & 0 & 0 \\ 0 & Ma_{22} & 0 \\ 0 & 0 & Ma_{33} \end{pmatrix} \ddot{q} + \begin{pmatrix} Va_1 \\ Va_2 \\ Va_3 \end{pmatrix} \dot{q} \quad (20)$$

A continuación, se obtendrá el modelo de la primera articulación y, el procedimiento será análogo para las restantes:

$$Kt_1 R_1 I m_1(t) = M a_{11} \ddot{q}_1(t) + V a_1 \dot{q}_1(t)$$

Se realizará una transformación al dominio de Laplace y, posteriormente, se expresará en forma de función de transferencia:

$$Kt_1 R_1 I m_1(s) = s^2 M a_{11} q_1(s) + s V a_1 q_1(s) \rightarrow \frac{q_1(s)}{I m_1(s)} = \frac{Kt_1 R_1}{s(M a_{11} s + V a_1)} \quad (21)$$

Por lo tanto, se definirá el modelo de cada articulación cómo:

$$G_1(s) = \frac{Kt_1 R_1}{s(M a_{11} s + V a_1)} \quad G_2(s) = \frac{Kt_2 R_2}{s(M a_{22} s + V a_2)} \quad G_3(s) = \frac{Kt_3 R_3}{s(M a_{33} s + V a_3)} \quad (22)$$

5.2. Diseño de controladores

En éste apartado, se analizará cómo se harán los controladores que, posteriormente se implementarán sobre el robot para hacer que se desplace a lo largo de una trayectoria que se generará mediante el control cinemático.

Cabe destacar que, en los controladores que se implementen junto con el compensador de gravedad, el de dinámica o el par calculado, la realimentación podrá ser por referencia, en lugar de emplear las medidas reales, de tal modo que se realimente con una señal sin ruido ni errores.

5.2.1. Controlador PD/PID

Para diseñar éstos controladores lineales, se empleará el modelo anteriormente mostrado y se implementarán directamente. El esquema de control será el esquema clásico de control, ya que éste tipo de controladores serán lo primeros en implementar y, por extensión, los más sencillos.

La función de transferencia de los controladores a implementar será:

$$C(s) = K_P(T_D s + 1) \quad C(s) = K_P \frac{T_D T_I s^2 + T_I s + 1}{T_I s} \quad (23)$$

5.2.2. Controlador PD/PID con compensación de gravedad

Para implementar éste controlador, se parte de la base de que, aunque la gravedad es una perturbación mantenida, se puede modelar, ya que se conoce de la obtención del modelo dinámico los efectos de la gravedad en el modelo del robot.

Por tanto, para implementar un controlador con compensación de gravedad se le sumará a la señal de control generada por el controlador diseñado anteriormente, los efectos de la gravedad en el robot.

Este bloque que añade los efectos de la gravedad tendrá como entrada la posición actual del robot (podría tener la referencia de posición) y la salida será la compensación de la señal de control. El esquema de montaje de éste tipo de control se muestra a continuación:

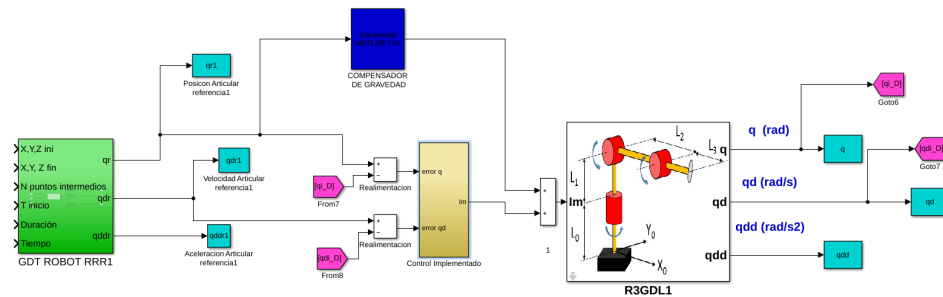


Figura 15: Diagrama de control del compensador de gravedad

5.2.3. Controlador PD/PID con compensación de dinámica (Feedforward)

La implementación de controladores por precompensación de dinámica será la primera vez que se implementen controladores basados en modelo. La principal diferencia de éste tipo de controladores es que dependen en gran parte de la bondad del modelo obtenido, ya que se realimentará con el modelo dinámico inverso obtenido. Sin embargo, cabe destacar que en control de brazos manipuladores se emplea bastante debido a que se suele lograr obtener buenos modelos dinámicos de los robots.

Para implementar éste controlador, también conocido como *Feedforward*, se modificará el modelo de control de tal modo que se precompensen los efectos del modelo dinámico completo del robot, no solo la gravedad.

$$I_m = M_A(q)q_{ref}'' + C(q, \dot{q})\dot{q} + G_A(q) + u \quad (24)$$

cómo se observa, la señal de control generada estará formada por el modelo dinámico del robot más una señal adicional, u .

Para conocer el valor de esa señal de control adicional, se restará al modelo del sistema la señal de control que se desea generar de tal modo que se obtenga la expresión del bucle cerrado interno de control.

$$\begin{aligned} Im &= M_A(q)\ddot{q} + C_A(q, \dot{q})\dot{q} + G_A(q) \\ - \quad Im &= M_A(q)q_{ref}'' + C_A(q, \dot{q})\dot{q} + G_A(q) + u \\ \hline M_A(q)\ddot{q} &= u \end{aligned} \quad (25)$$

de ese modo se ha obtenido la señal de control adicional, también conocida cómo la dinámica del error. Aunque no se lea correctamente, la ecuación obtenida es: $M_A(q)\ddot{q} = u$, dónde \ddot{q} será el error en aceleración. Habrá que diseñar controladores para la función de transferencia que se obtendrá a continuación. Para obtener dicha función de transferencia será necesario partir de condiciones iniciales nulas y transformarla al dominio de *Laplace*.

$$M_A(q)\ddot{q}(t) = u(t) \rightarrow M_A(q)\ddot{q}(s)s^2 = u(s) \rightarrow \frac{\ddot{q}(s)}{u(s)} = \frac{K_t R}{M_A s^2} \left[\frac{ud.error}{ud.sc} \right] \quad (26)$$

Se deberán diseñar tres funciones de transferencia, una por articulación. El esquema en diagrama de bloques de éste controlador se muestra a continuación:

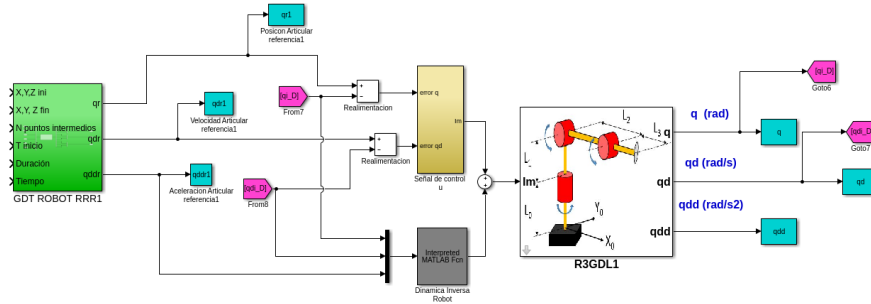


Figura 16: Esquema de un controlador Feed Forward

debido a que no se puede medir la aceleración de las variables articulares, cómo se observa, el modelo dinámico inverso del robot se alimentará con los valores de las variables articulares en posición y velocidad y en el caso de la aceleración, se alimentará con la referencia obtenida del generador de trayectorias.

Por lo tanto, a modo de resumen, las funciones de transferencia de las cuales hará que obtener controladores son:

$$G_1(s) = \frac{K_{t1}R_1}{M_{a11}s^2} \quad G_2(s) = \frac{K_{t2}R_2}{M_{a22}s^2} \quad G_3(s) = \frac{K_{t3}R_3}{M_{a33}s^2} \quad (27)$$

5.2.4. Controlador PD/PID con par calculado

El control mediante par calculado será el último en implementar en éste proyecto. El par calculado se basa en la búsqueda de desacoplar totalmente las interacciones del robot, resultando la dinámica del error cómo un doble integrador, para el cuál habrá que diseñar controladores.

Que las articulaciones de un robot estén desacopladas implica que un par en un determinado actuador únicamente afectará al movimiento de dicha articulación.

Por tanto, la diferencia de éste controlador frente al resto es que es un controlador dinámico. Además, es un controlador totalmente basado en el modelo, lo que conlleva que si el modelo es malo el control también lo será. Si, al igual que antes, se resta al sistema la señal de control que se desea implementar, se obtendrá la dinámica del error:

$$\begin{aligned} Im &= M_A(q)\ddot{q} + C_A(q, \dot{q})\dot{q} + G_A(q) \\ - \quad Im &= M_A(q)(\ddot{q}_{ref} + u) + C_A(q, \dot{q})\dot{q} + G_A(q) \\ \hline \ddot{q} &= u \end{aligned} \quad (28)$$

Por tanto, transformando la dinámica del error al dominio de *Laplace*, las funciones de transferencia a partir de las cuales hay que diseñar los controladores serán:

$$G_1(s) = \frac{1}{s^2} \quad G_2(s) = \frac{1}{s^2} \quad G_3(s) = \frac{1}{s^2} \quad (29)$$

Y, el esquema de control del par calculado se muestra a continuación:

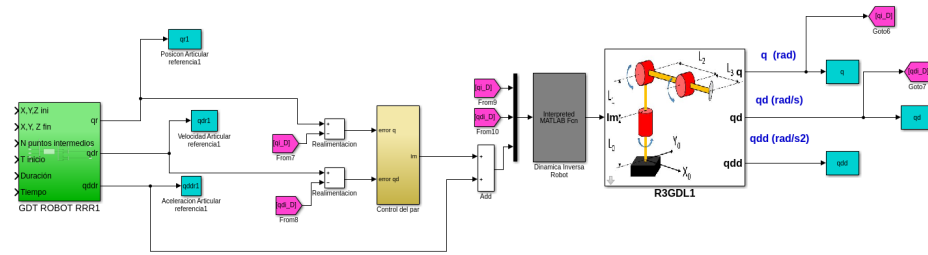


Figura 17: Esquema de un controlador Par Calculado

5.3. Análisis de experimentos para probar controladores

En éste último apartado, se diseñarán diversos experimentos de control en los cuales se acentuen los efectos de un controlador u otro, por ejemplo, se realizarán trayectorias en las cuales se noten los efectos gravitatorios y así se observe la mejoría del controlador empleando un compensador de gravedad, etcétera. Por tanto, los controladores que se van a implementar son los anteriores y los modelos a partir de los cuales se obtendrán serán el ideal y el real con y sin reductoras. También se diseñarán experimentos en el que se comparen los modelos obtenidos con medidas ideales y reales.

Por consiguiente, se realizarán los siguientes experimentos, en los cuales se definirá qué controladores se desea comparar y con qué fin:

- En primer lugar, se realizará una comparativa entre los controladores PD y PID, de tal modo que se acentúe el efecto integral del PID y se observe la diferencia entre ambos controladores. Para ello, se generará una trayectoria relativamente lenta en la dirección negativa del eje Z, es decir, se partirá de la posición inicial del robot y se hará descender el brazo.
- Debido a que la principal perturbación que se tendrá es la gravedad, la cuál es una perturbación mantenida y conocida, la acometida del efecto integral será contrarrestar ésta perturbación mantenida. Sin embargo, el principal problema del efecto integral es la agresividad del mismo. Por tanto, en éste caso, se comprobará si el error es menor al emplear un control PD con compensación de gravedad o un PID. En un ambiente real, teóricamente sería más correcto emplear el PD con la compensación para evitar los "latigazos" que podría llegar a generar en el robot el efecto integral. Sería conveniente para poder mostrar más correctamente éstos resultados buscar experimentos en los cuales se acentuen los efectos de la gravedad, para ello, lo ideal sería generar una trayectoria similar a la anterior pero en éste caso se emplearán los controladores diseñados para el robot real y se realimentarán con medidas reales. Es posible que, para mejorar el control, en lugar de realimentar el compensador de gravedad con la medida real de posición sea conveniente realimentarlo con la referencia de posición obtenida del generador de trayectorias.
- Posteriormente, se analizará el primer controlador basado en modelo, el control *FeedForward*. En primer lugar sería conveniente realizar un análisis de cómo afecta la bondad del modelo a éste controlador comparando robot reales y robots ideales. Tras ello, se comprobará éste control realizando experimentos dónde se afectúen los términos de Coirolis, para ello sería conveniente realizar trayectorias
- Finalmente, se analizará el controlador par calculado, el cuál se basará en desacoplar totalmente las articulaciones del robot, de tal modo que el modelo de cada articulación resulte en un doble integrador. Sería conveniente analizar trayectorias en las que se acentúen los términos inerciales del brazo robótico, ya que en éste caso se busca compensar todo el modelo. Teóricamente, si el modelo obtenido es buen modelo, debería ser el mejor controlador implementado.
- Además de todo, sería conveniente comparar el resultado del emplear una realimentación de los controladores con medidas reales o con referencia, ya que al realimentar con la referencia, se obtendrá una medida mucho más limpia y que, como es de esperar, generará un menor error. Este estudio sería conveniente en el caso de emplear los modelos con medidas reales obtenidos.

5.3.1. Comparativa controladores PD-PID ideales con reductoras

En éste experimento, se compararán los controladores diseñados a partir del modelo ideal con reductoras.

La trayectoria que se empleará para éste experimentos se muestra a continuación gráficamente:

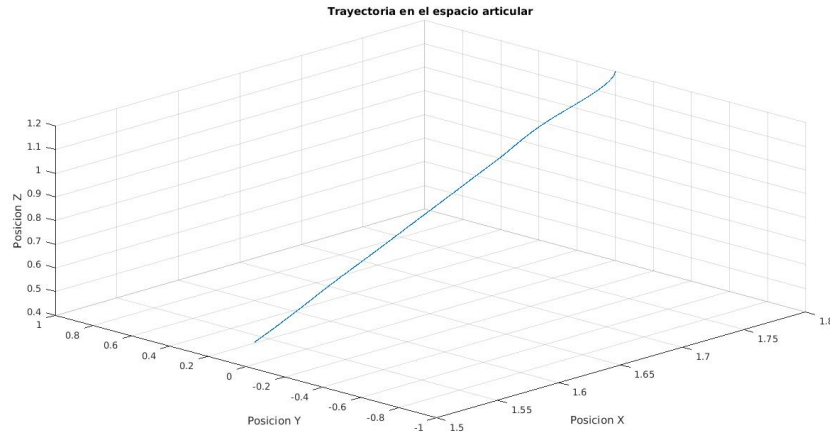


Figura 18: Trayectoria diseñada para evaluar el efecto integral de los controladores

Cómo se ha dicho anteriormente, con éste experimento se busca evaluar el efecto integral del controlador, lo que conllevará que cuando se implemente el PID el error en regimen permanente tenderá a ser nulo, sin embargo en el caso del PD se tendrá un error mantenido.

Se comparará el error en posición de las tres articulaciones, ya que es dónde se observa dicho efecto:

■ Controlador PD

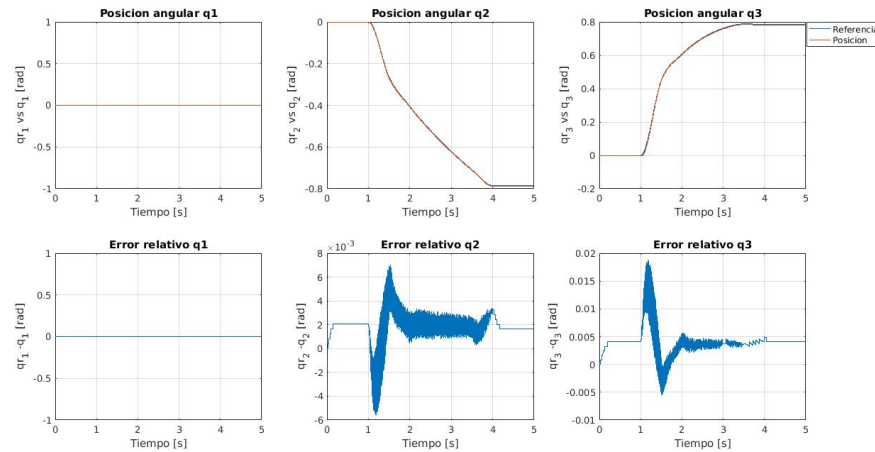


Figura 19: Seguimiento de referencia en posición con el controlador PD

■ Controlador PID

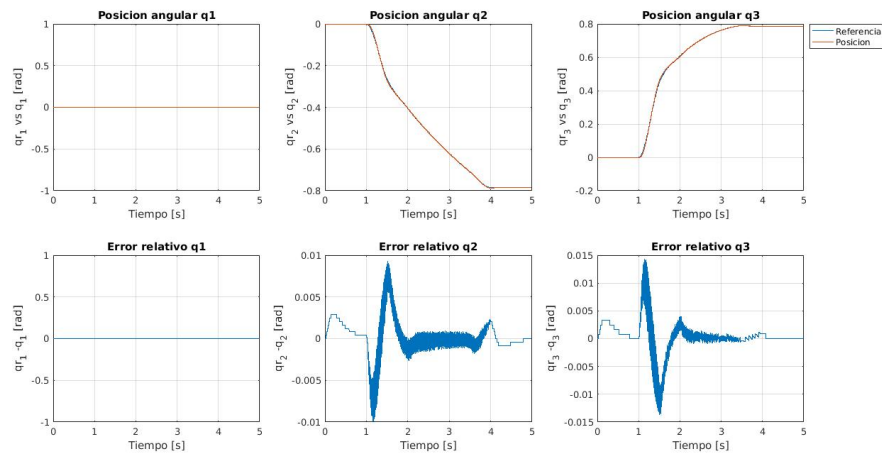


Figura 20: Seguimiento de referencia en posición con el controlador PID

Se observa claramente cómo el efecto integral compensa la perturbación mantenida que es la gravedad, por tanto, el controlador PID mejora la respuesta del PD asumiendo la agresividad del efecto integral.

5.3.2. Comparativa controlador PID y PD con compensación gravedad

En éste caso, se implementarán los controladores diseñados en base al modelo del robot real con reductoras. Debido a que se han empleado medidas reales, se realimentarán con las medidas reales del robot. Si la respuesta no fuese lo suficientemente aceptable, se alimentaría el compensador de gravedad con la referencia obtenida del generador de trayectorias.

Se ha optado por emplear la misma trayectoria que anteriormente debido a que se busca que se acentuen los efectos asociados a los terminos gravitatorios, al igual que antes.

■ Mejora del resultado obtenido con el controlador anterior

Si se compara el resultado obtenido anteriormente con el PID a partir del modelo ideal con el resultado obtenido al implementar un controlador PD con compensador de gravedad, se observa que éste segundo nos dará unos mejores resultados en el seguimiento de la trayectoria y una menor magnitud del error. La grafica de seguimiento se muestra a continuación:

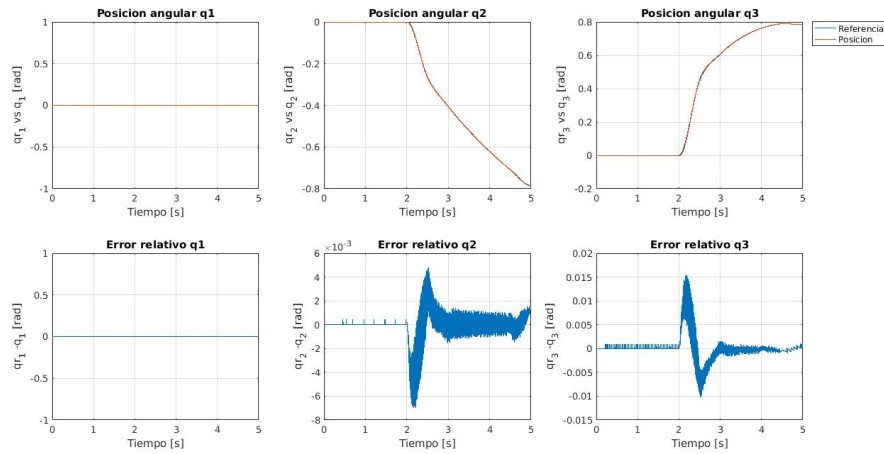


Figura 21: Seguimiento de referencia en posición con el controlador PD con compensación de gravedad

5.3.3. Comparativa del control con compensación de dinámica o *FeedForward*

En éste apartado, se comprobará la bondad de los modelos obtenidos. Para ello, implementará la primera técnica de control avanzado que se dispone, la cuál es el control con compensación de dinámica. Por tanto, para éste controlador se diseñará un controlador PD o PID que compense la dinámica del error y se compensará la dinámica del robot, en concreto los términos de Coirolis.

Se deberá buscar un experimento en el cuál se acentúen los terminos de Coirolis, por ello se ha optado por emplear el que se muestra a continuación, el cuál es un movimientos que contiene giro sobre la base en un tiempo medio, es decir, ni excesivamente rápido ni lento.

La trayectoria que se empleará para éste experimento partirá de las variables articulares nulas y girará sobre sí mismo bajando el brazo, intentando acentuar los términos de Coirolis.

Inicialmente, se comparará la bondad de los modelos obtenidos con reductoras. Por tanto, a continuación se mostrará el seguimiento de la trayectoria en el plano XYZ y el seguimiento de la referencia en posición de las variables articulares, tras ello, se sacarán una serie de conclusiones:

■ Robot ideal con reductoras

El seguimiento de la trayectoria en el plano XYZ será:

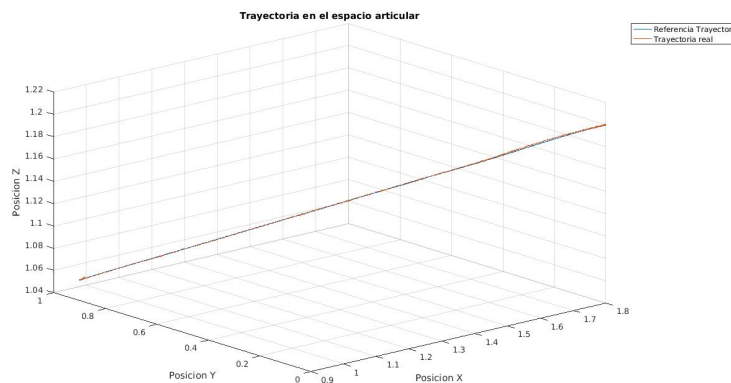


Figura 22: Seguimiento de la trayectoria en el plano XYZ

Se observa cómo, a primera vista, se ha obtenido un buen modelo, ya que el seguimiento será bastante bueno. A continuación se verá cómo siguen la referencia de posición las variables articulares del robot:

Aunque se aprecia un cierto error, la magnitud de éste es pequeña y, por tanto, fácilmente asumible. También cabe destacar que, la segunda articulación quizá se haya estimado algo peor, ya que es la que presenta error en regimen permanente tras la trayectoria.

■ Robot real con reductoras

El seguimiento de la trayectoria en el plano XYZ será:

En éste caso, debido a que se han tomado medidas reales para obtener el modelo del robot, se observa que se tendrá un error mantenido en el seguimiento de la trayectoria en el plano XYZ. Para analizar con mayor exactitud dicho error, es conveniente observar el seguimiento a referencia en posición realizado por las variables articuales del robot:

Se puede observar que todas las variables articulares tras la trayectoria presentarán un error mantenido, sin embargo, la magnitud del mismo será pequeña.

Por tanto, se puede estimar que no se ha obtenido un mal modelo real del robot con reductoras, aunque no es perfecto, ni tan bueno cómo el ideal, será un modelo válido para trabajar con el mismo.

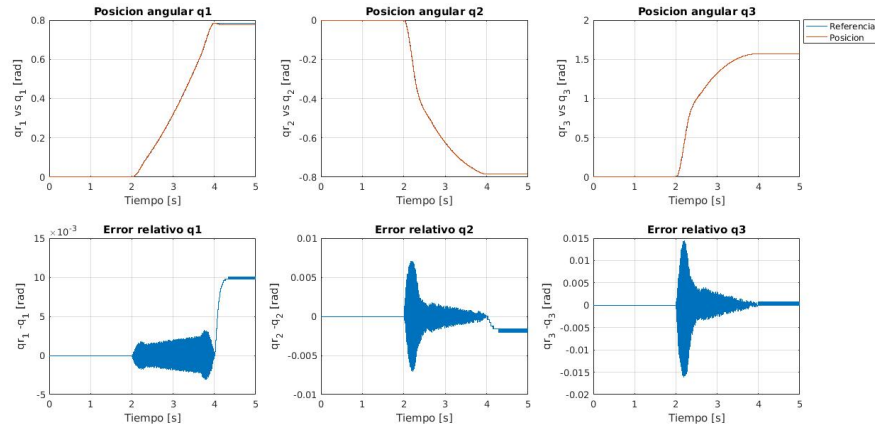


Figura 23: Seguimiento de referencia en posición de las variables articulares

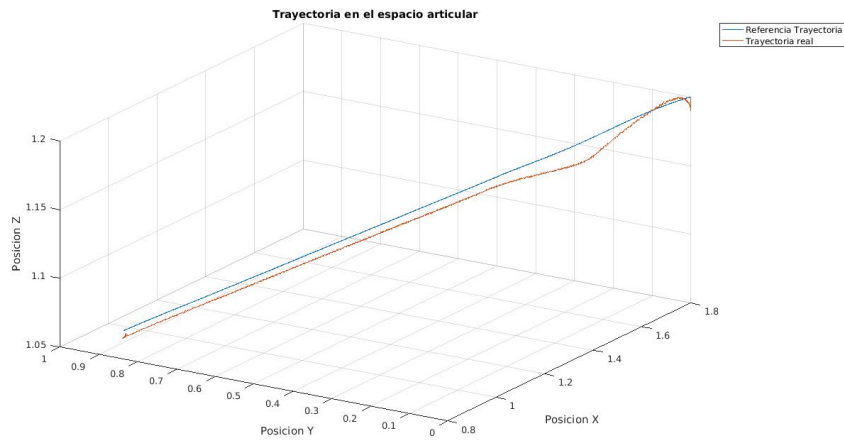


Figura 24: Seguimiento de la trayectoria en el plano XYZ

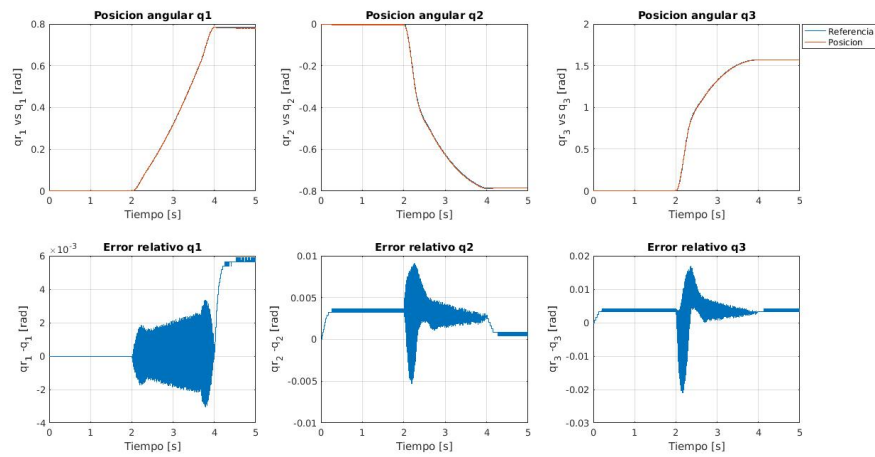


Figura 25: Seguimiento de referencia de las variables articulares

Otro buen experimento para probar éste controlador será la generación de trayectorias curvas, ya que se podrá llegar a acentuar los términos de Coirolis. Se ha optado por una trayectoria no muy rápida pero que sí recorra algo mas que media circunferencia, de tal modo que puedan llegar a obtenerse resultados concluyentes. Al igual que antes, se comparará la respuesta del robot a partir del controlador diseñado en base al modelo real con el diseñado en base al modelo ideal.

■ Robot Real con reductoras

La trayectoria a seguir por el robot en el plano XYZ se muestra a continuación:

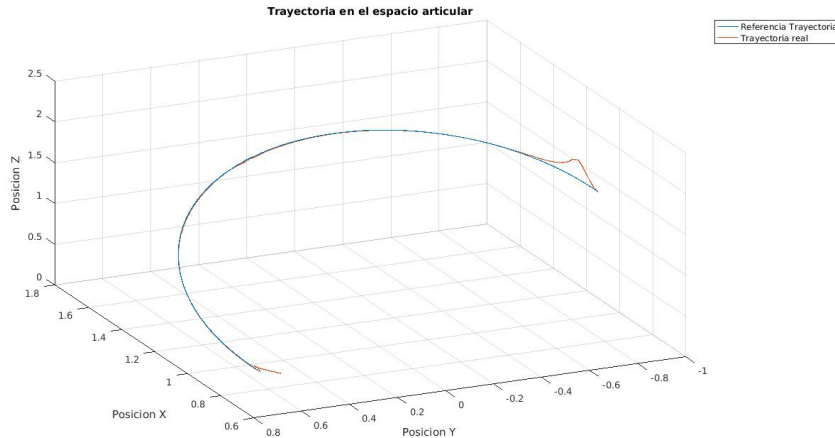


Figura 26: Referencia y seguimiento de trayectoria en el plano XYZ

Se observa que va a aparecer un error en la trayectoria, además será un error mantenido. En el caso del seguimiento de referencia en las variables articulares, el resultado obtenido es:

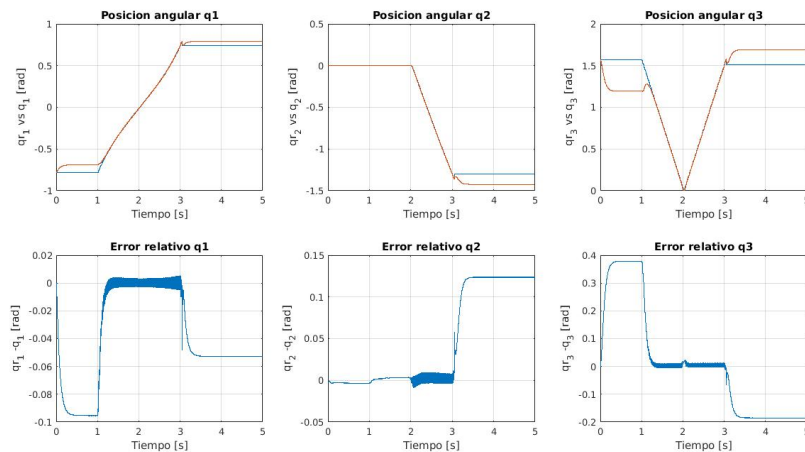


Figura 27: Seguimiento de referencia de las variables articulares

Se observa que hay un error mantenido, el cual podría llegar a no ser asumible. Por tanto, se optará por emplear el modelo ideal obtenido del robot y en aumentarle el tiempo en el que tiene que hacer la trayectoria, en post de minimizar el error del robot.

■ Robot Ideal con reductoras

En éste caso, al haber empleado más tiempo para seguir la trayectoria y el modelo ideal del robot, los errores cometidos por el robot, aunque siguen existiendo, han decrementado su magnitud cómo puede verse a continuación:

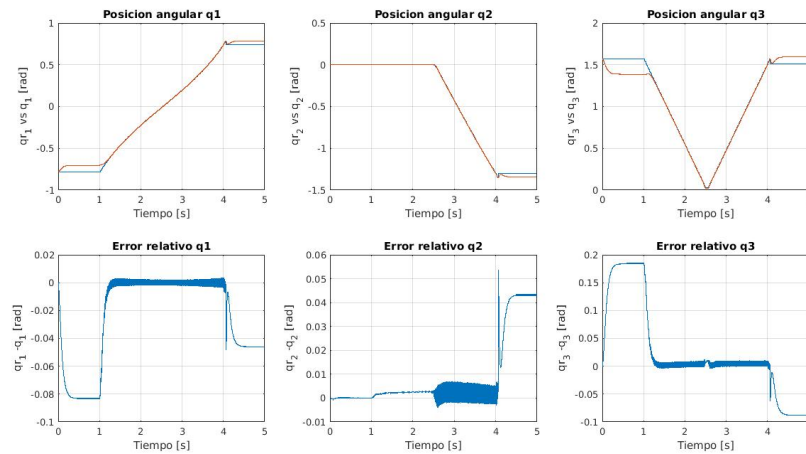


Figura 28: Seguimiento de referencia de las variables articulares

5.3.4. Comparativa del control par calculado

En el controlador mediante par calculado se ha desacoplado totalmente las articulaciones del robot, de tal modo que la dinámica del error resultante es un doble integrador.

Además de ello, se notará más éste controlador cuando se trabaje sin reductoras, ya que las reductoras son las encargadas de desacoplar las articulaciones del robot, por tanto, si no se emplean reductoras, el encargado de desacoplar las articulaciones será el propio control implementado.

Además de ello, también se buscará mostrar que, cuán más rápido se busque que sea el control, mayores serán los errores que se tendrán.

Por tanto, en primer lugar se empleará el robot ideal sin reductoras para mostrar el efecto de pedir mayor velocidad en el movimiento del robot y, posteriormente, se comparará el modelo ideal sin reductoras y el modelo real sin reductoras.

La trayectoria elegida será una trayectoria ascendente y de rotación que partirá de la posición inicial del robot.

■ Comprobación del efecto del aumento de la velocidad en la trayectoria

A continuación, en primer lugar se mostrará la comparativa del seguimiento de la trayectoria en el espacio articular XYZ en función de la velocidad pedida para que siga la misma. Además, cabe destacar que se ha implementado un control PD para la dinámica del error.

En primer lugar se mostrará la gráfica en la que se pedía que recorriese la trayectoria en 2 segundos, es decir, la que sería la trayectoria lenta:

Ahora, se mostrará el seguimiento en posiciones articulares cuando se pide que se siga la trayectoria algo más lenta:

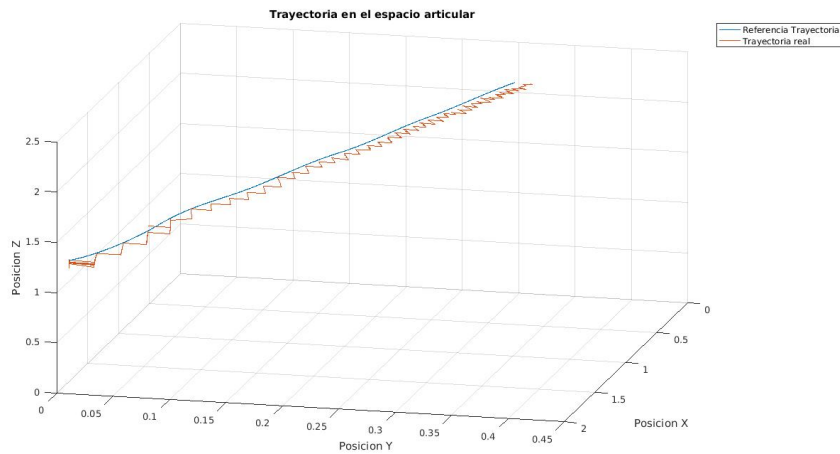


Figura 29: Seguimiento de la trayectoria lenta en el plano XYZ

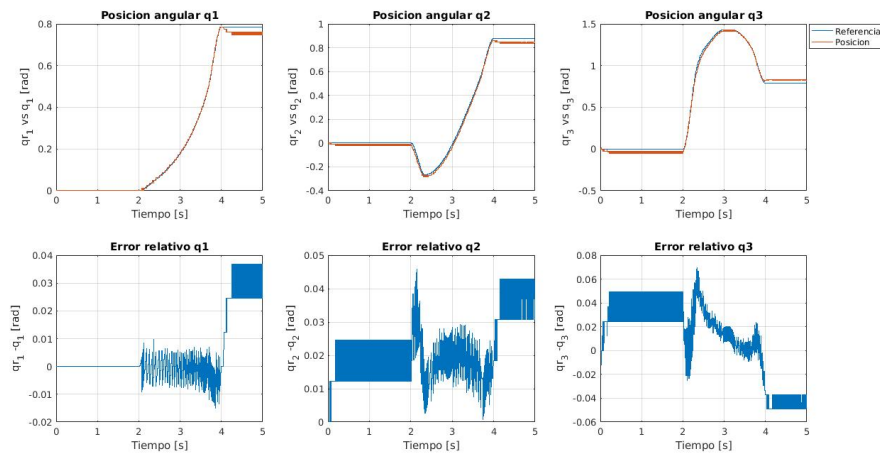


Figura 30: Seguimiento de las variables articulares

Se observa que, aunque existe un error en las tres variables, éste error es significativamente pequeño y asumible en el control del robot.

Sin embargo, en el caso de que se pida que realice la trayectoria en la mitad de tiempo, el error se incrementará significativamente cómo se muestra a continuación:

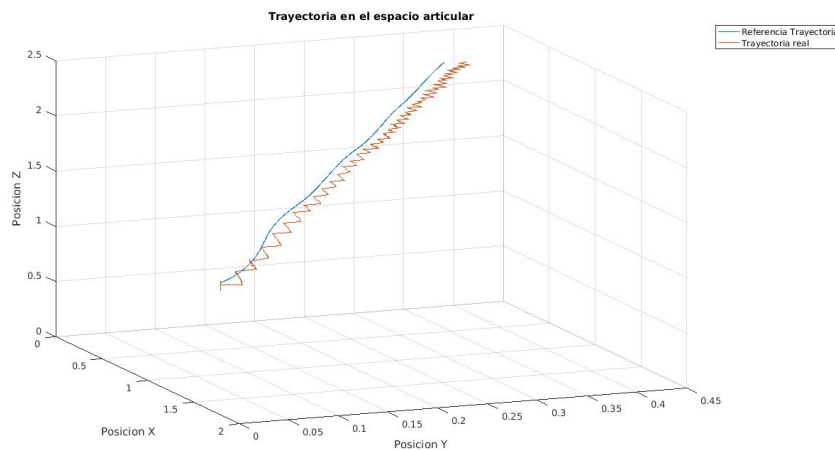


Figura 31: Seguimiento de la trayectoria rapida en el plano XYZ

Ahora, se mostrará el seguimiento en posiciones articulares cuando se pide que se siga la trayectoria en el doble de tiempo:

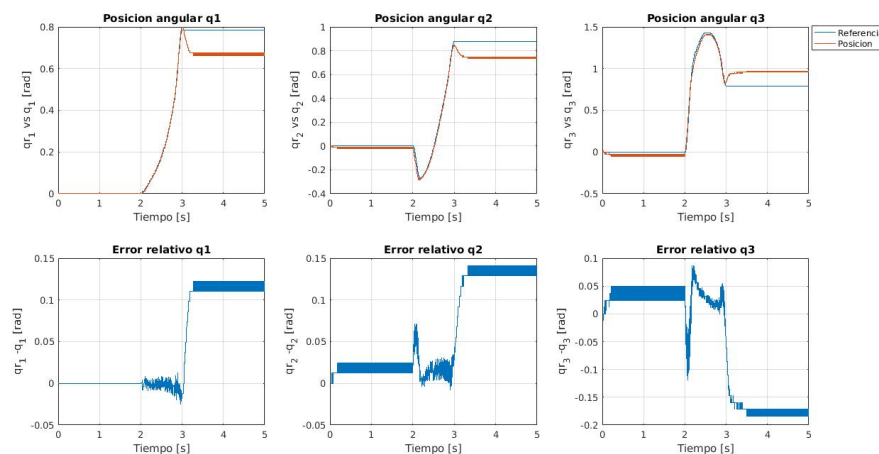


Figura 32: Seguimiento de las variables articulares

Por tanto, a modo de conclusión cabe destacar que, en el momento de pedirle al robot que siga una trayectoria será necesario hayar un compromiso entre la velocidad de la trayectoria y el error que se esté dispuesto a asumir en la misma.

■ Comparativa del modelo ideal sin reductoras obtenido y el real

Para comprobar la bondad del modelo real sin reductoras obtenido, se trazará con éste la primera trayectoria descrita en éste apartado, es decir, la que va mas despacio y de ese modo, se podrá ver de manera significativa la comparativa entre el modelo ideal sin reductoras y el modelo real sin reductoras obtenido.

Por tanto, en el espacio la trayectoria seguida empleando el control par calculado con el modelo real sin reductoras se muestra a continuación. Cabe destacar que éstas gráficas han sido obtenidas realimentando el control con la referencia de posición y de velocidad, no con las medidas tomadas del robot, ya que de ese modo se evita tener medidas tan ruidos y se puedes estimar mejor los resultados.

Posteriormente se observará cómo sería el resultado si se hubiese realimentado con las medidas

tomadas del robot.

Debido a que el modelo obtenido se degrada con el tiempo, es decir, cuanto mas tiempo de simulación pasa mayores serán los errores, hay un punto en el cual el sistema se hará inestable, ya que en éste tipo de control se depende totalmente del modelo obtenido, cómo se muestra a continuación:

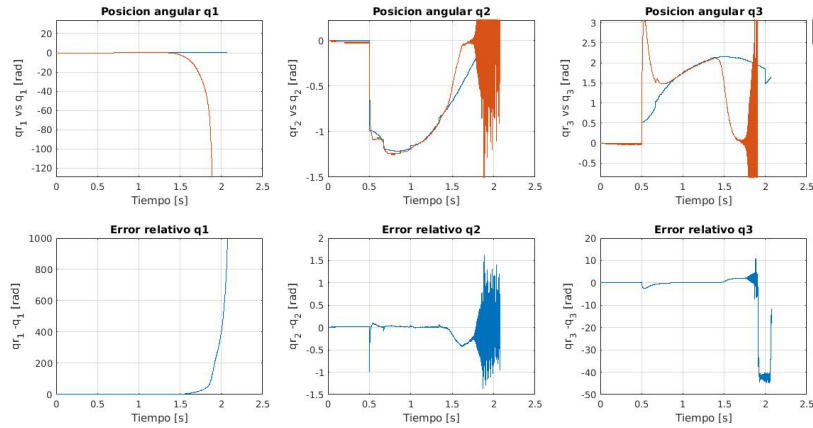


Figura 33: Seguimiento de las variables articulares

6. Anexos

6.1. Codigos de programacion

6.1.1. Implementación de controladores

La implementación de los controladores en tiempo discreto se llevo a cabo a partir de una S-Function en Matlab, es decir, un proyecto en lenguaje C compilado que simule el comportamiento de lo que sería una computadora, lo que podría ser similar a un controlador real.

Ésta funcion fue dada por el profesor *Ortega Linares, Manuel Gil* en el desarrollo de una asignatura de un año anterior y, tras analizar cuál era la mejor opcion de introducir controladores discretos para el robot, se optó por ésta.

Por tanto, los controladores PD y PID implementados, se muestran a continuación, los cuales serán programados en funciones que serán llamadas desde un programa principal llamado *Computadora.c*, la cuál no se mostrará en éste anexo debido a que no se ha considerado importante, lo que sí se mostrará será la implementación de los controladores, los cuales se realizaron de forma absoluta y la deficion de las variables de los mismos.

```

1 double tiempo; /* Para tiempo de simulacion */
2 static double ek1[3]; /* Valores anteriores del error */
3
4 /* Valores para el PD/PID absoluto */
5 static double Int_e[3], dt_e[3];
6
7 /* Valores de los parametros del PD/PID */
8 double Ti[3], Td[3], Kp[3];
9
10 /* SE DEBERA DAR VALORES A LOS PARAMETROS EN FUNCION DEL CONTROLADOR A IMPLEMENTAR.
11  * LOS VALORES DE LOS CONTROLADORES SE ENCUENTRAN EN OTRA FUNCION, SERA COPIAR-PEGAR
12  */
13 /* Parametros del PD/PID */
14 Ti[0]= ; Td[0]= ; Kp[0]= ;
15 Ti[1]= ; Td[1]= ; Kp[1]= ;
16 Ti[2]= ; Td[2]= ; Kp[2]= ;

```

Listing 1: Declaraciones.c

```

1 /* La entrada del sistema sera un vector columna de 3 componentes. */
2
3 /* Definicion de los terminos absolutos del PID */
4 int i;
5 /* -> IMPORTANTE <-
6  * SE DEBERA COMENTAR UNO DE LOS DOS CONTROLADORES CUANDO SE DESEE IMPLEMENTAR
7  */
8
9 for (i=0; i<3; i++){
10     Int_e[i] += ek[i]*Tm;
11     dt_e[i] = (ek[i] - ek1[i]) / Tm;
12 }
13
14 /* Calculo de la senal de control del PID */
15 for (i=0; i<3; i++){
16     Ireal[i] = Kp[i] * ( ek[i] + (Td[i]*dt_e[i]) + (1/Ti[i])*Int_e[i] );
17 }
18
19 /* Calculo de la senal de control del PD */
20 for (i=0; i<3; i++){
21     Ireal[i] = Kp[i] * ( ek[i] + (Td[i]*dt_e[i]) );
22 }
23
24 /* Actualizacion de las variables anteriores */
25 for (i=0; i<3; i++){
26     ek1[i] = ek[i];
27 }

```

Listing 2: Controlador.c

```
1 int i;  
2 for ( i=0; i<3; i++){  
3     ek1[i]=0.; /* Valores anteriores del error */  
4     dt_e[i]=0.;  
5     Int_e[i]=0.;  
6 }
```

Listing 3: Inicializaciones.c

6.2. Generadores de trayectorias

6.3. Estimacion de parámetros