



Universidad
Zaragoza

Proyecto Fin de Carrera

APLICACIÓN DE TÉCNICAS DE CONTROL SUBACTUADO A UN ROBOT DE ACCIONAMIENTO DIRECTO

Autor

Raúl Sampedro García

Director y ponente

Carlos Vivas Venegas

José María Martínez Montiel

Escuela de Ingeniería y Arquitectura

2013

Aplicación de técnicas de control subactuado a un robot de accionamiento directo

Resumen

El documento que se presenta a continuación expone de manera detallada el desarrollo realizado para el proyecto fin de carrera cuyo título es “Aplicación de técnicas de control subactuado a un robot de accionamiento directo”.

Dicho proyecto se ha realizado sobre una plataforma experimental, creada en la Escuela Superior de Ingeniería de la Universidad de Sevilla, la cual consta de un brazo robótico articulado de accionamiento directo con tres grados de libertad. De los tres grados de libertad nombrados tan solo dos de ellos disponían de acción de control. Esta característica pone de manifiesto que nos encontramos ante un sistema subactuado, es decir, aquél que dispone de menos acciones de control que grados de libertad. El grado de subactuación se determina hallando la diferencia entre los grados de libertad y las acciones de control, por tanto este sistema tiene un grado de subactuación de uno.

El trabajo ha desarrollado los siguientes elementos siguiendo varias etapas:

1. Tanto como para el sistema completo de tres grados de libertad como para un sistema simplificado de dos grados de libertad, se realizó el modelo matemático. El modelo simplificado también contaba con un grado de subactuación igual a uno al bloquearse la primera articulación, la cual disponía de acción de control.
2. Se ha utilizado el modelo simplificado para realizar ensayos experimentales con el fin de identificar y validar el conjunto de parámetros físicos constantes que definen al robot, así como la fricción. Este sistema simplificado con dos grados de libertad y una sola acción de control se conoce como “*Pendubot*”.
3. Una vez este modelo ha quedado completamente definido, se ha utilizado el mismo para el desarrollo de las técnicas de control subactuado. En nuestro caso, se ha elegido un control en dos fases diferenciadas. La primera se denomina *Swing Up*, (control necesario para llevar las articulaciones del robot a la posición vertical superior inestable), y balancing control (control en las proximidades de la posición vertical superior inestable). Se profundiza con más énfasis en la primera parte.
4. Se ha construido un modelo utilizando *Simulink* y se ha utilizado el entorno XPC-Target de MATLAB como interface robot-ordenador. Se han ejecutado numerosas simulaciones buscando el ajuste óptimo del controlador mediante *ensayo-error* para conseguir estabilizar el sistema en su posición superior de equilibrio inestable.
5. Por último, se han transferido todos los resultados de simulación a la plataforma experimental con el fin de comprobar las similitudes del modelo y la simulación con el robot real con el que contábamos en el laboratorio.

En la comparativa, se pone de manifiesto que el comportamiento real del robot esconde ciertos aspectos desconocidos. Se ha comprobado una significativa variabilidad en los resultados de laboratorio frente a la repetición constante de experimentos con idénticos parámetros de entrada. Con el objetivo de explicar este fenómeno se proponen varias explicaciones que precisarán de un estudio complementario para una completa comprensión de los mismos.

Índice general

Índice general	1
Índice de figuras	4
Introducción	7
1. Modelado del robot de tres grados de libertad	10
1.1. Introducción	10
1.2. Descripción del problema	10
1.3. Modelado dinámico del sistema	10
1.3.1. Matriz de inercia $D(q)$	12
1.3.2. Término de fuerzas centrífugas y de Coriolis $C(q, \dot{q})\dot{q}$	13
1.3.3. Término de fuerzas gravitatorias $G(q)$	14
1.3.4. Modelo de fricción	14
1.4. Ecuaciones del movimiento	15
1.4.1. Primera ecuación	15
1.4.2. Segunda ecuación	15
1.4.3. Tercera ecuación	15
1.4.4. Parámetros constantes	16
1.5. Consideraciones	17
1.5.1. Par cero en la muñeca, $\tau_3 = 0$	18
1.5.2. Simplificaciones en la fricción	18
2. Identificación	19
2.1. Introducción	19
2.2. Descripción del procedimiento	19
2.3. Procedimiento de identificación	21
2.3.1. Transformación matemática de las ecuaciones	21
2.3.2. Tratamiento de la fricción en el modelo	23
2.3.3. Seleccionar una señal adecuada	25
2.3.4. Realización de experimentos en bucle abierto	27
2.3.5. Filtrado y análisis de los datos	28
2.3.6. Método de ajuste: Ajuste Robusto - <i>Robust Fitting</i>	29
2.3.7. Resultados	29
2.4. Identificación con 3 constantes	31
2.4.1. Replanteamiento de las ecuaciones del sistema	31

2.4.2. Resultados	32
2.4.3. Validación de resultados	33
2.5. Resultados definitivos	35
3. Controlador	36
3.1. Introducción	36
3.2. Swing Up	37
3.2.1. Objetivo	37
3.2.2. Desarrollo teórico	37
3.2.3. Determinación de k_P , k_D y k_E	40
3.3. Ballancing Control	42
3.3.1. Objetivo	42
3.3.2. Desarrollo teórico	42
3.4. Switcher	44
3.4.1. Objetivo	44
3.4.2. Desarrollo teórico	44
4. Simulación	46
4.1. Introducción	46
4.2. Diagrama de bloques	46
4.2.1. Bloque: Controlador	47
4.2.2. Bloque: Modelo de Simulación	49
4.3. Resultados	50
4.4. Conclusión	52
5. Experimentación	53
5.1. Introducción	53
5.2. Diagrama de bloques para experimentación	53
5.3. Experimentos	54
5.4. Hipótesis	57
6. Conclusiones	58
6.1. Modelado e identificación	58
6.2. Experimentación	59
6.3. Fricción	59
6.4. Posible continuación	61
Bibliografía	62
Anexos	63
A. Descripción del sistema físico	64
A.1. Introducción	64
A.2. Robot de tres grados de libertad	64
A.2.1. Motores	64
A.2.2. Brazos de giro	64
A.2.3. Encoder incremental	70
A.3. Servoactuadores	70

A.4. Tarjetas	71
A.5. Sistema de control	71
A.5.1. PC Host	75
A.5.2. PC Target	76
A.5.3. Bus CAN	76
A.5.4. Tarjeta controladora DS1103 PPC	76
A.5.5. Software de configuración de ServoStar 600 (Drive)	77
A.6. Imágenes de los distintos elementos del sistema	78

Índice de figuras

1.1.	Robot de tres grados de libertad en movimiento	11
1.2.	Esquema del robot de tres grados de libertad	12
2.1.	Experimento considerando la fricción dinámica	24
2.2.	Bloque para el cálculo de la fricción en el codo	25
2.3.	Entrada sinusoidal	26
2.4.	Entrada sinusoidal de frecuencia linealmente creciente	27
2.5.	Entrada Gold Sequence	27
2.6.	Esquema de identificación	28
2.7.	Experimento identificación para 5 parámetros con Golden Sequence	30
2.8.	Experimento identificación para 5 parámetros con Chirp	31
2.9.	Experimento identificación para 3 parámetros con Golden Sequence	33
2.10.	Comprobación para 3 parámetros identificados con Chirp	33
2.11.	Comprobación para 3 parámetros identificados con seno de $\omega = 0,6rad/s$	34
4.1.	Esquema de simulación	46
4.2.	Interior del bloque Controlador	47
4.3.	Interior del bloque Swing Up	47
4.4.	Interior del bloque Switcher	48
4.5.	Interior del bloque Ballancing Control	48
4.6.	Interior del bloque Modelo del Robot de 3 GDL	49
4.7.	Evolución de la posición del codo con el tiempo	50
4.8.	Evolución de la posición de la muñeca con el tiempo	50
4.9.	Gráficas de la evolución del par del codo con el tiempo	51
4.10.	Evolución de las velocidades articulares con el tiempo	52
4.11.	Velocidad de la muñeca en el instante de activación del switcher	52
5.1.	Diagrama de bloques de Simulink utilizado para los experimentos	53
5.2.	Interior de los bloques controlador y robot	54
5.3.	Trayectoria del codo	54
5.4.	Trayectoria de la muñeca	55
5.5.	Evolución de las velocidades articulares	55
5.6.	Evolución del par del codo	56
5.7.	Velocidades articulares y par del codo en experimento fallido	57
5.8.	Evolución de las posiciones articulares en experimento fallido	57
A.1.	Vista frontal del robot	65

A.2. Vista del perfil del robot	66
A.3. Detalle de la sujeción al suelo	67
A.4. Detalle del motor del hombro	67
A.5. Detalle del motor del codo y del encoder incremental	68
A.6. Detalle de la tercera articulación	69
A.7. Características del enconder incremental	70
A.8. Kollmorgen Seidel Servostar 600	72
A.9. Componentes del sistema del servoactuador	73
A.10. Diagrama de conexiones del ServoStar 600	74
A.11. Esquema del sistema de control	75
A.12. Software Drive: menú para elegir el puerto	77
A.13. Software Drive: menú para manipular cada articulación	78
A.14. PC Target e interfaz gráfica	78
A.15. Servoactuadores y cuadro eléctrico	79
A.16. Cierre del cuadro eléctrico	80

Introducción

En el presente proyecto: se trabajará con un robot de accionamiento directo de tres grados de libertad que consta de un brazo articulado de más de metro y medio de longitud que se compone de tres articulaciones diferentes. Por similitud con un brazo humano y sus articulaciones se han denominado: *hombro, codo y muñeca*. La peculiaridad de este sistema es que, pese a que dispone de tres grados de libertad, solo existen dos acciones de control. Es por ello que nos hallamos ante un sistema subactuado.

En esta plataforma experimental, solamente dos de las tres articulaciones pueden ser controladas de manera directa, para la tercera se deberá desarrollar alguna estrategia específica. Las dos articulaciones que pueden ser controladas directamente son el hombro y el codo. La muñeca queda libre y se moverá simplemente porque su movimiento está íntimamente relacionado con el movimiento de las otras dos articulaciones.

El equipo experimental se componía del robot de accionamiento directo con sus servoactuadores y dos ordenadores conectados entre sí. En uno de los dos ordenadores se utilizaba un sistema operativo en tiempo real, todo el entorno XPCTarget que era el que realmente le enviaba las señales de control al robot y monitorizaba las variables. El otro ordenador estaba configurado con Windows XP y Simulink de MATLAB donde se construían los modelos de bloques de simulación y experimentación y se lanzaban al otro ordenador.

El trabajo se desarrolló en su totalidad en la Escuela Superior de Ingeniería de la Universidad de Sevilla bajo la tutela de Carlos Vivas Venegas, Profesor del Departamento de Ingeniería de Sistemas. El robot en sí mismo había sido utilizado para otro proyecto anterior a este, [BP09] realizado por Guillermo Bejarano el cual fue de gran ayuda para conseguir entender el funcionamiento del robot así como el software necesario para su utilización. En [BP09] se desarrolló el hardware de control del sistema, así como los modelos matemáticos necesarios para su control empleando la plataforma Simulink de MATLAB. Desde el punto de vista de control, se diseñó un controlador PID para minimizar el tiempo que tardaban el hombro y el codo, las dos articulaciones que poseen acción de control, en llegar a una determinada posición arbitraria. En cambio en el caso que nosotros abordamos solo tendremos una entrada de menos para conseguir las posiciones de referencia.

Para alcanzar nuestro objetivo, se partió del modelado matemático del sistema utilizando como base las ecuaciones de Lagrange. Se realizó un modelo global que incluía los tres grados de libertad y, posteriormente, se simplificó para utilizar únicamente un modelo de dos grados de libertad que solo tenía en cuenta el codo y la muñeca. Ésta última es la que no dispone de motor alguno. Este sistema “*simplificado*” de dos grados de libertad, con

tan solo uno de los dos actuado, se conoce en la literatura de la materia como *Pendubot*. Hay fundamentalmente dos diferentes. Existe uno cuya acción es lineal, habitualmente es un carro que se mueve adelante y atrás sobre unos railes, y otro tipo diferente cuya acción es rotatoria y se le aplica un par como entrada de control.

Una vez halladas las ecuaciones que rigen el sistema halladas todavía es necesario determinar ciertos parámetros constantes que incluye el modelo. Estos parámetros son productos de las masas, las longitudes y las inercias de las diferentes partes del brazo robótico. Esos datos, a priori conocidos, en realidad son desconocidos. Se debe a que las unidades de entrada al robot son voltios independientemente de que se trate de longitud, masa o inercia. Como la relación de conversión entre voltios y el resto de unidades es desconocida, se han de realizar experimentos de identificación para obtener los valores de las constantes en conjuntos de unidades relacionadas con los voltios.

Una vez obtenido un modelo completamente determinado, se hacen varios experimentos para comprobar su validez. Se compara el comportamiento que muestra el robot real con el que predice nuestro modelo en simulación, utilizando la misma entrada en la plataforma real y en Simulink. Se acepta el modelo conseguido porque las respuestas tienen una similitud considerable. A sabiendas que será prácticamente imposible llegar a una predicción exacta de los comportamientos reales.

A partir de aquí se diseñará una estrategia de control para la estabilización en un punto de equilibrio inestable. Se elige una técnica que se divide en dos etapas. La primera, *Swing Up*, consiste en levantar el codo “arrastrando” la muñeca hasta un entorno suficientemente cercano a la posición vertical inestable. Esto se consigue mediante una aproximación de la energía y su convergencia hacia la energía en el equilibrio. La segunda parte del controlador, el *Balancing Control*, trata de mantener el Pendubot en su lugar de equilibrio inestable. Funciona de manera correcta en unas *cuencas de estabilidad*, cuando el sistema se sale de ellas el controlador ya no lo puede recuperar.

Antes de ponerlo en funcionamiento en la plataforma real, se prueba que el controlador al menos trabaja como se espera en simulación. Se monta un diagrama de bloques que representa al robot y se ajustan los parámetros del controlador para intentar que consiga su objetivo. Mediante *ensayo-error* se van afinando los valores hasta que finalmente el robot alcanza el punto de equilibrio perseguido. En este punto, se añade un *switcher* para procurar que la transición entre controladores sea lo más suave posible, ya que el par no es continuo. Con esto, nos referimos a que en el momento del cambio, las dos fases del controlador calculan un par diferente y ocurre un salto repentino que puede sacar el sistema de su trayectoria.

A la hora de llevar todo lo obtenido anteriormente a la plataforma real se objetiva que la respuesta no es la esperada. El robot actúa de manera diferente a lo que habíamos estimado en un principio. Se observa una falta de repetibilidad y constancia en los resultados. El par de entrada al sistema es menor de lo que cabía esperar. Al ajustarse los valores manualmente se aprecia un comportamiento del sistema que podría definirse como caótico puesto que con las mismas condiciones iniciales y las mismas entradas llega casi al equilibrio y en

el siguiente intento incluso se inestabiliza.

Tras extensas baterías de experimentos, barriendo un conjunto muy amplio de parámetros se observa de forma repetida la gran variabilidad de respuestas en el sistema. Se objetiva también una diferencia a lo largo del día ya que los valores del controlador funcionan de manera adecuada en los primeros compases de la mañana mientras que el comportamiento del robot empeora según avanzan las horas y se acerca el mediodía del verano.

La hipótesis que se baraja es que la pared a la cual está anclado el robot recibe directamente la radiación solar, pudiendo alcanzar una diferencia de temperatura de 20°C del inicio al final de la mañana. Este importante incremento de temperatura hace que la fricción de las articulaciones se vea modificada de manera sustancial. Incluso el mismo calor propio generado por la fricción de las articulaciones hace que el robot se comporte de manera diferente después de haber estado funcionando durante un tiempo. Hasta tal punto ocurre esto que hace que el modelo matemático identificado no sea completamente válido.

Creemos que nuevos experimentos deberían continuarse en esta línea para corroborar dicha hipótesis y, en caso de ser confirmada, controlar la variable de temperatura teniéndola profundamente en cuenta a la hora de realizar un nuevo modelo de fricción.

Otra posible línea de trabajo sería buscar unos valores, al comienzo de la mañana, que consigan que el Pendubot se estabilice en el punto más alto y, posteriormente, se podría intentar el mismo experimento incluyendo el tercer grado de libertad del robot. Quizás se podría proponer hacer algún tipo de preludio de experimento que se ejecute cada vez que se vaya a usar el sistema y que trate de identificar los parámetros. De esta manera se tendrá un sistema identificado diferente cada vez que se ponga en marcha.

Capítulo 1

Modelado del robot de tres grados de libertad

1.1. Introducción

En el presente documento se pretende explicar el modelo que se ha seleccionado para el robot de tres grados de libertad de accionamiento directo, las razones por las que se ha elegido y las implicaciones que tiene la elección del modelo sobre el proceso de identificación.

1.2. Descripción del problema

Para realizar un control eficiente basado en el modelo sobre el robot de tres grados de libertad y, en general, sobre cualquier sistema real, es necesario identificar adecuadamente las características dinámicas y de fricción del mismo. En este caso la identificación partirá del modelado dinámico del robot y de la elección de un modelo de fricción para sus articulaciones.

En la Figura 1.1 se muestra una fotografía del robot de tres grados de libertad en movimiento.

En la Figura 1.2 se muestra un esquema de la estructura del robot, que resultará útil para la comprensión del modelo dinámico y de los parámetros involucrados en el mismo.

1.3. Modelado dinámico del sistema

Con el fin de determinar las ecuaciones de movimiento que rigen el sistema se emplea la metodología de Lagrange planteada en [SHV04] y en [JB91]. Esta formulación se basa en el balance energético del sistema. Para ello, se define la lagrangiana, \mathcal{L} , del sistema mecánico como:

$$\mathcal{L} = E_c - E_p = \mathcal{T} - \mathcal{V} \quad (1.1)$$

siendo E_c la energía cinética y E_p la energía potencial del sistema completo. En donde se llama q_i a las coordenadas generalizadas (en este caso los ángulos de giro del hombro, del



Figura 1.1: Robot de tres grados de libertad en movimiento

culo y de la muñeca que se tomaran positivos en sentido antihorario y negativos en sentido contrario), la fuerza generalizada (en este caso, por τ_i) aplicada sobre la articulación “ i ” viene determinada por la siguiente ecuación:

$$\tau_i = \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{T}}{\partial q_i}, \quad i = 1, 2, 3 \quad (1.2)$$

En forma matricial se tiene que:

$$D(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + G(q) + F_f(q, \dot{q}) = \tau(t) \quad (1.3)$$

$$\begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} + \begin{bmatrix} F_{f1} \\ F_{f2} \\ F_{f3} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (1.4)$$

Donde se distinguen claramente un término de inercia (matriz $D(q)$), un término de fuerzas centrífugas y de Coriolis (matriz $C(q, \dot{q}) \cdot \dot{q}$), un término de fuerzas gravitatorias (vector $G(q)$) y un término de fricción (vector $F_f(q, \dot{q})$). Esta es la descomposición clásica de la ecuación dinámica del movimiento de un robot.

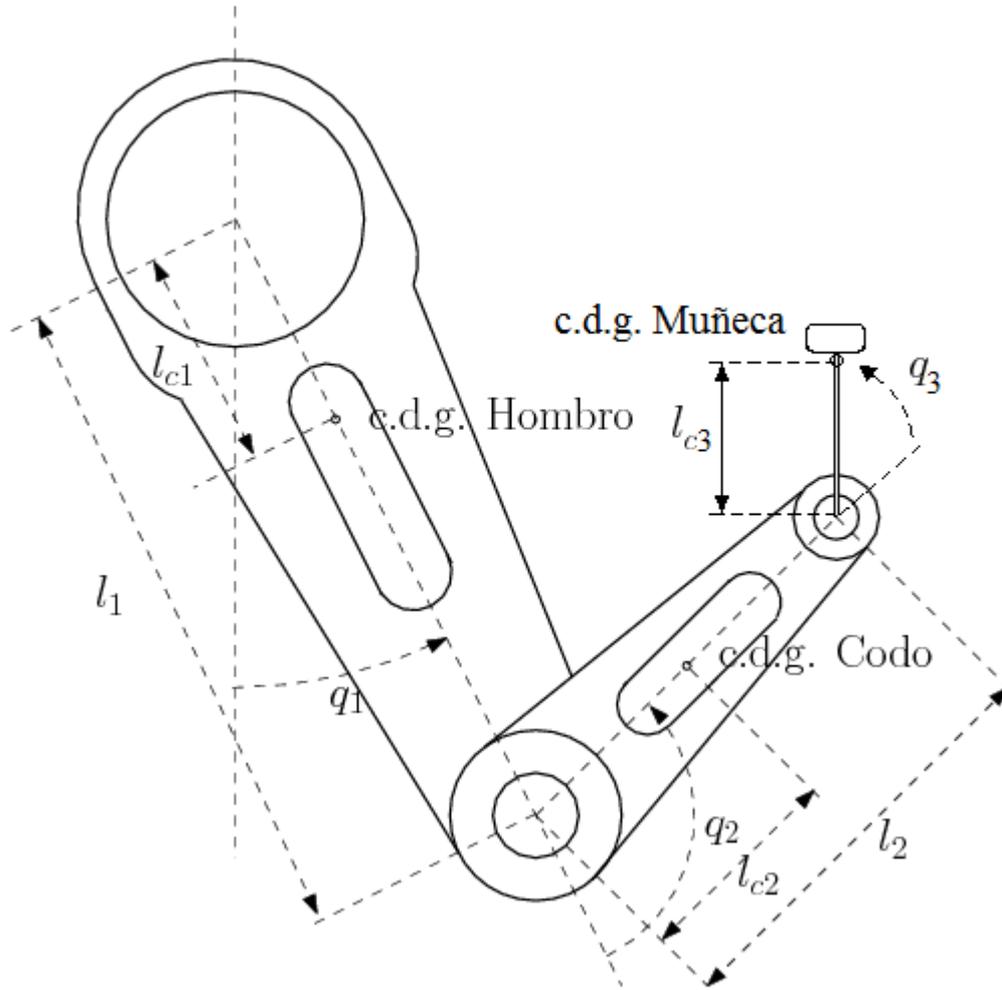


Figura 1.2: Esquema del robot de tres grados de libertad

1.3.1. Matriz de inercia $D(q)$

Para obtener la matriz $D(q)$ primero se ha calculado la matriz jacobiana de cada uno de los vectores de posición que van desde el punto fijo tomado en el centro del motor de la articulación del hombro hasta los centros de gravedad del hombro, del codo y de la muñeca, como se ve en las Ecuaciones 1.5 y 1.6.

$$OC_{G1} = \begin{pmatrix} L_{C1} \cos(q_1) \\ L_{C1} \sin(q_1) \\ 0 \end{pmatrix}, \quad OC_{G2} = \begin{pmatrix} L_1 \cos(q_1) + L_{C2} \cos(q_1 + q_2) \\ L_1 \sin(q_1) + L_{C2} \sin(q_1 + q_2) \\ 0 \end{pmatrix} \quad (1.5)$$

$$OC_{G3} = \begin{pmatrix} L_1 \cos(q_1) + L_2 \cos(q_1 + q_2) + L_{C3} \cos(q_1 + q_2 + q_3) \\ L_1 \sin(q_1) + L_2 \sin(q_1 + q_2) + L_{C3} \sin(q_1 + q_2 + q_3) \\ 0 \end{pmatrix} \quad (1.6)$$

$$J_{v_{ci}} = \begin{pmatrix} \left. \frac{\partial \vec{OC}_{Gi}}{\partial q_1} \right|_x & \left. \frac{\partial \vec{OC}_{Gi}}{\partial q_2} \right|_x & \left. \frac{\partial \vec{OC}_{Gi}}{\partial q_3} \right|_x \\ \left. \frac{\partial \vec{OC}_{Gi}}{\partial q_1} \right|_y & \left. \frac{\partial \vec{OC}_{Gi}}{\partial q_2} \right|_y & \left. \frac{\partial \vec{OC}_{Gi}}{\partial q_3} \right|_y \\ \left. \frac{\partial \vec{OC}_{Gi}}{\partial q_1} \right|_z & \left. \frac{\partial \vec{OC}_{Gi}}{\partial q_2} \right|_z & \left. \frac{\partial \vec{OC}_{Gi}}{\partial q_3} \right|_z \end{pmatrix} \quad (1.7)$$

Una vez obtenidas las matrices jacobianas, a partir de la Ecuación 1.7, para el hombro, el codo y la muñeca se puede obtener directamente la matriz $D(q)$ sabiendo que:

$$D(q) = m_1 J_{v_{c1}}^T J_{v_{c1}} + m_2 J_{v_{c2}}^T J_{v_{c2}} + m_3 J_{v_{c3}}^T J_{v_{c3}} + I_z$$

Donde la matriz I_z es la matriz de inercia del sistema completo. Puesto que tratamos con un robot que solamente actúa en un plano que nosotros consideraremos como el plano XY la única componente de inercia que aparecerá será la del eje z . La matriz de inercia, I_z , para el sistema completo se obtiene mediante la suma de las matrices de inercia de cada articulación por separado. Así, la matriz $D(q)$ resulta:

$$\begin{aligned} d_{11} &= k_1 + 2k_6 + k_8 + 2k_2 \cos(q_2) + 2k_6 \cos(q_2) + 2k_7 \cos(q_2 + q_3) + 2k_7 \cos(q_3) \\ d_{12} &= d_{21} = k_3 + k_6 + k_8 + k_2 \cos(q_2) + k_6 \cos(q_2) + k_7 \cos(q_2 + q_3) + 2k_7 \cos(q_3) \\ d_{13} &= d_{31} = k_8 + k_7 \cos(q_2 + q_3) + k_7 \cos(q_3) \\ d_{22} &= k_3 + k_6 + k_8 + 2k_7 \cos(q_3) \\ d_{23} &= d_{32} = k_8 + k_7 \cos(q_3) \\ d_{33} &= k_8; \end{aligned} \quad (1.8)$$

Por simplificar las expresiones se utilizan las constantes, k_1 a k_{10} , que se detallan posteriormente en la Tabla 1.1.

1.3.2. Término de fuerzas centrífugas y de Coriolis $C(q, \dot{q})\dot{q}$

La matriz $C(q, \dot{q})$ se calcula a partir de la matriz $D(q)$ con la expresión 1.10, que tenemos a continuación. Se utilizan los símbolos de Christoffel que reducen el esfuerzo de cálculo en aproximadamente un 50% introduciendo que $c_{ijk} = c_{jik}$.

$$c_{kj} = \sum_{i=1}^n c_{ijk}(\mathbf{q}) \dot{q}_i \quad (1.9)$$

$$c_{ijk} = \sum_{i=1}^n \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \dot{q}_i \quad (1.10)$$

Se obtiene cada uno de los elementos de la matriz $C(q, \dot{q})$:

$$\begin{aligned}
 c_{11} &= -\dot{q}_2(k_7 \sin(q_2 + q_3) + (k_2 + k_6) \sin(q_2)) - \dot{q}_3(k_7 \sin(q_2 + q_3) + k_7 \sin(q_3)) \\
 c_{12} &= -\dot{q}_1(k_7 \sin(q_2 + q_3) + (k_2 + k_6) \sin(q_2)) - \dot{q}_2(k_7 \sin(q_2 + q_3) + (k_2 + k_6) \sin(q_2)) \\
 &\quad - \dot{q}_3(k_7 \sin(q_2 + q_3) + k_7 \sin(q_3)) \\
 c_{13} &= -\dot{q}_1(k_7 \sin(q_2 + q_3) + k_7 \sin(q_3)) - \dot{q}_2(k_7 \sin(q_2 + q_3) + k_7 \sin(q_3)) \\
 &\quad - \dot{q}_3(k_7 \sin(q_2 + q_3) + k_7 \sin(q_3)) \\
 c_{21} &= \dot{q}_1(k_7 \sin(q_2 + q_3) + (k_2 + k_6) \sin(q_2)) - k_7 \dot{q}_3 \sin(q_3) \\
 c_{22} &= -k_7 \dot{q}_3 \sin(q_3) \\
 c_{23} &= -k_7 \dot{q}_1 \sin(q_3) - k_7 \dot{q}_2 \sin(q_3) - k_7 \dot{q}_3 \sin(q_3) \\
 c_{31} &= \dot{q}_1(k_7 \sin(q_2 + q_3) + k_7 \sin(q_3)) + k_7 \dot{q}_2 \sin(q_3) \\
 c_{32} &= k_7 \dot{q}_1 \sin(q_3) + k_7 \dot{q}_2 \sin(q_3) \\
 c_{33} &= 0
 \end{aligned} \tag{1.11}$$

Por simplificar las expresiones se utilizan las constantes, k_1 a k_{10} , que se detallan posteriormente en la Tabla 1.1.

1.3.3. Término de fuerzas gravitatorias $G(q)$

La componente de la gravedad para cada una de las tres ecuaciones de movimiento es:

$$\begin{aligned}
 G_1 &= (k_4 + k_9) \sin(q_1) + (k_5 + k_9) \sin(q_1 + q_2) + k_{10} \sin(q_1 + q_2 + q_3) \\
 G_2 &= (k_5 + k_9) \sin(q_1 + q_2) + k_{10} \sin(q_1 + q_2 + q_3) \\
 G_3 &= k_{10} \sin(q_1 + q_2 + q_3)
 \end{aligned} \tag{1.12}$$

Por simplificar las expresiones se utilizan las constantes, k_1 a k_{10} , que se detallan posteriormente en la Tabla 1.1.

1.3.4. Modelo de fricción

El modelo de fricción elegido se ha simplificado respecto al modelo de Lu-Gre que se eligió para tratar el robot con dos grados de libertad. Para el objetivo que perseguimos en este trabajo hemos considerado la reducción de los efectos de la fricción a un solo término de fricción dinámica que queda multiplicado por la velocidad articular correspondiente. Por tanto:

$$\begin{aligned}
 F_{f_{hombro}} &= F_{f1} = \sigma_{2h} = 0,053 \text{ V seg/rad} \\
 F_{f_{codo}} &= F_{f2} = \sigma_{2c} = 0,0581 \text{ V seg/rad} \\
 F_{f_{muñeca}} &= F_{f3} \approx 0
 \end{aligned} \tag{1.13}$$

Se han utilizado los valores de la fricción dinámica de hombro y codo de [BP09], proyecto que trabajó anteriormente con el mismo robot pero un grado menos de libertad, es decir, sin la muñeca pero con dos acciones de control.

Es importante remarcar que al aparecer la muñeca sin ninguna acción directa sobre ella se puede despreciar la fricción debido a que al no disponer de motor el rozamiento queda

sumamente disminuído. Se comprueba que el tiempo de oscilación libre de la muñeca es mucho mayor que el de la oscilación libre del codo, como si fueran pendulos. De aquí se desprende que la fricción es mucho menor en la muñeca.

1.4. Ecuaciones del movimiento

1.4.1. Primera ecuación

$$\begin{aligned}
 & [k_1 + 2k_6 + k_8 + 2k_7 \cos(q_2 + q_3) + 2k_6 \cos(q_2) + 2k_2 \cos(q_2) + 2k_7 \cos(q_3)] \cdot \ddot{q}_1 + \\
 & [k_3 + k_6 + k_8 + k_7 \cos(q_2 + q_3) + k_6 \cos(q_2) + k_2 \cos(q_2) + 2k_7 \cos(q_3)] \cdot \ddot{q}_2 + \\
 & [k_8 + k_7 \cos(q_2 + q_3) + k_7 \cos(q_3)] \cdot \ddot{q}_3 - \\
 & [k_6 \sin(q_2) + k_2 \sin(q_2) + k_7 \sin(q_2 + q_3)] \cdot \dot{q}_2^2 - \\
 & [k_7 \sin(q_3) + k_7 \sin(q_2 + q_3)] \cdot \dot{q}_3^2 - \\
 & [2k_6 \sin(q_2) + 2k_2 \sin(q_2) + 2k_7 \sin(q_2 + q_3)] \cdot \dot{q}_1 \dot{q}_2 - \\
 & [2k_7 \sin(q_3) + 2k_7 \sin(q_2 + q_3)] \cdot \dot{q}_1 \dot{q}_3 - \\
 & [2k_7 \sin(q_3) + 2k_7 \sin(q_2 + q_3)] \cdot \dot{q}_2 \dot{q}_3 + \\
 & (k_4 + k_9) \sin(q_1) + (k_5 + k_9) \sin(q_1 + q_2) + k_{10} \sin(q_1 + q_2 + q_3) + F_{f1} = \tau_1 \quad (1.14)
 \end{aligned}$$

1.4.2. Segunda ecuación

$$\begin{aligned}
 & [k_3 + k_6 + k_8 + k_7 \cos(q_2 + q_3) + k_2 \cos(q_2) + k_6 \cos(q_2) + 2k_7 \cos(q_3)] \cdot \ddot{q}_1 + \\
 & [k_3 + k_6 + k_8 + 2k_7 \cos(q_3)] \cdot \ddot{q}_2 + \\
 & [k_8 + k_7 \cos(q_3)] \cdot \ddot{q}_3 + \\
 & [k_6 \sin(q_2) + k_7 \sin(q_2 + q_3) + k_2 \sin(q_2)] \cdot \dot{q}_1^2 - \\
 & [k_7 \sin(q_3)] \cdot \dot{q}_3^2 - \\
 & [2k_7 \sin(q_3)] \cdot \dot{q}_1 \dot{q}_3 - \\
 & [2k_7 \sin(q_3)] \cdot \dot{q}_2 \dot{q}_3 + \\
 & (k_5 + k_9) \sin(q_1 + q_2) + k_{10} \sin(q_1 + q_2 + q_3) + F_{f2} = \tau_2 \quad (1.15)
 \end{aligned}$$

1.4.3. Tercera ecuación

$$\begin{aligned}
 & [k_8 + k_7 \cos(q_2 + q_3) + k_7 \cos(q_3)] \cdot \ddot{q}_1 + \\
 & [k_8 + k_7 \cos(q_3)] \cdot \ddot{q}_2 + \\
 & [k_8] \cdot \ddot{q}_3 + \\
 & [k_7 \sin(q_3) + k_7 \sin(q_2 + q_3)] \cdot \dot{q}_1^2 + \\
 & [k_7 \sin(q_3)] \cdot \dot{q}_2^2 + \\
 & [2k_7 \sin(q_3)] \cdot \dot{q}_1 \dot{q}_2 + \\
 & k_{10} \sin(q_1 + q_2 + q_3) + F_{f3} = 0 \quad (1.16)
 \end{aligned}$$

$\mathbf{k}_1 = m_1 L_{c1}^2 + I_1 + m_2 L^2 + m_2 L_{c2}^2 + I_2$	$\mathbf{k}_6 = L^2 m_3$
$\mathbf{k}_2 = m_2 L_{c2} L$	$\mathbf{k}_7 = L_{c3} L m_3$
$\mathbf{k}_3 = m_2 L_{c2}^2 + I_2$	$\mathbf{k}_8 = L_{c3}^2 m_3 + I_3$
$\mathbf{k}_4 = m_1 L_{c1} g + m_2 L g$	$\mathbf{k}_9 = L m_3 g$
$\mathbf{k}_5 = m_2 L_{c2} g$	$\mathbf{k}_{10} = L_{c3} m_3 g$

Tabla 1.1: Definición de las constantes empleadas en el modelo del robot

1.4.4. Parámetros constantes

Como se puede comprobar, la parte dinámica del modelo depende de catorce parámetros agregados, k'_1 a k'_{14} , cuyas expresiones en función de los parámetros físicos del robot y de los actuadores son:

$$\begin{aligned}
 k'_1 &= m_1 L_{c1}^2 + I_1 + m_2 L_1^2 + m_2 L_{c2}^2 + I_2 & k'_8 &= L_2 L_{c3} m_3 \\
 k'_2 &= m_2 L_1 L_{c2} & k'_9 &= L_{c3}^2 m_3 + I_3 \\
 k'_3 &= m_2 L_{c2}^2 + I_2 & k'_{10} &= (L_1^2 + L_2^2 + L_{c3}^2) m_3 + I_3 \\
 k'_4 &= m_1 L_{c1} g + m_2 L_1 g & k'_{11} &= L_2^2 m_3 \\
 k'_5 &= m_2 L_{c2} g & k'_{12} &= m_3 L_1 g \\
 k'_6 &= L_1 L_2 m_3 & k'_{13} &= m_3 L_2 g \\
 k'_7 &= L_1 L_{c3} m_3 & k'_{14} &= m_3 L_{c3} g
 \end{aligned} \tag{1.17}$$

Por suerte en el caso que aquí nos compete se cumple que la longitud de la articulación *hombro* es igual a la longitud de la articulación *codo*:

$$L_1 = L_2 = L$$

lo que implica que el número de parámetros constantes que necesitamos para definir nuestro sistema y que, posteriormente, tendremos que identificar se ve reducido a 10 parámetros. Éstos aparecen reflejados en la Tabla 1.1.

El significado físico de cada uno de los sumandos de los parámetros constantes aparece en la Tabla 1.2.

Parámetro	Significado
m_1	Masa de la articulación 1, hombro
m_2	Masa de la articulación 2, codo
m_3	Masa de la articulación 3, muñeca
L_1	Longitud de la articulación 1, hombro
L_2	Longitud de la articulación 2, codo
L_{c1}	Longitud desde el eje hasta el centro de gravedad de la articulación 1
L_{c2}	Longitud desde el eje hasta el centro de gravedad de la articulación 2
L_{c3}	Longitud desde el eje hasta el centro de gravedad de la articulación 3
L	$L_1 = L_2 = L$, Longitud del hombro = Longitud del codo
I_1	Inercia respecto al eje de giro de la articulación 1
I_2	Inercia respecto al eje de giro de la articulación 2
I_3	Inercia respecto al eje de giro de la articulación 3
g	Aceleración de la gravedad
F_{f1}	Fuerza de fricción en la articulación 1, hombro
F_{f2}	Fuerza de fricción en la articulación 2, codo
F_{f3}	Fuerza de fricción en la articulación 3, muñeca

Tabla 1.2: Descripción del significado de los parámetros del robot

1.5. Consideraciones

En los siguientes apartados se comentarán los detalles de gran relevancia para lo que resta de proyecto.

1.5.1. Par cero en la muñeca, $\tau_3 = 0$

Es muy importante recalcar la importancia de esta propiedad debido a que es la cuestión fundamental a la que nos enfrentamos a lo largo de todo este desarrollo.

Una característica de este tipo implica que no habrá ninguna acción directa sobre la última articulación, la *muñeca*. Todo el control que, a la postre, se desarrollará se basa en las acciones de los motores en las otras dos articulaciones para conseguir estabilizar la muñeca entorno a una posición de equilibrio inestable, como si de un péndulo invertido se tratara.

Es por esto que se utilizarán técnicas de control subactuado al disponer de una acción externa menos que grados de libertad tiene el sistema, es decir, dos acciones externas (pares sobre los motores del hombro y del codo) para controlar tres grados de libertad (ángulos de giro del hombro, del codo y de la muñeca).

1.5.2. Simplificaciones en la fricción

Se comprueba que la fuerza de fricción en la articulación *muñeca* en comparación con el *hombro* y el *codo* es prácticamente despreciable, por tanto se tomará $F_{f3} \approx 0$.

Realizando un sencillo experimento de caída libre para las articulaciones del *codo* y de la *muñeca* por separado se puede apreciar a simple vista que el tiempo que le cuesta a cada una detenerse en reposo en la posición más baja es mucho mayor para la muñeca que para el codo.

Esta diferencia de tiempo se debe a que la muñeca es la parte del brazo que no tiene ningún motor “reteniendo” ni actuando sobre ella por lo que la energía disipada por ciclo sea, probablemente, mucho menor al de las otras dos articulaciones.

Capítulo 2

Identificación

2.1. Introducción

Para realizar un control eficiente sobre el robot de tres grados de libertad, y en general sobre cualquier sistema real, es necesario identificar adecuadamente las características dinámicas y de fricción del mismo. En el presente documento se pretende describir el procedimiento llevado a cabo para realizar la identificación del sistema y la estimación de los parámetros del modelo adoptado para el robot.

Si bien es de gran importancia obtener un modelo con la máxima precisión posible cabe destacar que no siempre se puede conseguir unos parámetros ajustados al máximo ya sea por la precisión de los instrumentos, por la multitud de factores que aportan una incertidumbre extra en el desempeño de la tarea o por otros muchos condicionantes externos.

En el desarrollo de este trabajo la preocupación principal ha optado por enfocar sus esfuerzos en la robustez que debe tener el sistema de control para lidiar con todos estos aspectos desconocidos “*a priori*” ya que en el transcurso natural de la operación de control se debe compensar cualquier tipo de perturbación que no existiera en el modelo planteado.

2.2. Descripción del procedimiento

Para el propósito de este proyecto se ha tomado el primer grupo de constantes, k_1 a k_5 , que sólo involucran a las articulaciones hombro y codo, como conocidas. El valor de estos parámetros ha sido extraído del anterior trabajo con la misma plataforma experimental pero sin añadir la tercera articulación, la *muñeca*, que no dispone de motor y, por tanto, carece de acción de control. Se pueden comprobar los valores exactos en [BP09].

El procedimiento propuesto es una metodología de identificación total, de modo que en cada experimento realizado se identifica la totalidad de parámetros del modelo. Para ello, por simplicidad, se bloquea el hombro, que es la articulación más grande, y se trabaja únicamente con el codo y la muñeca. Puesto que los experimentos de identificación se realizan en lazo abierto es importante no entregar demasiada energía al sistema con la intención de que no llegue a inestabilizarse. Debido a la diferencia de masas entre la muñeca y las otras dos articulaciones no es necesaria una velocidad elevada para conseguirlo. Es por

esto que se deben inducir trayectorias de características apropiadas que resalten el modo de comportamiento de la muñeca bajo diversos pares de entrada al codo.

Para cada una de estas trayectorias se propone un algoritmo de análisis y filtrado de datos que posteriormente es alimentado a un algoritmo clásico de estimación mínima cuadrática, empleando para ello el regresor apropiado.

La aplicación de los algoritmos propuestos necesita disponer de medidas de la posición, velocidad y aceleración angulares de las articulaciones para realizar el cálculo de los parámetros del modelo. Se ha podido verificar que la calidad de estas señales es determinante para la correcta estimación de los mismos. Para ello se analizó la mejor forma de obtener estas señales, proceso descrito en el Cap. 5 de [BP09], optando finalmente por la aplicación de un filtro Savitzky-Golay (filtro SG) de quinto orden para la obtención numérica de la velocidad y la aceleración a partir de la medida de la posición proporcionada por los encoder de las articulaciones.

Es necesario reseñar además que la plataforma de experimentación aplica una señal de par de control referenciada en una escala entre -10 V y $+10$ V. No ha sido posible determinar la correspondencia entre la señal en voltios y el par en unidades del Sistema Internacional, si bien no es necesario a efectos de control o identificación, sabiendo que ambas señales son aproximadamente proporcionales. Por este motivo, las unidades consignadas en los parámetros identificados están referenciadas a una señal de par medida en voltios.

Los pasos que se siguieron en el proceso de identificación fueron los siguientes:

1. Transformación matemática de las ecuaciones para poder extraer la información de los resultados de forma adecuada.
2. Selección de una señal que nos aporte suficiente información de todo el rango de comportamientos del sistema.
3. Realización de experimentos en bucle abierto con una entrada conocida para observar la salida.
4. Filtrado y análisis de los datos.
5. Comprobación en simulación si los resultados conseguidos hacen que el modelo matemático obtenido se asemeje al real.

A continuación se detallará más profundamente cada uno de los pasos seguidos para identificar todos los parámetros.

2.3. Procedimiento de identificación

2.3.1. Transformación matemática de las ecuaciones

A la vista de las ecuaciones 1.14, 1.15 y 1.16 deberemos manipularlas matemáticamente para dejarlas de tal forma que se pueda obtener un regresor, es decir, un vector que tiene tantas componentes como parámetros constantes deseamos identificar. Tiene la forma:

$$u = \varphi\theta^T$$

El regresor del que hablamos es el vector θ cuyas componentes son las constantes de k_6 a k_{10} ambas inclusive. Este vector será:

$$\theta^T = \begin{bmatrix} k_6 \\ k_7 \\ k_8 \\ k_9 \\ k_{10} \end{bmatrix} \quad (2.1)$$

Como se comentó en la introducción el hombro permanecerá bloqueado para todos los experimentos de identificación. A nivel matemático esto implica que la primera ecuación del sistema, la que concierne a τ_1 , se cumplirá automáticamente y el par aplicado se convertirá en un par de “*reacción*” puesto que la articulación esta completamente quieta debido a un freno mecánico. Esto afecta indirectamente a las otras dos ecuaciones porque no variará ni la posición, ni la velocidad, ni la aceleración del hombro, esto es:

$$q_1 = \dot{q}_1 = \ddot{q}_1 = 0$$

Bajo estas condiciones de operación las dos ecuaciones restantes del sistema quedan de la siguiente manera:

Ecuación 2

$$\begin{aligned} & [k_3 + k_6 + k_8 + 2k_7\cos(q_3)] \cdot \ddot{q}_2 + \\ & [k_8 + k_7\cos(q_3)] \cdot \ddot{q}_3 - \\ & [k_7\sin(q_3)] \cdot \dot{q}_3^2 - [2k_7\sin(q_3)] \cdot \dot{q}_2\dot{q}_3 + \\ & (k_5 + k_9)\sin(q_2) + k_{10}\sin(q_2 + q_3) + F_{f2} = \tau_2 \end{aligned} \quad (2.2)$$

Ecuación 3

$$\begin{aligned} & [k_8 + k_7\cos(q_3)] \cdot \ddot{q}_2 + \\ & [k_8] \cdot \ddot{q}_3 + \\ & [k_7\sin(q_3)] \cdot \dot{q}_2^2 + \\ & k_{10}\sin(q_2 + q_3) = 0 \end{aligned} \quad (2.3)$$

Persiguiendo nuestro objetivo de hallar un regresor, θ , con la forma de la Ecuación 2.1 dejamos las ecuaciones anteriores en función de los parámetros k_6 a k_{10} . Y así las ecuaciones se transforman en:

Ecuación 2

$$\begin{aligned}
& [\ddot{q}_2] \cdot k_6 + \\
& \left[2\ddot{q}_2 \cos(q_3) + \ddot{q}_3 \cos(q_3) - \dot{q}_3^2 \sin(q_3) - 2\dot{q}_2 \dot{q}_3 \sin(q_3) \right] \cdot k_7 + \\
& [\ddot{q}_2 + \ddot{q}_3] \cdot k_8 + \\
& [\sin(q_2)] \cdot k_9 + \\
& [\sin(q_2 + q_3)] \cdot k_{10} = \tau_2 - k_5 \sin(q_2) - k_3 \ddot{q}_2 - F_{f2}
\end{aligned} \tag{2.4}$$

Ecuación 3

$$\begin{aligned}
& [0] \cdot k_6 + \\
& \left[\ddot{q}_2 \cos(q_3) + \dot{q}_2^2 \sin(q_3) \right] \cdot k_7 + \\
& [\ddot{q}_2 + \ddot{q}_3] \cdot k_8 + \\
& [0] \cdot k_9 + \\
& [\sin(q_2 + q_3)] \cdot k_{10} = 0
\end{aligned} \tag{2.5}$$

Podemos escribirlo en forma matricial para distinguir la matriz φ y el vector u :

$$\varphi = \begin{bmatrix} \ddot{q}_2 & 2\ddot{q}_2 \cos(q_3) + \ddot{q}_3 \cos(q_3) - \dot{q}_3^2 \sin(q_3) - 2\dot{q}_2 \dot{q}_3 \sin(q_3) & \ddot{q}_2 + \ddot{q}_3 & \sin(q_2) & \sin(q_2 + q_3) \\ 0 & \ddot{q}_2 \cos(q_3) + \dot{q}_2^2 \sin(q_3) & \ddot{q}_2 + \ddot{q}_3 & 0 & \sin(q_2 + q_3) \end{bmatrix} \tag{2.6}$$

$$u = \begin{bmatrix} \tau_2 - k_5 \sin(q_2) - k_3 \ddot{q}_2 - F_{f2} \\ 0 \end{bmatrix} \tag{2.7}$$

Esto hace que las ecuaciones tengan una forma muy interesante para nosotros puesto que conocemos el valor de todos y cada uno de los términos menos de los parámetros constantes como se explica a continuación:

- q_2 : es la posición angular del codo proporcionada por el encoder en pulsos de encoder por vuelta.
- \dot{q}_2 : es la velocidad angular del codo, derivada offline de la posición mediante un filtro SG con una ventana de 15 puntos a cada lado.
- \ddot{q}_2 : es la aceleración angular del codo, derivada offline de la velocidad mediante un filtro SG con una ventana de 15 puntos a cada lado.
- q_3 : es la posición angular de la muñeca proporcionada por el encoder en pulsos de encoder por vuelta.
- \dot{q}_3 : es la velocidad angular de la muñeca, derivada offline de la posición mediante un filtro SG con una ventana de 15 puntos a cada lado.
- \ddot{q}_3 : es la aceleración angular de la muñeca, derivada offline de la velocidad mediante un filtro SG con una ventana de 15 puntos a cada lado.

- τ_2 : es el par de entrada al codo, medido por la tarjeta de adquisición de datos.
- k_3 y k_5 : son los valores de los parámetros identificados en el anterior trabajo, [BP09], con la misma plataforma.
- F_{f2} : es la fuerza de fricción que afecta al codo.

En la Ecuación 2.6 se ve que tenemos una matriz $\mathbb{M}(\mathbb{R})$ de tamaño 2×5 . Tal y como la hemos construido hemos colocado los coeficientes relacionados con los términos de la segunda ecuación en la primera fila y los relacionados con la tercera ecuación en la segunda fila. Ahora, por columnas, tenemos colocados los valores relacionados con k_6 en la primera columna, para k_7 en la segunda y así sucesivamente hasta la quinta columna con los valores para k_{10} .

Los experimentos de identificación realizados duran cada uno 30segs y teniendo en cuenta que nuestro sistema muestrea valores con una frecuencia de 1000Hz, lo que quiere decir que toma un dato cada milésima de segundo. Eso significa que vamos a tener una matriz de tamaño 2×5 cada milisegundo. Esto hace un total de 30.001 matrices al acabar cada experimento (30.000 matrices de cada instante de experimento más una para el instante cero).

Con el fin de tener los datos organizados en cada experimento se propone una manera de almacenar los datos obtenidos. En nuestra matriz φ tenemos separada la información que obtenemos de la segunda ecuación y de la tercera ecuación en la primera y segunda fila respectivamente. Como vamos a obtener 30.001 matrices de tamaño 2×5 montaremos una “*supermatriz*” de tamaño 60002×5 que incluya todos los resultados de cada prueba.

Esta nueva “supermatriz” será la matriz φ de cada experimento y, por supuesto, variará cada vez que ejecutemos un experimento aunque todos los parámetros de entrada que introduzcamos en la acción llevada a cabo sean iguales. Queda de manifiesto así que existen aspectos desconocidos que le confieren cierta incertidumbre al modelo y al sistema. Éstos son muchas veces difícilmente modelables y medibles, incluso en ocasiones es complicado imaginar de donde provienen.

Almacenando los datos de esta manera y utilizando los métodos de ajuste se podrá apreciar visualmente, mediante representación gráfica, el grado de ajuste que nos proporciona el regresor obtenido. A posteriori se puede comparar mediante simulación si el comportamiento de nuestro esquema se asemeja al de la plataforma experimental real, que siempre tiene multitud de perturbaciones e incertidumbres como ya hemos comentado.

2.3.2. Tratamiento de la fricción en el modelo

En un comienzo se pensó en despreciar ciertos aspectos de la fricción reduciendo el modelo completo a tan solo la fricción dinámica. Intuitivamente se podía creer que los efectos que no se tenían en cuenta aparecerían absorbidos de alguna manera por las constantes que se estaban identificando aunque, por el simple hecho de estar ignorando parte del modelo, sabíamos que los valores de los parámetros estarían, en cierta medida, subestimados o sobreestimados. Ni siquiera podríamos llegar a estimar en que medida estaba ocurriendo

esto y si estaba afectando profundamente a nuestras estimaciones. La única manera posible que disponíamos para hacernos una idea cualitativa era comparar el comportamiento del robot real y de nuestro modelo de simulación ante una entrada idéntica.

Se tomó el valor de la fricción dinámica previamente identificado en [BP09] y se comprobaron los resultados directamente sobre el robot para aceptar como válida o rechazar la hipótesis. El valor de la fricción tomado es:

$$F_d = 0,0612(V\text{seg}/\text{rad})$$

Tras los primeros experimentos se comprobaría que la fricción no tenía un efecto despreciable sobre el comportamiento del sistema. A la vista queda en la Figura 2.1 que el modelo no es capaz de predecir el comportamiento correctamente cuando gira en sentido horario más allá del punto de partida. La predicción conseguida de esta manera es poco deseable y poco fiable.

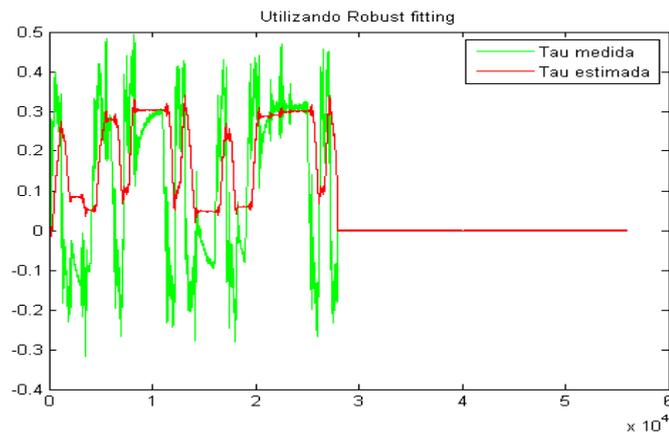


Figura 2.1: Experimento considerando la fricción dinámica

Así que optamos por introducir el *modelo de LuGre* de fricción cuyas ecuaciones son:

$$\begin{aligned} \frac{dz}{dt} &= \dot{q} - \frac{\sigma_0}{g(\dot{q})} z |\dot{q}| \\ g(\dot{q}) &= F_c + (F_s - F_c) e^{-(\dot{q}/\dot{q}_s)^2} \\ F_f &= \sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 \dot{q} \end{aligned} \quad (2.8)$$

La medida de la fuerza de fricción según el modelo de LuGre no se puede realizar directamente puesto que depende de variables como z que son una medida del estado interno del sistema y de los mecanismos que producen la fricción. No obstante tenemos la ventaja de poder heredar del anterior trabajo ([BP09]) un bloque de simulink que nos devuelve la fuerza que ejerce la fricción. Se contemplan sus distintos parámetros, depende de la posición y de la velocidad así como del lado hacia el que gire. El modelo que utilizamos en Simulink es el representado en la Figura 2.2.

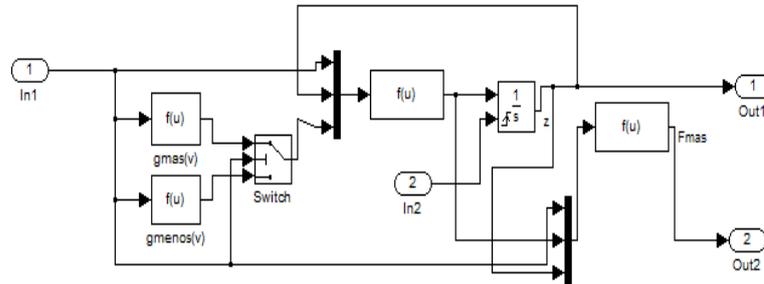


Figura 2.2: Bloque para el cálculo de la fricción en el codo

No obstante, era de esperar que incluyendo tan solo la fricción dinámica no existiera mucha variación respecto a no considerarla. Si tenemos en cuenta los valores de los otros coeficientes a la hora de hallar la fuerza de fricción nos damos cuenta que son varios órdenes de magnitud superiores al de la fricción dinámica por tanto era plausible que no modificara mucho los resultados. Veamos la siguiente Tabla 2.1:

$\sigma_0 = 42.9609 \text{ (Vs/rad)}$	Coficiente de rigidez
$\sigma_1 = 2.4593 \text{ (Vs/rad)}$	Coficiente de amortiguamiento
$\sigma_2 = 0.0612 \text{ (Vs/rad)}$	Coficiente de fricción dinámica

Tabla 2.1: Valores de los tres coeficientes de la fricción

2.3.3. Seleccionar una señal adecuada

Se eligieron tres señales como posibles candidatas para convertirse en la señal de entrada:

1. Señal sinusoidal estándar, Figura 2.3
2. Señal sinusoidal de frecuencia linealmente creciente con el tiempo, Figura 2.4
3. Señal pseudo-aleatoria *Gold Sequence*, Figura 2.5(b)

La señal de entrada que nosotros podemos controlar, o sea la acción de control, es el par del motor, tanto para el codo como para el hombro pero como sabemos el hombro permanecerá bloqueado. Esta señal de entrada no es exactamente la que recibe la articulación puesto que el servo tiene en su interior un lazo de control que ejerce de seguidor de par. No obstante podemos asegurar que será lo suficientemente próxima a la señal que nosotros elegimos como para no variar sustancialmente el comportamiento de nuestro sistema.

Para saber como será la respuesta de un sistema ante diferentes entradas se puede revisar, por ejemplo, el diagrama de Bode del sistema siempre y cuando se disponga del mismo. En nuestro caso no es conocido. Entonces, la idea es conseguir una señal de par que haga que el robot recorra un amplio rango de frecuencias representativo para medir la salida y, de esta manera, aproximar lo más precisamente posible el comportamiento del robot para las diferentes señales de entrada posibles.

El espectro de frecuencia caracteriza qué amplitud presenta la respuesta del sistema para cada frecuencia de un fenómeno con una variación de frecuencia en su entrada. Esto es exactamente lo que se comprueba y lo que se sabe cuando se lee un diagrama de Bode o un diagrama de Nyquist. Es un “*mapa*” de las respuestas de un sistema en función de la frecuencia con la que se está alimentando la señal de entrada pertinente. Por eso buscamos un espectro rico en frecuencias bajas que serán las de funcionamiento del robot.

Sabiendo esto la señal sinusoidal de la Figura 2.3 queda descartada desde el principio porque su espectro frecuencial es muy limitado, de hecho la señal solo contiene una única frecuencia elegida a nuestro parecer.

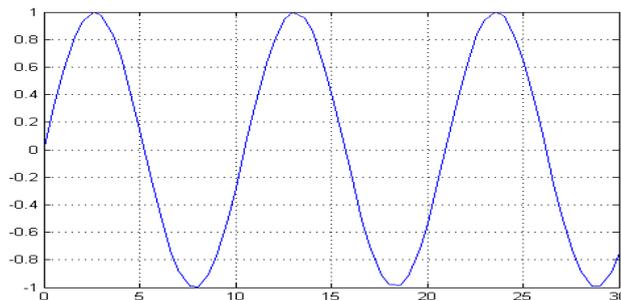


Figura 2.3: Entrada sinusoidal

La segunda señal candidata la hemos producido a través de un bloque de la librería de Simulink que se llama *Chirp*. Esta señal consiste en una función seno estándar, por ejemplo: $y(t) = A \sin(\omega t)$ en la que el argumento ha sido modificado para que la frecuencia aumente linealmente con el tiempo, por ejemplo $y'(t) = A \sin(B t \omega t)$, donde B es una constante elegida arbitrariamente, t es el tiempo y ω es la frecuencia angular.

Es obvio que esta señal recorrería todo el espectro de frecuencias. También nos aporta más información que una señal periódica pero para no producir una variación brusca de la frecuencia y no alargar los experimentos de identificación a un tiempo elevado necesitamos variar las frecuencias con un coeficiente B pequeño. En los experimentos realizados hemos variado la señal desde $0,5\text{Hz}$ a $1,5\text{Hz}$ en un período de 15 segundos. Podemos hacernos a la idea de lo que nos costaría llegar a frecuencias más elevadas aunque no sea necesario debido a que nuestro sistema las filtraría automáticamente debido a su naturaleza mecánica.

Finalmente la señal que se escogió es una Gold Sequence pura filtrada mediante un *paso bajo* para conseguir una entrada que se asemeja a la respuesta de un sistema de primer

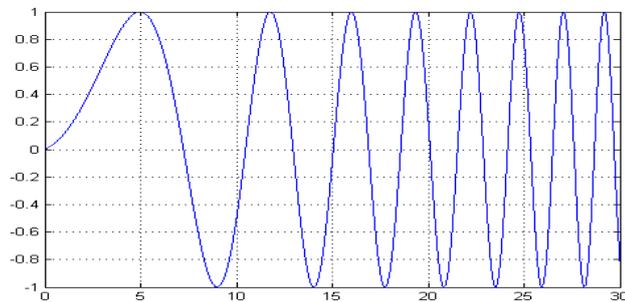
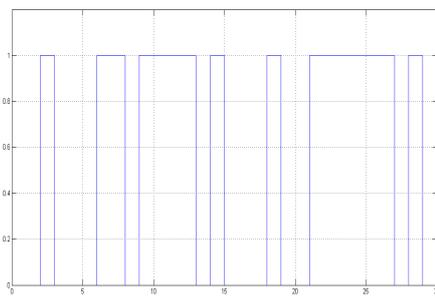
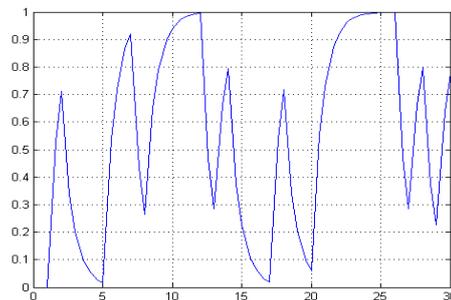


Figura 2.4: Entrada sinusoidal de frecuencia linealmente creciente

orden, con un tiempo de respuesta de $t_r = 0,8\text{seg}$, al que se le introducen escalones positivos y negativos siempre de amplitud unidad. El tiempo que se mantiene el valor del escalón varía lo que hace que el sistema sea capaz (o no) de llegar al estado estacionario, esto es, variar la frecuencia de la entrada de una manera *pseudoaleatoria*. Se obtuvo esta señal desde las librerías de Simulink sin cambiar ninguno de los parámetros que tenía configurados por defecto.



(a) Gold Sequence pura



(b) Gold Sequence a través de un filtro paso-bajo

Figura 2.5: Entrada Gold Sequence

2.3.4. Realización de experimentos en bucle abierto

En la Figura 2.6 vemos el esquema utilizado en Simulink para la realización de todos los experimentos de identificación. Cabe resaltar que no se aprecia ningún tipo de realimentación puesto que todas las pruebas se llevan a cabo en bucle abierto. Teniendo en cuenta esto se debe tener cierta precaución.

El motor del codo es suficientemente potente como para inestabilizar el sistema con cierta facilidad y la muñeca no cuenta con ninguna acción sobre ella, por tanto no se podrá detener tal y como se puede hacer con el codo y el hombro. Tras la experiencia que he tenido probando diferentes valores recomendaría para experimentos en bucle abierto no superar una amplitud de $A = 1V$, ni una frecuencia de $\omega = 1,5\text{rad/s}$.

Esquema para identificación parámetros tercer GDL

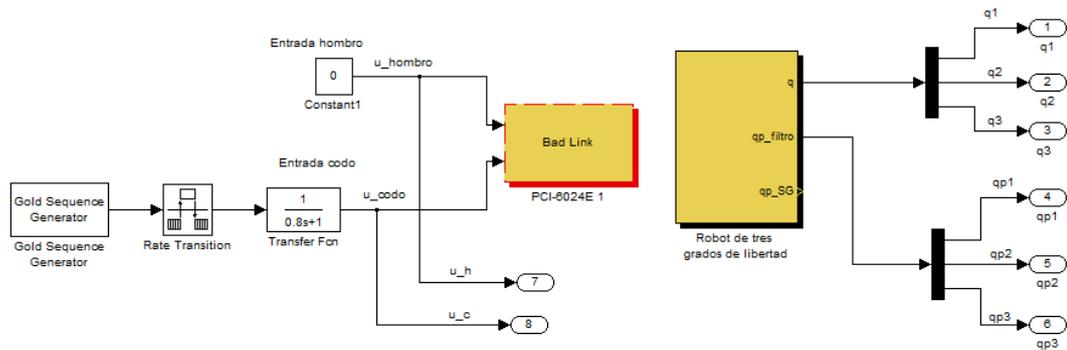


Figura 2.6: Esquema de identificación

En el esquema de la Figura 2.6 se comprueba, como ya se había explicado anteriormente, que la entrada en el hombro es de $0V$. El hombro está bloqueado mecánicamente así que podríamos ponerle la entrada que quisieramos que no se movería pero no tiene ningún sentido estar forzando a la articulación y al motor a moverse cuando no pueden hacerlo. Se producirían tensiones internas que provocarían un deterioro del montaje mucho más rápido de lo que debería ser. Bloqueado de esta manera el hombro actuará exactamente igual que la bancada.

El bloque “*Rate Transition*” que aparece después del filtro paso-bajo, con una constante de tiempo $\tau = 0,8$, se encarga de alimentar las tarjetas de comunicación con el robot cada milisegundo, que es el mismo tiempo de muestreo con el que trabajan las tarjetas de adquisición de datos. Así obtendremos información de nuestro robot cada milisegundo.

En el otro lado tenemos el bloque amarillo, “*Robot de tres grados de libertad*”, del que salen dos líneas que nos dan la medida de la posición y la velocidad de hombro, codo y muñeca. Como sabemos, para el hombro, posición, velocidad y aceleración serán cero en todo momento. Los encoders nos proporcionan la medida directa de la posición para el codo y la muñeca. La velocidad se calcula en tiempo real mediante un filtro de derivación que nos da una medida suficientemente precisa para nuestro propósito.

2.3.5. Filtrado y análisis de los datos

Para el filtrado y análisis de los datos se utilizó en MATLAB una función, modificada del anterior trabajo [BP09] y ligeramente modificada, que cogía los resultados del experimento y los volcaba al *Workspace* desde donde se organizaban como nos interesaba mediante otra función programada por nosotros para construir la matriz φ y el vector u , que es un vector 60002×1 que tiene en la primera mitad una entrada de par modificada por los parámetros constantes ya conocidos y por la fricción del codo, la segunda mitad son todos ceros porque

no existe ninguna entrada para la muñeca.

Las primeras 30001 filas corresponden a la Ecuación 2 del movimiento, ver Ec. 2.4, y las filas restantes hasta el final corresponden a la Ecuación 3 del movimiento, ver Ec. 2.5.

En un inicio se introdujo un método de filtrado eliminando de la matriz todas las filas que estuvieran relacionadas con una velocidad del codo, $\dot{q}_2 \leq \dot{q}_s$, siendo \dot{q}_s la velocidad de Stribeck relacionada con el fenómeno de fricción del deslizamiento. Posteriormente se comprobó gráficamente que el ajuste era mejor cuando se consideraban todos los datos sin eliminar ninguno.

2.3.6. Método de ajuste: Ajuste Robusto - *Robust Fitting*

Una vez que teníamos todos los datos almacenados como necesitábamos solo quedaba aplicar un algoritmo de ajuste para obtener los valores de los parámetros que buscábamos. Mediante una función predefinida de MATLAB de *ajuste robusto* obtendremos los resultados finales. Esta función solo necesita como entrada la matriz φ que tenemos y el vector u y te devuelve el regresor θ que perseguíamos.

Se eligió el ajuste robusto frente al ajuste por mínimos cuadrados porque el primero es capaz de tratar con valores marginales “*defectuosos*”. Estos datos erróneos actúan de puntos palanca que hacen que todo el ajuste se desplace hacia su situación, mayor será el desplazamiento cuanto mayor sea el valor del error. Esto no nos produce ningún beneficio. Por tanto, se escogió el método de ajuste robusto para poder eliminar de nuestra identificación estos *puntos palanca* que producirían que los parámetros identificados absorbieran un error que alejaría de la realidad el comportamiento de nuestro modelo.

Este método, en Matlab, tiene una peculiaridad. El regresor que te devuelve tiene una componente más de las que esperaríamos, es decir, devuelve 6 componentes en vez de 5. Esta componente extra da una medida de la desviación constante que tiene todo el experimento completo. Se podría entender como el “*offset*” que tiene una señal alterna. Se espera que el valor de esta nueva componente se acerque lo máximo posible a cero como un indicador de que la identificación parece ir por buen camino.

2.3.7. Resultados

Se presentan en este apartado los resultados obtenidos después de seguir todo el proceso descrito anteriormente. Se pueden comprobar las Figuras 2.7 y 2.8 donde aparecen, en verde, el vector u medido sobre el robot, utilizado en la identificación y, en rojo, el vector u' estimado mediante la multiplicación de la matriz φ con los parámetros identificados en nuestro regresor θ^T , esta operación nos devuelve nuestra estimación del par.

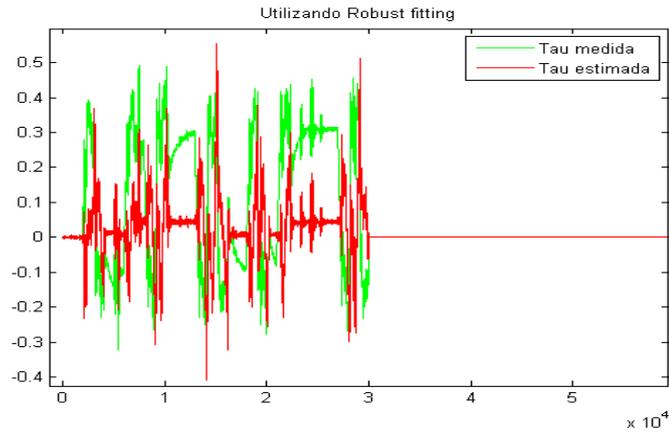


Figura 2.7: Experimento identificación para 5 parámetros con Golden Sequence

Para los experimentos que representa la Figura 2.7 se utilizó la entrada de la Figura 2.5 que es la utilizada en todos los experimentos de identificación. Aquí puede apreciarse que el comportamiento que predice nuestro regresor se aleja de la realidad notablemente. Queda de manifiesto que los valores obtenidos no son aceptables cuando observamos el regresor que hemos obtenido:

$$\theta^T = \begin{bmatrix} \text{Offset} \\ k_6 \\ k_7 \\ k_8 \\ k_9 \\ k_{10} \end{bmatrix} = \begin{bmatrix} 0,0000 \\ -0,0011 \\ 0,0054 \\ -0,0010 \\ 0,2808 \\ -0,1333 \end{bmatrix} \quad (2.9)$$

No se pueden dar por válidos parámetros negativos puesto que si revisamos la definición de los parámetros constantes en la Tabla 1.1 no tiene ningún sentido físico real pensar en un producto entre longitudes, masas, gravedad e inercias, en cualquiera de sus combinaciones, que de como resultado un número negativo. Hay que encontrar una manera de solucionar este inconveniente.

Merece la pena detenerse a mirar la Figura 2.8. La gráfica en rojo que se puede apreciar está generada con el regresor obtenido en el experimento utilizando la entrada Gold Sequence de la Figura 2.5. La predicción que obtenemos es aceptable puesto que la curva parece seguir, con cierto retraso, a la trayectoria de par marcada. Es preciso remarcar, viendo este caso, que pese a que tengamos valores negativos en el regresor la aproximación podría ser *matemáticamente* válida. Lo que nunca debemos hacer es perder de vista el modelo físico real del sistema que es la razón por la cual desechamos cualquier regresor que incluya un valor negativo.

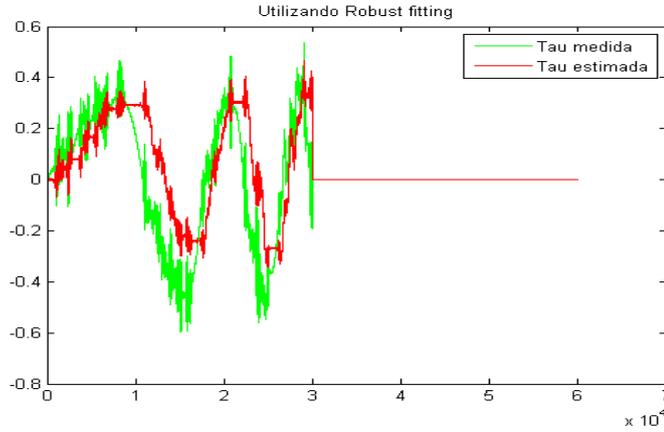


Figura 2.8: Experimento identificación para 5 parámetros con Chirp

2.4. Identificación con 3 constantes

Se plantea como solución alternativa introducir un factor constante conocido deducido directamente de las definiciones de los parámetros físicos con el fin de simplificar la identificación de 5 parámetros a solo 3. Esto hará que la identificación pierda un poco de flexibilidad puesto que ahora el ajuste involucra menos parámetros que antes. Podría entenderse como un ajuste de 5^o orden frente a uno de 3^{er} orden aunque ésta sea meramente una idea intuitiva. Aún así se busca que el regresor represente al sistema.

Todo el procedimiento descrito anteriormente es exactamente igual con el único cambio que la matriz φ que antes tenía 5 columnas ahora pasa a tener 3 y los parámetros k_6 y k_9 quedan definidos indirectamente al realizar la identificación de k_7 y k_{10} .

2.4.1. Replanteamiento de las ecuaciones del sistema

A sabiendas de que las constantes k_6 , k_7 , k_9 y k_{10} están definidas según la Ecuación 2.10:

$$\begin{aligned} k_6 &= L^2 m_3, & k_7 &= L_{c3} L m_3 \\ k_9 &= L m_3 g, & k_{10} &= L_{c3} m_3 g \end{aligned} \quad (2.10)$$

Descubrimos que existe la misma relación de distancias entre las constantes k_6 y k_7 que entre k_9 y k_{10} , la cual es sencillamente medible.

$$\left. \begin{aligned} \frac{k_6}{k_7} &= \frac{k_9}{k_{10}} = \frac{L}{L_{c3}} \\ \frac{L}{L_{c3}} &= \frac{450}{189} = 2,3809 \end{aligned} \right\} \Rightarrow \frac{k_6}{k_7} = \frac{k_9}{k_{10}} = 2,3809 = R_L \quad (2.11)$$

Por tanto:

$$k_6 = 2,3809 \cdot k_7$$

$$k_9 = 2,3809 \cdot k_{10}$$

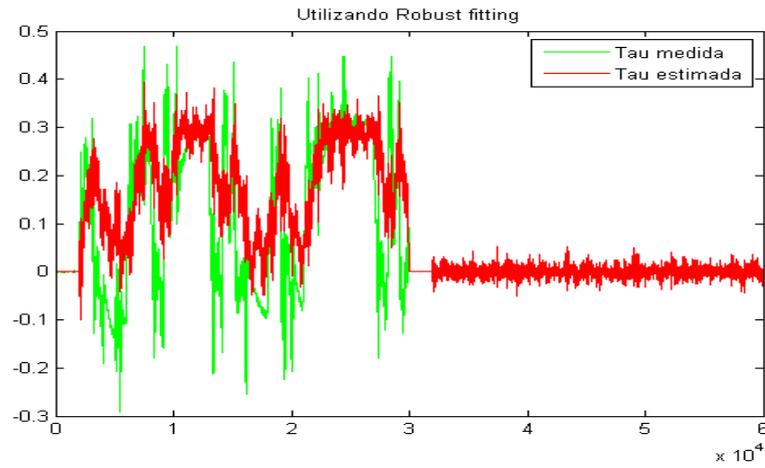


Figura 2.9: Experimento identificación para 3 parámetros con Golden Sequence

Los parámetros obtenidos en este experimento de identificación son los que aparecen en la Ecuación 2.15.

$$\theta_3^T \text{ Constantes} = \begin{bmatrix} \text{Offset} \\ k_7 \\ k_8 \\ k_{10} \end{bmatrix} = \begin{bmatrix} 0,0001 \\ 0,0063 \\ 0,0044 \\ 0,2235 \end{bmatrix} \Rightarrow \theta^T = \begin{bmatrix} 2,3809 \cdot k_7 \\ k_7 \\ k_8 \\ 2,3809 \cdot k_{10} \\ k_{10} \end{bmatrix} = \begin{bmatrix} 0,0151 \\ 0,0063 \\ 0,0044 \\ 0,5322 \\ 0,2235 \end{bmatrix} \quad (2.15)$$

2.4.3. Validación de resultados

A la vista del regresor obtenido podemos decir, al menos, que los parámetros cumplen con la condición necesaria de no negatividad. Para asegurarnos de que al asumir estos valores no nos estamos alejándonos en exceso del comportamiento real del sistema se realizan varios experimentos con otras entradas diferentes, usando en este caso las señales *Chirp* y *seno* de la Figura 2.4 y 2.3 respectivamente.

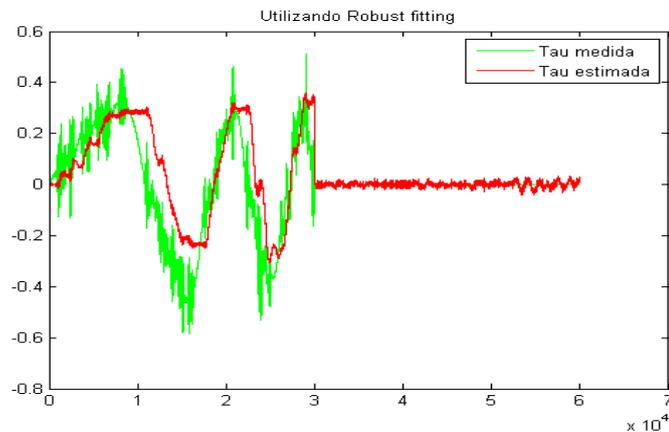


Figura 2.10: Comprobación para 3 parámetros identificados con Chirp

Utilizando los parámetros obtenidos comparamos el par real introducido por nosotros y el par predicho con nuestro regresor identificado. Se aprecia una similitud entre las curvas aceptable. Debe existir un fenómeno más involucrado en la fricción que no hemos conseguido representar por completo en nuestro modelo ya que en la predicción aparece un pequeño retraso. La forma que sigue la curva aparenta ser la misma pero retrasada un breve instante de tiempo.

También se puede apreciar en la Figura 2.10 que cuando el robot gira en sentido antihorario el modelo no llega a predecir el máximo par. Esto puede deberse a múltiples razones pero entre otras la fricción se ha podido ver modificada con el paso del tiempo por el desgaste o simplemente porque es diferente en una dirección y en otra.

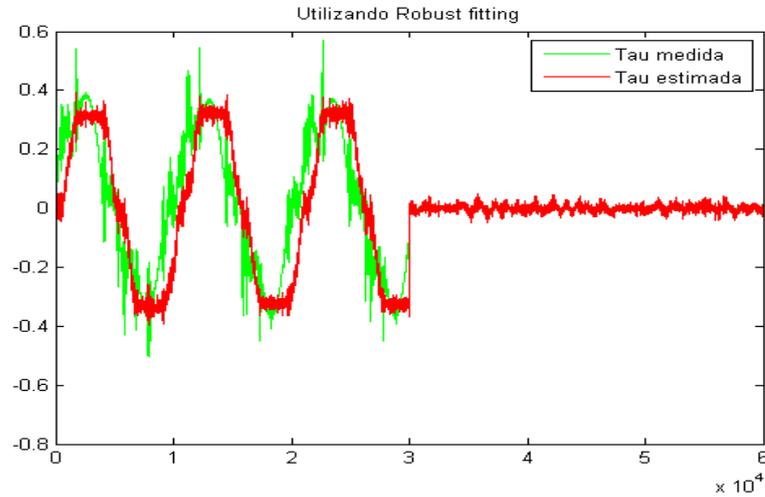


Figura 2.11: Comprobación para 3 parámetros identificados con seno de $\omega = 0,6\text{rad/s}$

Existen dos relaciones deducidas a partir de la definición de los parámetros que pueden utilizarse para comprobar, relativamente, la *fiabilidad* de los resultados obtenidos. Puesto que la relación es la misma el cociente entre los parámetros debe ser bastante parecido, aceptando como un buen resultado siempre que sean del mismo orden de magnitud.

$$\left. \begin{aligned} \frac{k_2}{k_6} &= \frac{m_2 L L_{c2}}{m_3 L^2} = \frac{m_2 L_{c2}}{m_3 L} = 0,3046 \\ \frac{k_5}{k_9} &= \frac{m_2 L_{c2} g}{m_3 L g} = \frac{m_2 L_{c2}}{m_3 L} = 0,9301 \end{aligned} \right\} \Rightarrow \frac{k_2}{k_6} = \frac{k_5}{k_9} \Rightarrow \frac{0,3046}{0,9301} = 0,3275 \quad (2.16)$$

Aunque uno de los dos cocientes es tres veces mayor que el otro ambos son del mismo orden de magnitud lo que nos lleva a pensar que la identificación ha ido por el buen camino, que los parámetros que hemos identificado son capaces de representar el comportamiento de nuestro sistema de manera satisfactoria y que los valores que hemos obtenido concuerdan con los heredados del anterior trabajo [BP09].

2.5. Resultados definitivos

La Tabla 2.2 es una tabla resumen de todos los parámetros que se utilizarán de aquí en adelante. Tanto los parámetros de k_1 a k_5 , como los σ_0 , σ_1 y σ_2 son valores tomados como válidos del anterior trabajo [BP09], aunque también asumimos que con el paso del tiempo, el deterioro del robot y otras razones hayan podido variar ligeramente.

Parámetro	Valor
k_1	$0,1386 V \cdot s^2/rad$
k_2	$0,0046 V \cdot s^2/rad$
k_3	$0,0372 V \cdot s^2/rad$
k_4	$2,183 V \cdot s^2/rad$
k_5	$0,4950 V \cdot s^2/rad$
k_6	$= 2,3809 \cdot k_7 = 0,0151 V \cdot s^2/rad$
k_7	$0,0063 V \cdot s^2/rad$
k_8	$0,0044 V \cdot s^2/rad$
k_9	$= 2,3809 \cdot k_{10} = 0,5322 V \cdot s^2/rad$
k_{10}	$0,2235 V \cdot s^2/rad$
σ_0	$42,9609 V \cdot s/rad$
σ_1	$2,4593 V \cdot s/rad$
σ_2	$0,0612 V \cdot s/rad$
F_{f3}	≈ 0 , despreciable frente a F_{f2}
R_L	$2,3809$ (<i>adimensional</i>)

Tabla 2.2: Tabla resumen de los valores de los parámetros del robot

Capítulo 3

Controlador

3.1. Introducción

Una vez halladas las ecuaciones del movimiento de nuestro sistema y habiendo realizado la identificación podemos asegurar que nuestro modelo matemático ha quedado completamente definido. A partir de este punto podemos empezar a realizar las simulaciones para cualquier tipo de controlador y comportamiento que queramos. Por otra parte, aunque el modelo matemático es perfectamente conocido tendremos que “montar” en *Simulink* un modelo basado en bloques para agilizar y simplificar nuestra tarea. Ésto también hace que la tarea se haga mucho más sencilla y visual pudiendo comprobar los resultados de cada simulación rápidamente.

En este capítulo se pretende explicar de manera teórica, sin profundizar en exceso, el controlador que se utilizará sobre el robot. Con este regulador basado en una aproximación de la energía del sistema se realizarán las simulaciones y los experimentos para estabilizar el sistema en una posición de equilibrio inestable en capítulos posteriores.

Se procederá al ajuste del controlador sobre el modelo de simulación hasta conseguir el resultado que buscamos y, a la postre, llevaremos el controlador ajustado al modelo real, en el que sufrirá ciertos cambios, para conseguir que el comportamiento sea el adecuado puesto que siempre existen incertidumbres sin modelar.

El sistema de control que se utilizará para la estabilización consta de dos etapas:

1. *Swing Up*: se encargará de elevar el codo y la muñeca hasta un entorno de la posición vertical, $(q_2, q_3) = (\pi, 0)$.
2. *Ballancing Control*: se encargará de mantener el codo y la muñeca en la posición vertical.

A continuación vamos a explicar cada una de las dos partes más detalladamente con el desarrollo matemático necesario. La base de estas explicaciones aparecen en [Ted09].

3.2. Swing Up

3.2.1. Objetivo

El objetivo de la primera etapa, el controlador *Swing Up*, es conseguir llevar el sistema de dos grados de libertad con una sola acción de control que, de aquí en adelante llamaremos *Pendubot*, hasta un entorno de la posición de equilibrio superior inestable, como un péndulo invertido. Ésto es donde la posición del codo es $q_2 = \pi$ y la muñeca sigue su dirección en la vertical $q_3 = 0$ y la velocidad del codo y de la muñeca es nula: $\dot{q}_2 = 0$ y $\dot{q}_3 = 0$.

3.2.2. Desarrollo teórico

El desarrollo teórico que haremos a continuación utilizará la nomenclatura que hemos venido utilizando en capítulos anteriores. Tratando los elementos de las matrices del modelo como términos para simplificarlo y hacerlo más corto porque, de otra manera, las ecuaciones podrían ser incómodamente largas. En este apartado se ha seguido las pautas dadas en los artículos [FLS00] y [XKO02] adaptándolo a nuestro sistema y a las ecuaciones que habíamos obtenido nosotros previamente.

El modelo que tenemos de nuestro robot es, en realidad, de tres grados de libertad pero sabemos que tiene la primera articulación bloqueada, es decir:

$$q_1 = \dot{q}_1 = \ddot{q}_1 = 0$$

Lo que quiere decir que la primera fila y la primera columna de nuestras matrices quedan anuladas por tanto las matrices con las que trabajaremos serán 2×2 y se tomarán los elementos (2, 2), (2, 3), (3, 2) y (3, 3). Aún así nuestro modelo sigue siendo:

$$D(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + G(q) + F_f(q, \dot{q}) = \tau(t) \quad (3.1)$$

Donde

$$\begin{bmatrix} d_{22} & d_{23} \\ d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} c_{22} & c_{23} \\ c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} G_2 \\ G_3 \end{bmatrix} + \begin{bmatrix} F_{f2} \\ F_{f3} \end{bmatrix} = \begin{bmatrix} \tau_2 \\ 0 \end{bmatrix} \quad (3.2)$$

La energía del sistema se calculará mediante la siguiente expresión:

$$E = E_c + E_p \quad (3.3)$$

$$E_c = \frac{1}{2} \dot{q}^T D(q) \dot{q} \quad (3.4)$$

$$E_p = (1 - \cos(q_2))(m_2 L_{c2} g + m_3 L g) + (1 - \cos(q_2 + q_3))(m_3 L_{c3} g) \quad (3.5)$$

Utilizando la definición de las constantes:

$$k_5 = m_2 L_{c2} g; \quad k_9 = m_3 L g; \quad k_{10} = m_3 L_{c3} g$$

La energía potencial se define como:

$$E_p = (k_5 + k_9)(1 - \cos(q_2)) + (k_{10})(1 - \cos(q_2 + q_3)) \quad (3.6)$$

Así la expresión de la energía total del sistema utilizando los parámetros anteriormente identificados es:

$$E_T = \frac{1}{2} \dot{q}^T D(q) \dot{q} + (k_5 + k_9)(1 - \cos(q_2)) + (k_{10})(1 - \cos(q_2 + q_3)) \quad (3.7)$$

La energía del Pendubot cuando esté en reposo sobre su posición de equilibrio se reducirá a únicamente su energía potencial. En equilibrio, detenido en su posición superior, se puede resumir como $(q_2, \dot{q}_2, q_3, \dot{q}_3) = (\pi, 0, 0, 0)$, por tanto:

$$E_{eq} = (k_5 + k_9)(1 - \cos(\pi)) + (k_{10})(1 - \cos(\pi + 0)) = 2(k_5 + k_9 + k_{10}) \quad (3.8)$$

Si monitorizamos la energía del sistema debemos saber que debe converger a la energía definida en la Ecuación 3.8. Cuanto más rápido alcancemos el equilibrio más rápida sera la convergencia y viceversa.

Nuestro siguiente paso será intentar llevar al Pendubot hasta una *órbita homoclina*. Una órbita homoclina es un estado del movimiento del robot en el que el codo permanecerá en un entorno del equilibrio, en $q_2 = (\pi + \epsilon)$, y la muñeca oscilará tan cerca como queramos del punto de equilibrio, realizando ciclos casi completos, adquiriendo una fracción más de energía en cada oscilación. La órbita homoclina se define mediante la siguiente trayectoria:

$$\frac{1}{2} k_8 \dot{q}_3^2 = k_{10}(1 - \cos(q_3)) \quad (3.9)$$

Se puede comprobar como se ha deducido las Ecuacione 3.9 a través de su correspondiente desarrollo en el artículo [FLS00] o [XKO02]. Resolver el problema de llevar al robot hasta la órbita homoclina es, básicamente, conseguir el controlador Swing Up que nosotros perseguimos.

Consideremos un cambio de variable como el propuesto a continuación:

$$\tilde{q}_2 = (q_2 - \pi) \quad (3.10)$$

$$\tilde{E} = E - E_{eq} \quad (3.11)$$

Pretendemos así conseguir que tanto la energía total del Pendubot como la posición del codo se hagan cero cuando se encuentre en el punto de equilibrio deseado.

Se utiliza ahora una función de Lyapunov, que es una herramienta matemática de gran ayuda a la hora de asegurar la convergencia hacia la estabilidad de un sistema. El presente trabajo no tiene como objetivo el estudio de estas funciones pero se mencionan porque son necesarias en el desarrollo teórico del controlador. Así, se propone la siguiente función de Lyapunov:

$$V(q, \dot{q}) = \frac{k_E}{2} \tilde{E}(q, \dot{q})^2 + \frac{k_D}{2} \dot{q}_2^2 + \frac{k_P}{2} \tilde{q}_2^2 \quad (3.12)$$

Donde las constantes k_E , k_D y k_P son estrictamente positivas y se definirán más adelante. En definitiva, ajustar las constantes se convertirá en el principal problema para obtener el controlador deseado. Si derivamos V en función del tiempo obtenemos:

$$\dot{V} = k_E \tilde{E} \dot{\tilde{E}} + k_D \dot{q}_2 \ddot{q}_2 + k_P \tilde{q}_2 \dot{q}_2 = \dot{q}_2 \left(k_E \tilde{E} \tau_2 + k_D \dot{q}_2 + k_P \tilde{q}_2 \right) \quad (3.13)$$

A modo de recordatorio, los elementos que necesitamos de las matrices $D(q)$ y $C(q, \dot{q})$ estaban definidos como:

$$\begin{aligned} d_{22} &= k_3 + k_6 + k_8 + 2k_7 \cos(q_3) \\ d_{23} &= k_8 + k_7 \cos(q_3) \\ d_{32} &= k_8 + k_7 \cos(q_3) \\ d_{33} &= k_8; \end{aligned} \tag{3.14}$$

$$\begin{aligned} c_{22} &= -k_7 \dot{q}_3 \sin(q_3) \\ c_{23} &= -k_7 \dot{q}_1 \sin(q_3) - k_7 \dot{q}_2 \sin(q_3) - k_7 \dot{q}_3 \sin(q_3) \\ c_{32} &= k_7 \dot{q}_1 \sin(q_3) + k_7 \dot{q}_2 \sin(q_3) \\ c_{33} &= 0 \end{aligned} \tag{3.15}$$

Necesitamos una expresión de la aceleración del codo, \ddot{q}_2 , que podemos despejar del modelo reducido presentado en la Ecuación 3.2:

$$\begin{bmatrix} \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} d_{22} & d_{23} \\ d_{32} & d_{33} \end{bmatrix}^{-1} \left(\begin{bmatrix} \tau_2 \\ 0 \end{bmatrix} - \begin{bmatrix} c_{22} & c_{23} \\ c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} - \begin{bmatrix} G_2 \\ G_3 \end{bmatrix} - \begin{bmatrix} F_{f2} \\ F_{f3} \end{bmatrix} \right) \tag{3.16}$$

De donde podemos resolver facilmente \ddot{q}_2 :

$$\ddot{q}_2 = d_{22}^{inv} (\tau_2 - c_{22}\dot{q}_2 - c_{23}\dot{q}_3 - G_2 - F_{f2}) - d_{23}^{inv} (c_{32}\dot{q}_2 + c_{33}\dot{q}_3 + G_3 + F_{f3}) \tag{3.17}$$

Los términos d_{22}^{inv} y d_{23}^{inv} representan el elemento (2,2) y (2,3) respectivamente de la inversa de la matriz $D(q)$ que se calculará sencillamente en MATLAB sobre el modelo de Simulink utilizado. Ahora introduciremos la Ecuación 3.17 en la Ecuación 3.13 para obtener una expresión en función de las velocidades del codo y de la muñeca, \dot{q}_2 y \dot{q}_3 :

$$\begin{aligned} \dot{V} &= \dot{q}_2 [\tau_2 (k_E \tilde{E} + k_D d_{22}^{inv}) - k_D (d_{22}^{inv} (\tau_2 - c_{22}\dot{q}_2 - c_{23}\dot{q}_3 - G_2 - F_{f2}) - \\ &\quad - d_{23}^{inv} (c_{32}\dot{q}_2 + c_{33}\dot{q}_3 + G_3 + F_{f3})) + k_P \tilde{q}_2] \end{aligned} \tag{3.18}$$

Proponemos una ley de control que nos lleve hasta:

$$\dot{V} = -\dot{q}_2^2 \tag{3.19}$$

Por tanto de las Ecuaciones 3.18 y 3.19 podemos determinar una ley de control:

$$\begin{aligned} -\dot{q}_2^2 &= \dot{q}_2 [\tau_2 (k_E \tilde{E} + k_D d_{22}^{inv}) - k_D (d_{22}^{inv} (\tau_2 - c_{22}\dot{q}_2 - c_{23}\dot{q}_3 - G_2 - F_{f2}) - \\ &\quad - d_{23}^{inv} (c_{32}\dot{q}_2 + c_{33}\dot{q}_3 + G_3 + F_{f3})) + k_P \tilde{q}_2] \end{aligned} \tag{3.20}$$

De donde despejaremos la expresión explícita de la ley de control que estamos buscando:

$$\tau_2 = \frac{k_D [d_{22}^{inv} (c_{22}\dot{q}_2 + c_{23}\dot{q}_3 + G_2 + F_{f2}) + d_{23}^{inv} (c_{32}\dot{q}_2 + c_{33}\dot{q}_3 + G_3 + F_{f3})] - k_P \tilde{q}_2 - \dot{q}_2}{k_E \tilde{E} + K_D d_{22}^{inv}} \tag{3.21}$$

Para simplificar la ley de control, reducirla ligeramente y aumentar su manejabilidad vamos a considerar:

$$F(q, \dot{q}) = d_{22}^{inv} (c_{22}\dot{q}_2 + c_{23}\dot{q}_3 + G_2 + F_{f2}) + d_{23}^{inv} (c_{32}\dot{q}_2 + c_{33}\dot{q}_3 + G_3 + F_{f3})$$

De esta manera nuestra ley de control se verá reducida a:

$$\tau_2 = \frac{k_D F(q, \dot{q}) - k_P \tilde{q}_2 - \dot{q}_2}{k_E \tilde{E} + K_D d_{22}^{inv}} \quad (3.22)$$

Es importante comprobar que nuestra ley de control no tendrá ninguna singularidad si se cumple la condición de que:

$$(k_E \tilde{E} + k_D d_{22}^{inv}) \neq 0 \Rightarrow k_E \tilde{E} \neq -\frac{k_D k_8}{(k_3 + k_6)k_8 - k_7 \cos^2 q_3} \quad (3.23)$$

Únicamente puede existir una singularidad si el denominador de d_{22}^{inv} se hace cero. Por tanto, la condición impuesta en la Ecuación 3.23 será satisfecha si:

$$(k_3 + k_6)k_8 - k_7 \cos^2 q_3 \neq 0$$

Si esto no ocurre para el valor máximo del coseno no ocurrirá nunca. Haciendo $\cos^2 q_3 = 1$ queda:

$$(k_3 + k_6)k_8 - k_7 > 0 \Rightarrow (0,0372 + 0,0151) \cdot 0,0044 - 0,0063^2 = 1,9043 \cdot 10^{-4} > 0 \quad (3.24)$$

Queda así demostrado que no existirán singularidades en nuestra ley de control que hagan que el par quede sin definir en alguna situación y desemboquen en errores del sistema.

3.2.3. Determinación de k_P , k_D y k_E

Hasta ahora no tenemos un controlador adecuado para nuestro sistema en particular. Simplemente hemos desarrollado de manera formal la aproximación y la ley de control que utilizaremos para conseguir nuestro objetivo. Así hemos evitado las singularidades en nuestra ley de control y, bajo las condiciones impuestas, llevaremos al Pendubot a la situación que perseguíamos. No debemos perder de vista que sobre el robot real necesitaremos reajustar los parámetros de nuevo.

El problema ahora reside en que para definir completamente el controlador de Swing Up debemos conocer los valores de k_P , k_D y k_E que hacen que el Pendubot converja hacia la órbita homoclina. Esto parece un problema trivial y realmente no lo es. Lo único que sabemos es que han de ser estrictamente positivos obligatoriamente.

En el primer artículo consultado (ver [FLS00]) explican como obtener la ley de control pero la única información que aportan acerca de como encontrar un valor de las constantes que funcione es que si uno elige una k_P “suficientemente cercana a cero” y una k_E “no muy pequeña” entonces $|\tilde{E}\tau_2|$ será pequeño. Lo cual es bastante poco preciso.

Para, al menos, tener unos valores iniciales desde los que ajustar el controlador se siguieron las pautas dadas en [XKO02] y así, después, hacer un ajuste mucho más fino a base

del método *ensayo - error* en la simulación. Se explica brevemente a continuación.

Se utilizan los resultados del *Teorema 1* presentados en [XKO02] para deducir unos valores iniciales desde los que partir para comenzar la simulación.

Si se cumplen las condiciones

$$V(0) \leq \frac{1}{2}k_E c_1^2 \quad (3.25)$$

$$c_1 := \text{mín}(2k_{10}, \frac{k_D - \epsilon}{k_E(k_3 + k_6)}) \quad (3.26)$$

Bajo la ley de control dada en la Ecuación 3.19, si

$$k_P > \eta^* k_E (k_5 + k_9)^2 \quad (3.27)$$

Entonces las siguientes relaciones se cumplen:

$$\lim_{t \rightarrow \infty} \tilde{E} = 0 \quad (3.28)$$

$$\lim_{t \rightarrow \infty} \tilde{q}_2(t) = 0 \quad (3.29)$$

$$\lim_{t \rightarrow \infty} V(t) = 0 \quad (3.30)$$

Esto quiere decir que bajo las condiciones 3.26 y 3.27 el Pendubot alcanzará asintóticamente la órbita homoclina que es el resultado que nosotros queremos conseguir. La prueba de este Teorema aparece en [XKO02]. Como punto de partida, utilizaremos unos parámetros pequeños y comprobaremos si servirían en nuestro caso:

$$k_P = 1,5, \quad k_D = 0,5, \quad k_E = 1, \quad \epsilon = 0,05, \quad \eta^* = 0,3146$$

Ahora:

$$c_1 := \text{mín}(2k_{10}, \frac{k_D - \epsilon}{k_E(k_3 + k_6)}) = \text{mín}(0,447, 8,604) = 0,447$$

$$V(0) \leq \frac{1}{2}k_E c_1^2 \Rightarrow k_E E_{eq}^2 + k_P \pi^2 \leq k_E c_1^2 \Rightarrow 21,06 \leq 0,0999 \quad \# \text{No cumple} \quad (3.31)$$

No cumple las condiciones iniciales para poder aplicar el resultado, no obstante utilizaremos la relación 3.27 que no te asegura que vayan a funcionar si no cumplen las hipótesis (aunque pueden hacerlo) para hallar unos valores de los que partir.

$$k_P > \eta^* k_E (k_5 + k_9)^2 \Rightarrow k_P > k_E \cdot 0,3146 (0,495 + 0,5322)^2 \Rightarrow k_P > 0,332 k_E \Rightarrow 1,5 > 0,332 \cdot 1$$

En nuestras simulaciones partiremos de estos valores para luego ir probando manualmente otros valores buscando mejores resultados, es decir, un Swing Up más rápido. k_D ni siquiera aparece en la relación por eso es libre, podría tomar cualquier valor, aunque elegimos un valor más pequeño que las otras dos constantes para que se asemeje a ellas. Por tanto,

$$k_P = 1,5, \quad k_D = 0,5, \quad k_E = 1$$

3.3. Ballancing Control

3.3.1. Objetivo

Una vez hemos conseguido meter al Pendubot dentro de la órbita homoclina éste, eventualmente, entrará en la *base de atracción* de la segunda fase de nuestro controlador, el *Ballacing Control*. La base de atracción es el conjunto de valores límite de q_2 , q_3 , \dot{q}_2 y \dot{q}_3 para los que el controlador es capaz de estabilizar el sistema en el equilibrio inestable.

Para realizar esta tarea hemos seguido el consejo de los artículos [FLS00] y [XKO02] que recomiendan el uso de un controlador lineal cuadrático conocido como *LQR*. Además tenemos la ventaja de que una vez conocido el modelo del sistema el propio MATLAB calcula los valores directamente con una función nativa.

3.3.2. Desarrollo teórico

El controlador hace una realimentación de las variables de estado, $u = -Kx$, de manera que el coste de la ley de control es mínimo. No entraremos en profundidad pero se calcula a partir de la siguiente expresión:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

La implementación de este controlador es trivial puesto que se utiliza una ganancia calculada como se acaba de indicar. Cada sistema tiene su propio espacio de estados por eso existe un único controlador para cada sistema.

Para hallar los valores del controlador LQR necesitamos:

1. Linealizar el sistema en torno al punto de equilibrio donde queremos que actúe.
2. Expresar nuestro modelo en el *espacio de estados*.

Linealizar las dos ecuaciones del movimiento en torno a la posición de equilibrio vertical, $P_{eq} = (q_2, \dot{q}_2, q_3, \dot{q}_3) = (\pi, 0, 0, 0)$, es tarea sencilla mediante los desarrollos en serie de Taylor, esto es:

$$f(x, y) \approx f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b) + \frac{1}{2} [f_{xx}(a, b)(x - a)^2 + 2f_{xy}(a, b)(x - a)(y - b) + f_{yy}(a, b)(y - b)^2] \dots$$

Viendo las funciones que tenemos dentro de nuestras ecuaciones de movimiento tomaremos directamente las siguiente simplificaciones:

$$\begin{array}{ll} \cos(q_2)|_{\pi} = -1, & \text{sen}(q_2)|_{\pi} = -q_2 \\ \cos(q_3)|_0 = 1, & \text{sen}(q_3)|_0 = q_3 \\ \cos(q_2 + q_3)|_{(\pi, 0)} = -1, & \text{sen}(q_2 + q_3)|_{(\pi, 0)} = -(q_2 + q_3) \\ \dot{q}_2|_0 = 0 & \dot{q}_3|_0 = 0 \end{array}$$

Por tanto las Ecuaciones 2.2 y 2.3 linealizadas en torno a P_{eq} quedan:

$$[k_3 + k_6 + k_8 + 2k_7] \ddot{q}_2 + [k_7 + k_8] \ddot{q}_3 - (k_5 + k_9)q_2 - k_{10}(q_2 + q_3) + F_{f2} = \tau_2 \quad (3.32)$$

$$[k_7 + k_8] \ddot{q}_2 + [k_8] \ddot{q}_3 + k_{10}(q_2 + q_3) = 0 \quad (3.33)$$

De la Ecuación 3.33 podemos despejar \ddot{q}_2 y \ddot{q}_3 :

$$\ddot{q}_2 = \frac{1}{k_7 + k_8} (k_{10}(q_2 + q_3) - k_8 \ddot{q}_3) \quad (3.34)$$

$$\ddot{q}_3 = \frac{k_{10}(q_2 + q_3) - [k_7 + k_8] \ddot{q}_2}{k_8} \quad (3.35)$$

Y utilizar estas expresiones para sustituirlas en la Ecuación 3.32 y hallar una función lineal de la aceleración dependiendo únicamente de las posiciones de codo y muñeca. Obtenemos nuestro sistema linealizado:

$$\ddot{q}_2 = \frac{(\tau_2 - F_{f2}) - q_2 k_{q22} - q_3 k_{q32}}{k_{q2pp}} \quad (3.36)$$

$$\ddot{q}_3 = \frac{(\tau_2 - F_{f2}) - q_2 k_{q23} - q_3 k_{q33}}{k_{q3pp}} \quad (3.37)$$

Donde

$$k_{q22} = \left[\frac{k_{10}(k_7 + k_8) - k_8(k_5 + k_9 + k_{10})}{k_8} \right] \quad (3.38)$$

$$k_{q32} = \left[\frac{k_{10}(k_7 + k_8) - k_8 k_{10}}{k_8} \right] \quad (3.39)$$

$$k_{q2pp} = \left[\frac{k_8(k_3 + k_6 + k_8 + 2k_7) - (k_7 + k_8)^2}{k_8} \right] \quad (3.40)$$

$$k_{q23} = \left[\frac{k_{10}(k_3 + k_6 + k_8 + 2k_7) - (k_5 + k_9 + k_{10})(k_7 + k_8)}{k_7 + k_8} \right] \quad (3.41)$$

$$k_{q33} = \left[(k_7 + k_8) - \frac{k_8(k_3 + k_6 + k_8 + 2k_7)}{k_7 + k_8} \right] \quad (3.42)$$

$$k_{q3pp} = \left[\frac{k_{10}(k_3 + k_6 + k_8 + 2k_7)}{k_7 + k_8} - k_{10} \right] \quad (3.43)$$

El modelo representado según su espacio de estados tiene la forma:

$$\dot{x} = Ax + Bu$$

Siendo x el vector de estados que representa las coordenadas necesarias para definir un sistema y su estado. Podemos utilizar las expresiones anteriores para hallar el espacio de estados de nuestro Pendubot:

$$\begin{bmatrix} \dot{q}_2 \\ \ddot{q}_2 \\ \dot{q}_3 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{k_{q22}}{k_{q2pp}} & 0 & \frac{k_{q32}}{k_{q2pp}} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_{q23}}{k_{q3pp}} & 0 & \frac{k_{q33}}{k_{q3pp}} & 0 \end{bmatrix} \begin{bmatrix} q_2 \\ \dot{q}_2 \\ q_3 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{k_{q2pp}} & 0 \\ 0 & 0 \\ \frac{1}{k_{q3pp}} & 0 \end{bmatrix} \begin{bmatrix} \tau_2 \\ 0 \end{bmatrix} \quad (3.44)$$

Aquí vemos los valores que tienen las matrices A y B :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 16,3400 & 0 & -7,3941 & 0 \\ 0 & 0 & 0 & 1 \\ 11,0595 & 0 & 68,7765 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 0 \\ 23,1056 & 23,1056 \\ 0 & 0 \\ -56,1886 & -56,1886 \end{bmatrix} \quad (3.45)$$

Una vez obtenidas las matrices del espacio de estados solo queda el paso más fácil de todos. Utilizaremos la función “*lqr*” de MATLAB que calcula automáticamente los valores de las ganancias necesarias para utilizar este controlador de manera óptima. Las entradas que necesita esta función son las matrices A y B del espacio de estados, el tiempo de muestreo de nuestro sistema $\delta = 0,001\text{seg}$ y una matriz de ponderación Q para la cual se ha usado una matriz identidad de 4×4 .

$$K_{LQR} = \begin{pmatrix} k_{q2} \\ k_{q2} \\ k_{q3} \\ k_{q3} \end{pmatrix} = \begin{pmatrix} -17,0010 \\ -3,7563 \\ -16,4435 \\ -2,3801 \end{pmatrix} \quad (3.46)$$

Se aprecia que todos los valores son negativos. No importa el signo que tengan las ganancias en este tipo de controladores. Solamente significa que la señal se invertirá o no. Puede modificarse, sin tener ninguna consecuencia en el comportamiento del sistema, al variar la manera de *montar* el modelo de Simulink que se usa en los experimentos.

3.4. Switcher

3.4.1. Objetivo

El objetivo de este dispositivo es cambiar del primer controlador, *Swing Up*, al segundo controlador, *Ballancing Control*, cuando ciertas condiciones se cumplan. Es básicamente un interruptor que decide que regulador funciona mejor en cada momento.

3.4.2. Desarrollo teórico

Los valores que se utilizan para tomar la decisión son aquellos que definen la base de atracción del controlador LQR aunque, por asegurarnos, no utilizaremos los valores que marcan el límite de la zona de actuación sino otros que estén completamente dentro y sepamos que el controlador puede funcionar exitosamente.

Es importante saber si la primera fase del controlador, el *Swing Up*, es capaz de hacer que el Pendubot entre por sí mismo en la base de atracción porque si esto no es así la segunda parte no funcionará jamás puesto que no es capaz de “compensar” las perturbaciones como para traerlo hasta el equilibrio.

Para la realización práctica se implementan cuatro constantes con los valores de referencia para comparar con las variables posición y velocidad de codo y muñeca. Cuando la variable, en valor absoluto, es menor que el valor de referencia significa que esa condición se cumple y se convierte en “*true*” su variable booleana asociada. Cuando las cuatro variables

booleanas asociadas son “*true*” entonces el interruptor conmuta.

Hay que remarcar un aspecto relevante de la conmutación de controladores. Ambos están calculados por separado lo que implica que no existe ninguna continuidad entre ellos. Esto quiere decir que una vez dentro de la base de atracción, para los mismos valores de posición y velocidad de codo y muñeca, cada controlador calculará un valor del par diferente. Teniendo en cuenta que la conmutación se realiza de manera prácticamente instantánea nos encontramos con un problema.

En el momento exacto del cambio existirá una discontinuidad en el par calculado. Normalmente esta singularidad en el par producirá un pico o un impulso que podría incluso hacer que el Pendubot se saliera de la zona de actuación del controlador LQR porque no siempre es capaz de compensarla. Si el salto no es muy grande el propio sistema será capaz de absorberlo y compensarlo por si mismo.

Se impusieron ciertos valores de referencia que, *a priori*, parecían lógicos aunque luego se variaron para ajustarlo a un comportamiento más suave y fino del Pendubot. Estos valores son:

$$\begin{aligned}
 q_2^{ref} = 15^\circ = 0,2618rad &\Rightarrow |q_2 - q_2^{eq}| \leq q_2^{ref} \Rightarrow |q_2 - \pi| \leq 0,2618 \\
 \dot{q}_2^{ref} = 0,8rad/s &\Rightarrow |\dot{q}_2 - \dot{q}_2^{eq}| \leq \dot{q}_2^{ref} \Rightarrow |\dot{q}_2 - 0| \leq 0,8 \\
 q_3^{ref} = 15^\circ = 0,2618rad &\Rightarrow |q_3 - q_3^{eq}| \leq q_3^{ref} \Rightarrow |q_3 - 0| \leq 0,2618 \\
 \dot{q}_3^{ref} = 2,5rad/s &\Rightarrow |\dot{q}_3 - \dot{q}_3^{eq}| \leq \dot{q}_3^{ref} \Rightarrow |\dot{q}_3 - 0| \leq 2,5
 \end{aligned}$$

Capítulo 4

Simulación

4.1. Introducción

El propósito de este capítulo es explicar el conjunto de simulaciones que se realizaron y los resultados que se obtuvieron aplicando el modelo matemático hallado e identificado en apartados anteriores. Se mostrará también el diagrama de bloques que se construyó en Simulink para realizar todas las simulaciones así como sus peculiaridades. Al final se verán los resultados conseguidos y las conclusiones que se sacaron a la luz de los resultados.

4.2. Diagrama de bloques

El modelo que se ha utilizado para los experimentos de simulación es el que aparece en la Figura 4.1. En él se distinguen claramente dos bloques principales. El bloque azul de la izquierda que tiene dentro todo el subsistema del controlador sin ajustar y el bloque gris de la parte derecha que contiene todo el modelo matemático calculado para el robot. A continuación se mostrará el interior de cada uno. El resto de bloques que aparecen son medidores o instrumentos para poder llevarnos el valor de las variables almacenadas al *workspace* de MATLAB y poder trabajar con ellas.

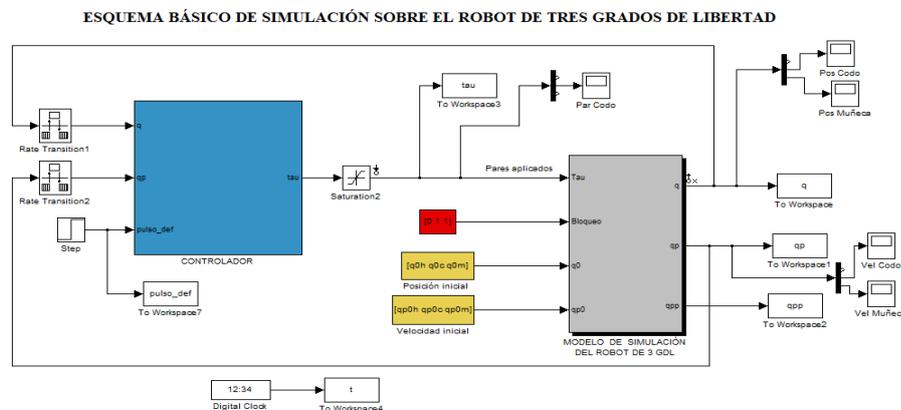


Figura 4.1: Esquema de simulación

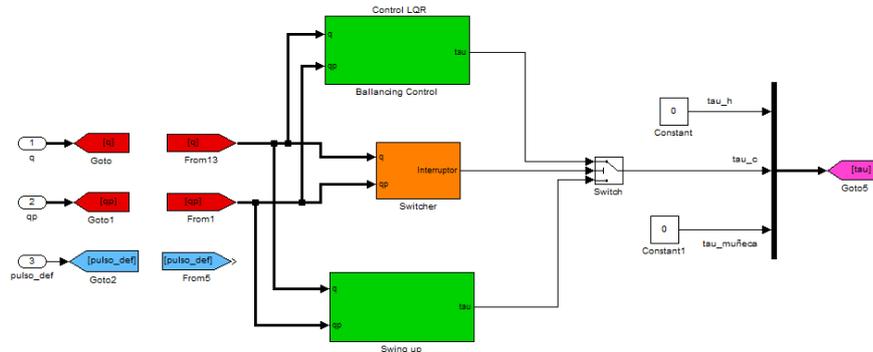


Figura 4.2: Interior del bloque Controlador

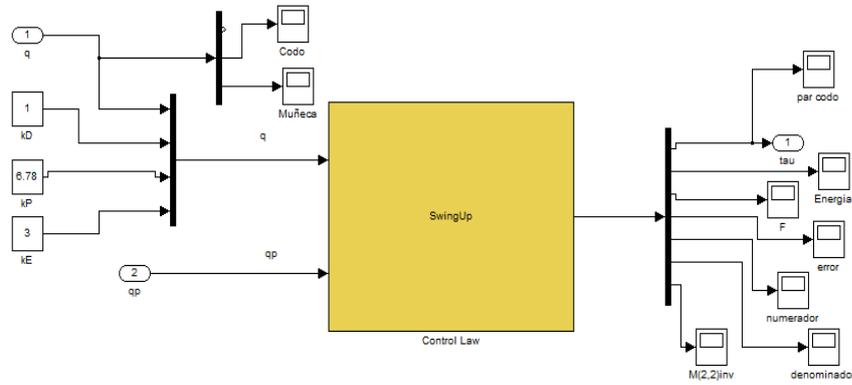


Figura 4.3: Interior del bloque Swing Up

4.2.1. Bloque: Controlador

Aquí se encuentran las dos etapas del controlador, el Swing Up y el Ballancing Control como podemos apreciar en la Figura 4.2. Además se ha añadido un interruptor, el Switcher, que se encarga de monitorizar los valores de las posiciones y velocidades y llegado el momento hace que deje de funcionar el Swing Up para comenzar a controlar el sistema mediante el Ballancing Control.

El bloque Swing Up lleva dentro una función embebida programada en lenguaje C. Se hizo porque el cálculo de matrices necesario a cada paso de la simulación se realiza más rápidamente con una función de este tipo. A esto le añadimos que manejar tal cantidad de operaciones y matrices es más cómodo en lenguaje programado que mediante bloques de Simulink. No obstante, los tres parámetros k_P , k_D y k_E que se debían ajustar se sacaron fuera para tratarlos como ganancias constantes y pasándole el dato de entrada a la función aumentando así la manejabilidad del modelo. Se ve lo que contiene en la Figura 4.2.

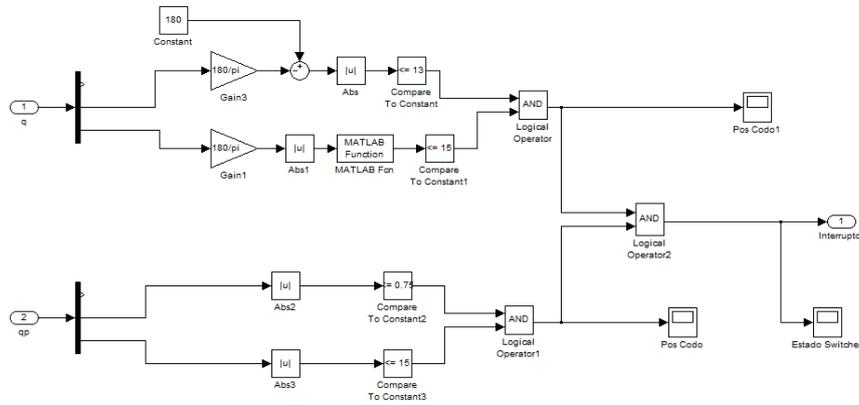


Figura 4.4: Interior del bloque Switcher

El bloque Switcher es un interruptor que decide que controlador actúa en cada momento en base a unos ciertos parámetros de corte. Se puede observar que los cuatro parámetros que monitoriza son: q_2 , \dot{q}_2 , q_3 y \dot{q}_3 . Cuando el pendubot se encuentra en torno a la posición de equilibrio y a velocidades suficientemente bajas el switcher se activa. El valor de estos parámetros aparece en el apartado 3.4.

Cabe resaltar un detalle. La posición de la muñeca en ocasiones superaba la vuelta completa, $q_3 > 2\pi$ lo que hacía que aún estando próxima a la situación de equilibrio el Ballancing Control no se activara nunca. Para solucionar este pequeño problema se normalizó la posición limitándola únicamente al rango $(0, 2\pi)$. En la práctica se introdujo una función *mod* que devuelve el resto de una división, acotando así la medida de la posición de la muñeca.

El bloque Ballancing Control es la implementación del controlador LQR que consiste en alimentar las variables de estado a través de una ganancia calculada mediante el método que minimiza el coste, explicado en el Capítulo 5, en el apartado correspondiente.

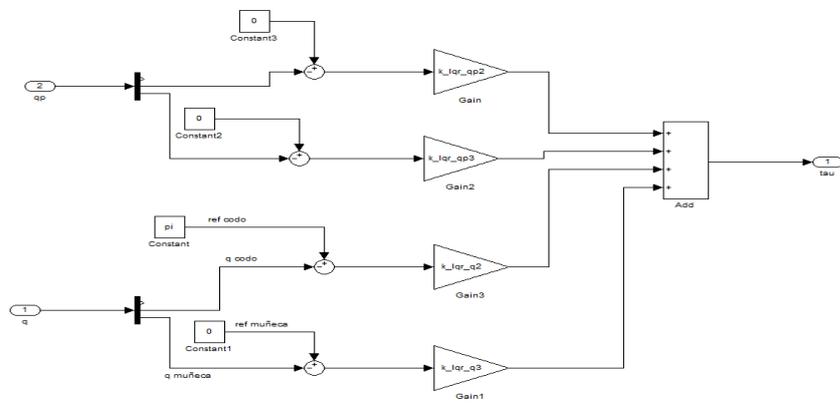


Figura 4.5: Interior del bloque Ballancing Control

4.2.2. Bloque: Modelo de Simulación

Aquí se encuentra todo el modelo matemático que hemos obtenido anteriormente. Para producir un movimiento en el robot se calculaba un par de entrada para el codo, τ_2 , ya que el par en la muñeca, τ_3 , es cero por definición al tratarse de un sistema subactuado. Para apreciar las similitudes en el esquema recordaremos que la forma que tenía el modelo:

$$\begin{bmatrix} \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} d_{22} & d_{23} \\ d_{32} & d_{33} \end{bmatrix}^{-1} \left(\begin{bmatrix} \tau_2 \\ 0 \end{bmatrix} - \begin{bmatrix} c_{22} & c_{23} \\ c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} - \begin{bmatrix} G_2 \\ G_3 \end{bmatrix} - \begin{bmatrix} F_{f2} \\ F_{f3} \end{bmatrix} \right) \quad (4.1)$$

Cada una de las diferentes matrices de la ecuación aparece reflejado en un bloque amarillo. Estos bloques son funciones programadas en C, como también lo es el controlador, puesto que el manejo de las matrices es más rápido, sencillo y eficaz que teniendo que tratar cada vector por separado.

La salida del bloque “3gdl_m_inv_param_chsel” nos entrega las aceleraciones de codo y muñeca. A partir de esa señal se han colocado dos integradores consecutivos para poder disponer de las señales de velocidad y posición de las articulaciones. Estos integradores tienen incluida una entrada para especificar las condiciones iniciales del experimento, de hecho, el robot no tendría porque empezar de la situación de reposo.

Existe una entrada más a todo el sistema del robot. Nos referimos a la entrada “BLK” que representa el bloqueo que se le podía ejercer a la articulación del hombro y del codo. Utilizando esta herramienta era como simulábamos que estaban bloqueadas y no tenían movimiento. Si el codo se bloqueaba tendría la misma posición que el hombro y si, por el contrario, se bloqueaba el hombro, éste actuaría de bancada y el movimiento del codo sería completamente independiente.

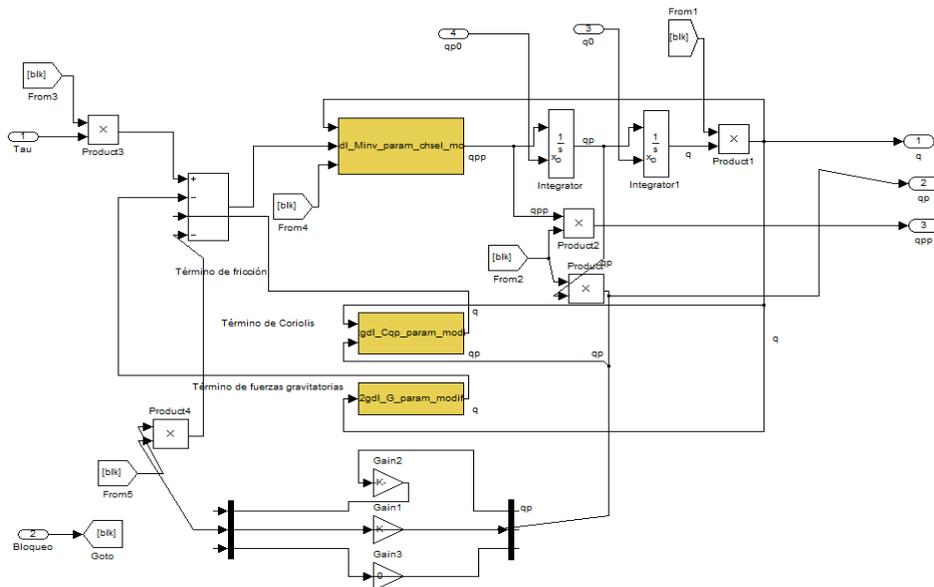


Figura 4.6: Interior del bloque Modelo del Robot de 3 GDL

4.3. Resultados

El objetivo de la simulación del comportamiento del pendubot es conseguir que las dos partes del controlador funcionen correctamente, incluyendo el momento de cambio gobernado por el estado del switcher y la situación de discontinuidad que crea el salto en el valor del par. Más allá se persigue y se desea que, tras simular el comportamiento, los valores ajustados del controlador se puedan exportar al sistema real, manteniendo en mente que los valores finales deberán ser aún ligeramente modificados.

En las Figuras 4.7 y 4.8 se puede examinar el resultado final del ajuste del controlador. Se alcanza un comportamiento deseado aunque se sospecha que puede existir otros valores que mejoran el tiempo de respuesta final del pendubot. Al fijarnos en la posición de la muñeca se observa que entra en la órbita homoclina y que, una vez dentro, oscila varias veces hasta entrar dentro de la cuenca de atracción que delimita el cambio de controlador.

Los valores para los que se consiguió este resultado con el controlador fueron: $k_P = 6,78$, $k_D = 1$ y $k_E = 3$.

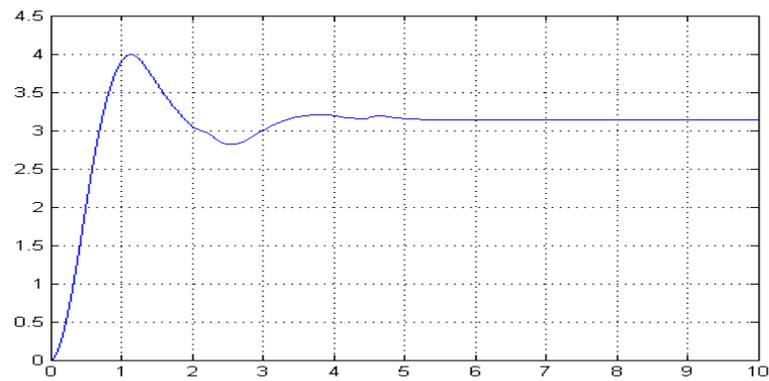


Figura 4.7: Evolución de la posición del codo con el tiempo

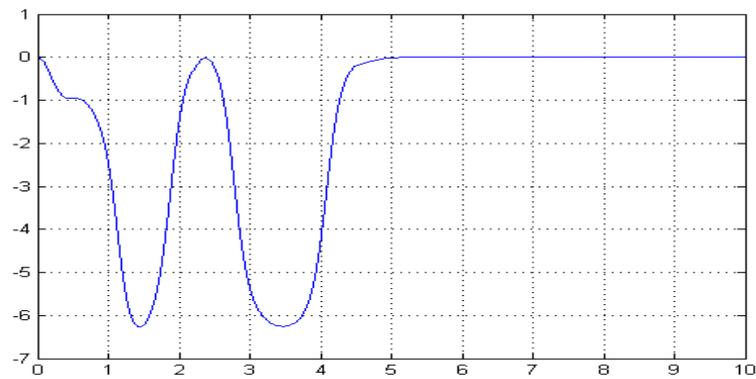
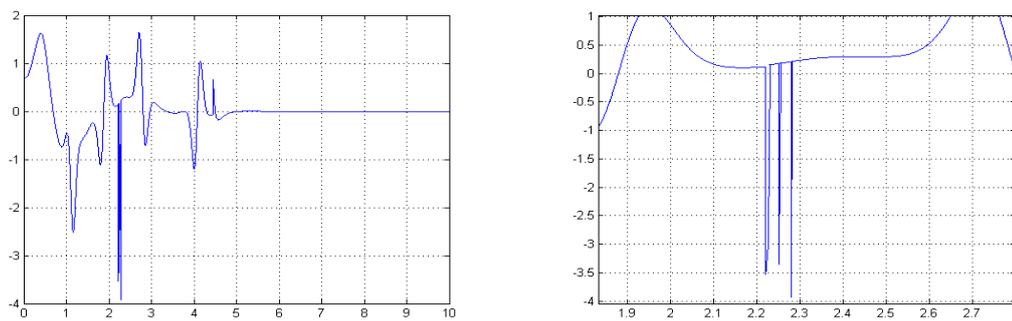


Figura 4.8: Evolución de la posición de la muñeca con el tiempo

La ley de control calculada y ajustada nos ha llevado al cálculo de un par, medido en voltios y que nunca podía superar los 10V ya que era el punto de saturación del sistema, con la forma que nos muestra la Figura 4.9(a). La única referencia que teníamos de la evolución del par era la vista anteriormente en los artículos [FLS00] y [XKO02]. El parecido es elevado lo que, lejos de asegurarnos nada, parece indicar que la senda es la correcta.

En lo referente al par si que es importante reseñar la pequeña discontinuidad que aparece en la entrada en el momento en que el switcher cambia del Swing Up al Ballancing Control. De hecho, en la Figura 4.9(b) se observa que el interruptor no actúa una única vez sino que entra en acción tres veces antes de cambiar definitivamente al Ballancing Control. Esto es debido a que el pendubot se encuentra en la frontera de la cuenca de atracción entrando y saliendo hasta que definitivamente cae dentro.

La variación que tiene el par es suficientemente pequeña como para no sacar al pendubot de su trayectoria por lo que los inconvenientes han sido mínimos a la hora de tratarlo. El switcher debería estar ajustado de tal manera que solamente se ejecutara una vez aunque no fue posible hallar los valores límite que hicieran que ocurriera así.



(a) Evolución del par de entrada al codo

(b) Detalle del par en el momento de cambio de controladores

Figura 4.9: Gráficas de la evolución del par del codo con el tiempo

Si revisamos las Figuras 4.7 y 4.8 vemos que el cambio de controlador no afecta a la posición de ninguna de las dos articulaciones. En este sentido el sistema absorbe la pequeña perturbación de manera satisfactoria. Sin embargo si que puede apreciarse un cambio en la velocidad de la muñeca. No obstante el pico de velocidad no es muy elevado tampoco. Se pueden revisar las Figuras 4.10(a) y 4.10(b) para ver hasta que punto les ha afectado.

En la Figura 4.11 aparece representado un detalle de la variación de la velocidad articular de la muñeca en el instante en que se activa el switcher. Existen tres pequeños picos antes de que la velocidad vuelva a lo que parece su trayectoria natural.

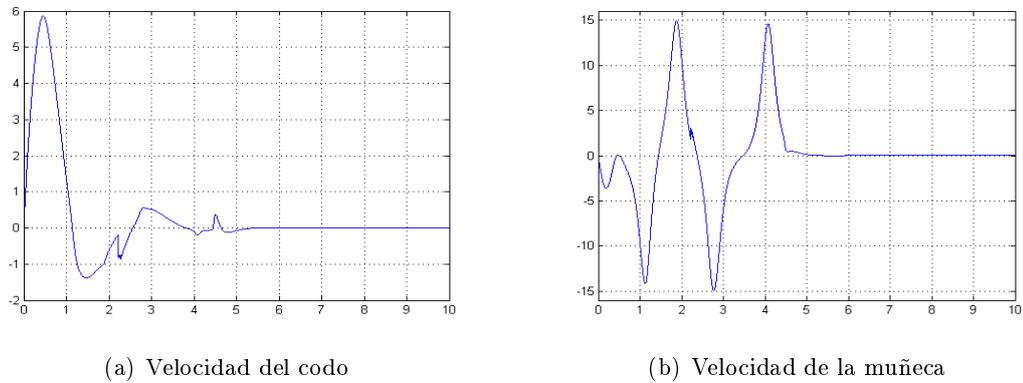


Figura 4.10: Evolución de las velocidades articulares con el tiempo

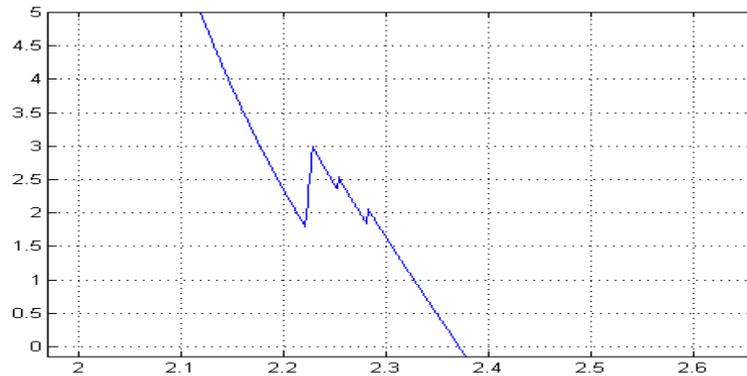


Figura 4.11: Velocidad de la muñeca en el instante de activación del switcher

4.4. Conclusión

A la vista de los resultados podemos afirmar que las simulaciones han obtenido una respuesta muy satisfactoria. El tiempo de respuesta del robot es relativamente pequeño. Llega al punto de equilibrio superior inestable en un tiempo reducido de aproximadamente $t_r = 5$ segs. Se sospecha que un ajuste incluso más rápido es posible en el que el pendubot llegue a estabilizarse en el punto más alto justo cuando la muñeca realiza la primera oscilación. Se cree que podría conseguirse ya que cumple, al menos, las condiciones de posición de codo y muñeca pero las condiciones de velocidad deberían ser ligeramente más laxas lo que podría hacer que no entrara en la cuenca de atracción o que entrara pero no fuera capaz de permanecer dentro.

En esta sección se ha conseguido utilizar el controlador de dos etapas con Swing Up y Ballancing Control para estabilizar el robot en una situación de doble péndulo invertido inestable utilizando el modelo matemático hallado e identificado en apartados anteriores. A continuación se procederá a probar si al exportar los valores a la plataforma real el resultado es positivo.

Capítulo 5

Experimentación

5.1. Introducción

En este apartado se pretende explicar el conjunto de experimentos que se llevaron a cabo así como sus resultados. Los parámetros ajustados obtenidos en simulación se exportaron a la plataforma real. Debido a la incertidumbre que siempre existe en los sistemas reales se esperaba que los valores del controlador tuvieran que ser ajustados para un correcto comportamiento del pendubot. A continuación se explicará la respuesta que se obtuvo.

5.2. Diagrama de bloques para experimentación

A lo largo de las etapas de simulación y experimentación se ha tratado de llevar en paralelo la similitud de los modelos con el fin de facilitar el entendimiento y el manejo de los mismos. Habituarse a la utilización de uno y el entendimiento de ambos. En la Figura 5.1 se puede ver el diagrama de bloques general que se ha utilizado para los experimentos con el pendubot. También hay multitud de salidas hacia el *workspace* de Matlab que han servido para monitorizar las variables que más influencia tenían sobre el comportamiento del robot.

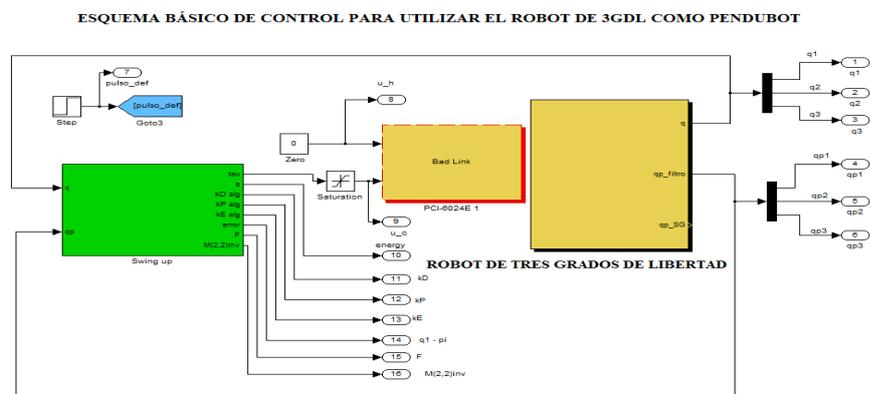


Figura 5.1: Diagrama de bloques de Simulink utilizado para los experimentos

Tal y como hemos mencionado, en el interior de los bloques “Controlador” y “Robot de tres grados de libertad” se puede apreciar el mismo tipo de esquema que vimos en simulación, véanse las Figuras 5.2(a) y 5.2(b). La diferencia más apreciable aparece en el interior del bloque robot real donde el modelo ha desaparecido para dejar paso al mecanismo del robot en sí mismo. Los bloques que aparecen representan las tarjetas de adquisición de datos que nos abastecerán con las medidas de posición de las articulaciones.

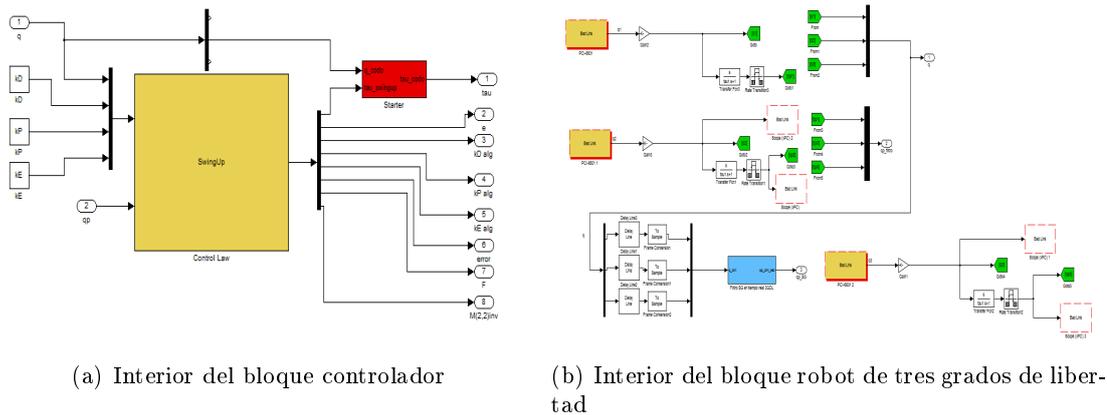


Figura 5.2: Interior de los bloques controlador y robot

5.3. Experimentos

Se realizaron multitud de baterías de experimentos hallando, finalmente, una colección de valores que funcionaban de manera adecuada sin llegar a conseguir una respuesta del todo satisfactoria. Como podemos observar en las Figuras 5.3 y 5.4, el sistema consigue entrar en la órbita homoclina, paso previo a conseguir un cambio de controlador, y caer en la atracción de la cuenca de control del Ballancing Control.

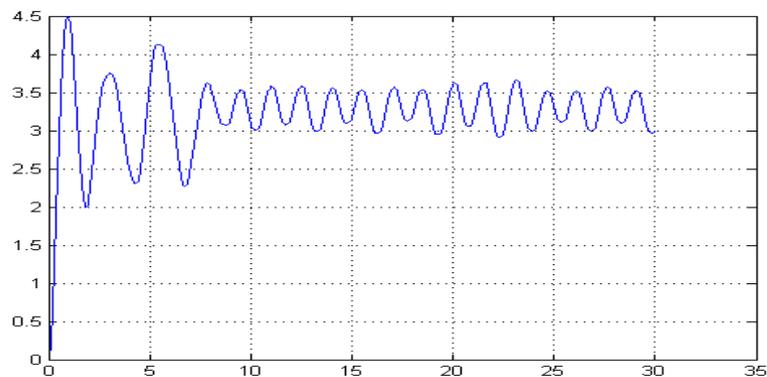


Figura 5.3: Trayectoria del codo

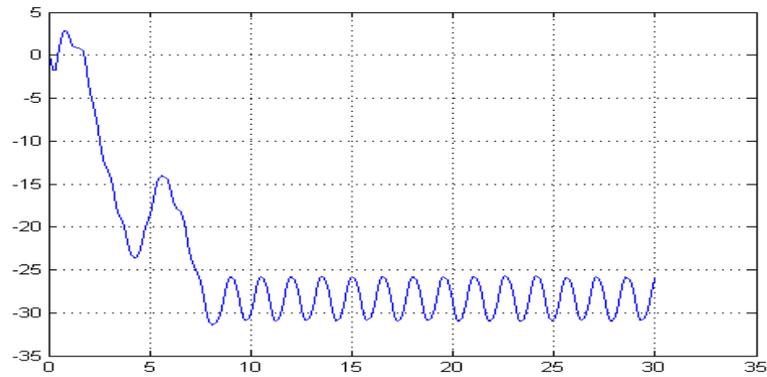
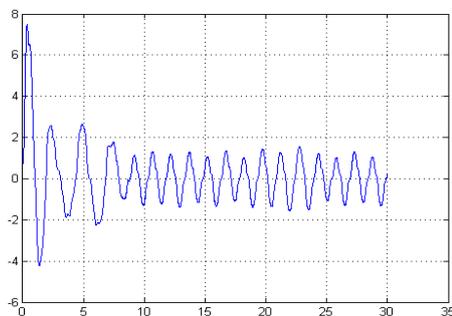


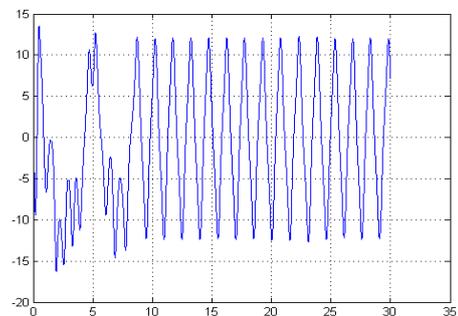
Figura 5.4: Trayectoria de la muñeca

Los valores de los parámetros utilizados en este experimento fueron: $k_P = 3,5$, $k_D = 0,63$ y $k_E = 6$. Si bien estos valores se parecen a los ajustados en la simulación, resulta que no están asignados a las mismas variables. Este hecho se podría deber a que los términos de mayor relevancia en el modelo son diferentes en la simulación y en la realidad. No obstante, se propone un estudio posterior de la dependencia del modelo con estos valores para hallar la verdadera causa.

En la experimentación con la plataforma real apareció un comportamiento que en la simulación nunca llegó a darse. Con el impulso inicial que proporcionaba el codo, la muñeca comenzaba a girar dando más de una vuelta hasta que llegaba a un punto en el que el codo conseguía “estabilizarla” en un entorno de la posición de equilibrio. Es decir, la muñeca giraba libremente hasta que entraba en la órbita homoclina y ahí se quedaba oscilando como se ve en la Figura 5.4, primero aproximadamente da nueve vueltas y después comienza a oscilar. En comparación con los comportamientos exhibidos en simulación son bastante similares a excepción del número de vueltas que gira la muñeca antes de estabilizarse.



(a) Velocidad del codo



(b) Velocidad de la muñeca

Figura 5.5: Evolución de las velocidades articulares

La comparación entre velocidades articulares es bastante positiva. Tanto la velocidad del codo como la de la muñeca es bastante similar en forma y valores. Si bien es cierto que la muñeca adquiere una menor velocidad en experimentación de la que alcanzaba en simulación. Este pequeño detalle se puede deber a que se desprecia la fricción de la muñeca y, por ínfima que sea, siempre habrá un mínimo que podría ser la razón por la que vemos esta diferencia.

El valor del codo que vemos en la Figura 5.6 está acotado entre $(-3,5, 4,5)$ lo cual es una buena señal puesto que el sistema satura a más de 10V, aunque para un correcto control no debería ser necesario superarlos. La tendencia que se aprecia es decreciente. Ésto concuerda con la idea del controlador que hemos desarrollado. Se entrega poco a poco par al codo y va decreciendo paulatinamente hasta, teóricamente, quedar a cero cuando el sistema se sitúe en su posición de equilibrio.

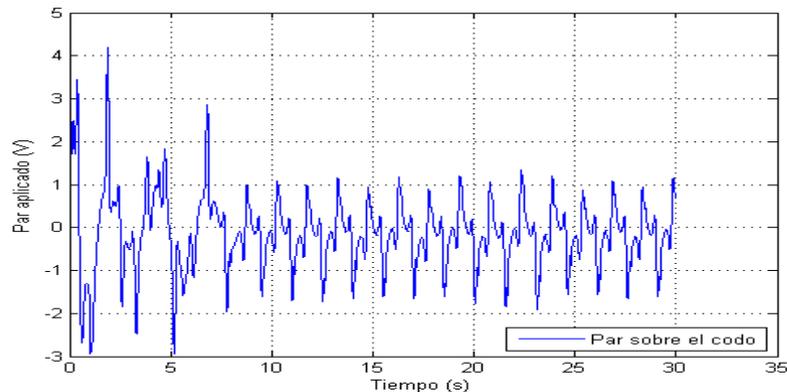


Figura 5.6: Evolución del par del codo

A continuación se muestran los resultados de la realización del mismo experimento que antes pero apenas tres horas más tarde. Solo este desfase temporal hizo que el comportamiento del robot fuera absolutamente diferente. El sistema se inestabilizó a los pocos segundos del comienzo del experimento.

Se puede comprobar fácilmente en la Figura 5.7 donde aparece la representación de las velocidades articulares (Figura 5.7(a)) y el par del codo (Figura 5.7(b)). Si nos fijamos en los valores del par del codo, oscilan entre 40V y 60V lo cual es imposible porque el sistema solo permite un rango entre -10V y +10V. Esta medida está tomada antes de llegar al sistema por eso aparecen semejantes valores. Nos da una idea de la cantidad de energía entregada al sistema y la velocidad a la que se movería si no se parara.

Por otra parte la muñeca llega a adquirir picos de velocidad de 120rad/s lo que quiere decir que estaría dando, aproximadamente, 20 vueltas por segundo que es una velocidad muy elevada. Desde ese punto no puedes recuperar el control del sistema hasta que no vuelve a una situación de reposo y empiezas el experimento de nuevo.

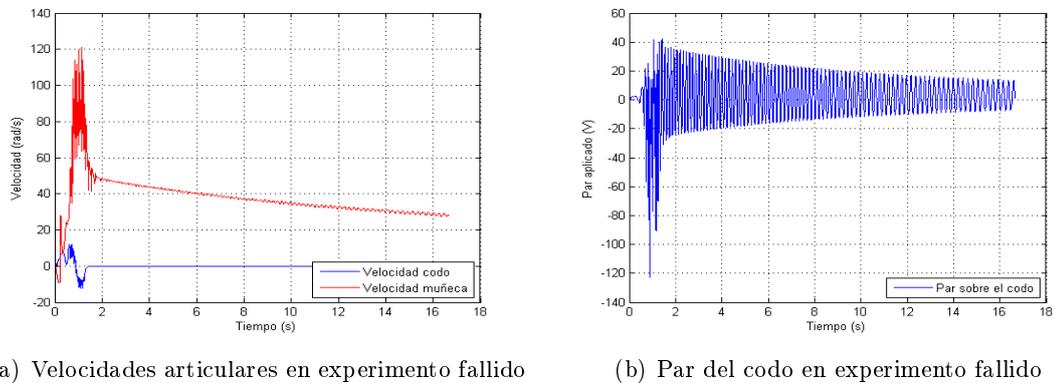


Figura 5.7: Velocidades articulares y par del codo en experimento fallido

La última figura del experimento fallido nos muestra la trayectoria tanto de codo como de muñeca. De esta gráfica se desprende que se apretó la seta de emergencia para parar el codo de golpe. La muñeca al no tener ninguna acción sobre ella siguió girando hasta llegar al reposo. Se puede ver que durante el tiempo que duró el experimento no había dejado de girar. Debido al escalado de la gráfica no se aprecia pero el codo permaneció inmóvil después de accionar el freno, como era de esperar.

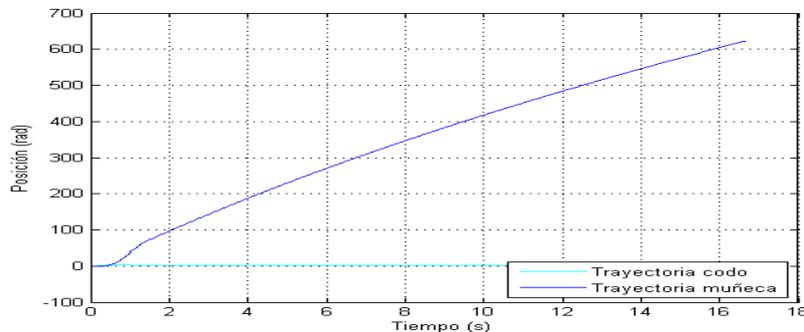


Figura 5.8: Evolución de las posiciones articulares en experimento fallido

5.4. Hipótesis

Se propone a continuación una hipótesis sobre la causa que puede la razón de esta variabilidad que ha aparecido en los resultados. Puede que tenga su origen en el cambio de temperatura que sufre la pared sobre la que está anclado el robot. Ésta recibe la radiación solar de pleno a lo largo de la mañana. La fricción, que puede tener una dependencia acusada de la temperatura, puede variar hasta el punto de modificar sustancialmente el comportamiento predicho en secciones anteriores.

No obstante se precisaría un estudio centrado en esta hipótesis para validar o refutar la idea que aquí se ha propuesto.

Capítulo 6

Conclusiones

Se pretende en este capítulo final explicar las conclusiones que se han desprendido de la realización de este proyecto. Abarcan fundamentalmente tres aspectos diferentes dentro del trabajo comprendido a lo largo de este tiempo.

6.1. Modelado e identificación

Se ha obtenido un modelo matemático de la mecánica del robot siguiendo el procedimiento de las ecuaciones de *Euler-Lagrange*. Se trata de un procedimiento sistemático que permite obtener las ecuaciones dinámicas para cualquier sistema mecánico de cadena abierta, como es nuestro caso. Estas ecuaciones asumen sin embargo una serie de hipótesis para su obtención que pueden satisfacerse en mayor o menor grado en función del tipo de sistema mecánico para el que se apliquen. En el caso particular de la robótica y para el caso que nos ocupa, las hipótesis más relevantes serían las siguientes:

- **Hipótesis de sólido rígido:** Las ecuaciones asumen que los diferentes enlaces del robot son sólidos rígidos ideales, indeformables. Si bien la mayoría de los robots se diseñan para satisfacer esta hipótesis, no es descartable la influencia de deformaciones estructurales o la aparición de modos de vibración asociados al seguimiento de ciertos tipos de trayectorias.
- **Hipótesis de articulación perfecta:** Se asume que las articulaciones se comportan de modo ideal. En la práctica, fenómenos como las holguras, fricciones, etc, influyen notablemente en la dinámica del sistema.
- **Hipótesis de actuador perfecto:** Se asume que los actuadores, motores eléctricos brushless (sin escobillas) en nuestro caso, proporcionan en todo momento un par motor conocido y controlable. Esta hipótesis, aunque razonablemente aproximada, presenta dificultades debido a los lazos de control internos propios de estos motores.
- **Hipótesis de modelo cinemático conocido:** Se asumen que el modelo cinemático del sistema, que involucra los parámetros geométricos del robot, es perfectamente conocido. En la práctica, inexactitudes propias de los procedimientos de medida están siempre presentes.

- **Hipótesis de sensores ideales:** Se asume que los sensores de posición y velocidad articular proporcionan medidas fiables y exactas. En la prácticas, y especialmente en el caso de los sensores de velocidad, aparecen inexactitudes que afectan al comportamiento del sistema. Por otra parte, en caso de que se precise de una identificación de parámetros en nuestro sistema de trabajo nos encontraremos bajo la influencia de la precisión de que dispongan los aparatos de medida así como de la elección de una batería de experimentos bien seleccionada con el fin de provocar la mayor disparidad de comportamientos de nuestro equipo. Solo de esta manera podremos identificar su respuesta frente a cualquier variedad de entradas que puedan introducirse.

Cuando nos enfrentamos a sistemas mecánicos, llegar a las ecuaciones de movimiento será un trabajo más o menos tedioso dependiendo del mecanismo que tengamos delante, pero siempre podremos alcanzarlas. El problema que puede surgir viene a partir de la ignorancia de algún fenómeno involucrado en el movimiento de nuestra máquina. La relevancia que pueda tener dicho fenómeno en el comportamiento del sistema solo quedará de manifiesto cuando propongamos una comparación del modelo con el aparato real.

Es importante en la tarea de identificación de parámetros no perder de vista el sistema físico real que estamos tratando. Se debe tener claro que hay resultados que, pese a ser matemáticamente posibles e incluso correctos, no pueden ser aceptados como válidos por las implicaciones que tienen en la realidad. Véase como ejemplo que al identificar una masa no podremos obtener un valor negativo, como ocurrió en las etapas iniciales de identificación en este proyecto.

6.2. Experimentación

Nos hemos encontrado una falta de repetibilidad y continuidad en los resultados de los experimentos. Aplicando una variación pequeña a los parámetros del controlador podíamos observar un comportamiento completamente diferente del robot. Se ponía de manifiesto que había ciertos aspectos que quedaban al margen de nuestro modelo y que hacían que la respuesta difiriera de la respuesta esperada.

Se observado por tanto una gran fragilidad de los controladores propuestos en simulación a la hora de trasladarlos a la plataforma experimental real. En la práctica han sido necesarios ajustes paramétricos sobre el propio controlador experimental para compensar efectos que, en buena medida, se identificaron cómo debidos primordialmente a los efectos de una fricción irregular de una magnitud importante respecto a los pares articulares aplicados.

6.3. Fricción

Hemos comprobado en nuestra propia experiencia que la fricción es un fenómeno muy complejo. Una enorme cantidad de fenómenos de diferente índole ocurren para dar lugar a lo que nosotros conocemos como fricción macroscópica. Cualquier pequeño evento de desalineación o descentrado termina por provocar un rozamiento añadido y dependiente de la posición que hace que el sistema se comporte de manera inesperada en determinadas

situaciones.

Se observó incluso variabilidad en la fricción dependiendo de las condiciones ambientales y del tiempo que el robot llevase en operación. Todo esto sugiere que el fenómeno de la fricción es muy relevante en la dinámica de nuestro sistema, imponiendo severas limitaciones al comportamiento controlado del mismo.

Se ha de tratar con sumo cuidado ya que no siempre los controladores pueden absorber el efecto que la fricción tiene sobre la respuesta del sistema. Este fenómeno se puede entender como una perturbación, aunque es tan variable y tan variada que en muchas ocasiones no se puede evitar que se interponga en el correcto control del robot.

Hay una extensa literatura que trata esta problemática y de la que se pueden aprender numerosas lecciones. Una vez conocido esto hay que buscar la mejor manera de llevar, con los medios disponibles, la solución a la práctica porque esto no suele ser una situación de trivial respuesta.

6.4. Posible continuación

Se proponen como posible continuación al presente proyecto una serie de ideas que bien podrían ser desarrolladas en posteriores trabajos:

1. El ajuste de los parámetros del actual controlador para conseguir finalmente mantener controlado el pendubot en el punto de equilibrio superior inestable que se procuraba en este trabajo.
2. El estudio y realización de un compensador de fricción adaptativo que, antes de comenzar la experimentación con el robot, hiciera una serie de movimientos propuestos con el fin de identificar los valores más representativos de la fricción para cada estado diferente del sistema. Esto permitiría mitigar en cierta medida los efectos que la variabilidad de los parámetros de la fricción tienen en el comportamiento del sistema.
3. El desarrollo de un controlador que sea capaz de controlar el sistema con tres grados de libertad en vez de solo dos grados de libertad. Sabiendo de antemano que el movimiento de la tercera articulación va a estar íntimamente relacionado con las otras dos, teniendo que buscar un conjunto de valores para los pares que produzcan una respuesta adecuada del conjunto.
4. Ensayo de técnicas de swing-up alternativas a la propuesta en este trabajo. Por su robustez, las técnicas basadas en los conocidos como "modos deslizantes" son especialmente prometedoras.

Bibliografía

- [BP09] Guillermo Bejarano Pellicer. Identificación y control de un robot de dos grados de libertad de accionamiento directo, 2009.
- [FLS00] Isabelle Fantoni, Rogelio Lozano, and Mark W. Spong. Energy based control of the pendubot. *IEEE Transactions on Automatic Control*, vol. 45, 2000.
- [JB91] Daniel Jerome Block. Mechanical design and control of the pendubot, 1991.
- [SHV04] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Dynamics and Control*, chapter 9, pages 193–229. Second edition, 2004.
- [Ted09] Russ Tedrake. *Underactuated Robotics: Learning, Planning and Control Efficient and Agile Machines, Course Notes for MIT 6832*. Massachusetts Institute of Technology, 2009.
- [XKO02] Xin Xin, Masahiro Kaneda, and Toshitaka Oki. The swing up control for the pendubot based on energy control approach. *IFAC*, 2002.

ANEXOS

Anexo A

Descripción del sistema físico

A.1. Introducción

En el presente documento se procederá a describir los distintos dispositivos que componen el sistema, así como el hardware y el software utilizados para el control del mismo. La información aquí expuesta se puede revisar en [BP09] puesto que la plataforma experimental utilizada es exactamente la misma para ambos proyectos.

A.2. Robot de tres grados de libertad

En las Figuras A.1, A.2, A.3, A.4 y A.5 se pueden observar distintas perspectivas y detalles del robot.

Se describirá a continuación cada uno de sus componentes.

A.2.1. Motores

Los motores pertenecen a la gama F Series Frameless DDR Motor de Kollmorgen, y se encuadran dentro de la familia de Direct Drive Rotary Brushless Motors.

Los motores en cuestión son el F7950 para el hombro y el F4318 para el codo. Las características principales de estos motores se muestran en la tabla A.1.

	Par continuo (Nm)	Par de pico (Nm)	Peso (Kg)
F4318	18	60.6	5.2
F7950	128	305	29.8

Tabla A.1: Características de los motores

A.2.2. Brazos de giro

Los brazos de giro están realizados en placa de aluminio de 15 mm de espesor, siendo la distancia entre los huecos para los motores de 450 mm. El primer brazo, denominado en adelante *hombro*, es el que se sitúa entre ambos motores, mientras que el segundo



Figura A.1: Vista frontal del robot



Figura A.2: Vista del perfil del robot



Figura A.3: Detalle de la sujeción al suelo



Figura A.4: Detalle del motor del hombro



Figura A.5: Detalle del motor del codo y del encoder incremental



Figura A.6: Detalle de la tercera articulación

brazo, al que se llamará *codo*, es el que se encuentra entre el motor más pequeño y el encoder incremental final. Desde el encoder incremental cuelga una tercera articulación, la última de todas, que denominaremos *muñeca*. No dispone de motor (o de freno) por lo que gira libremente impulsada por el movimiento de las otras dos articulaciones. Debido a la ausencia de actuador en la *muñeca* la fricción es sustancialmente menor que en las otras dos articulaciones. Se aprecia en detalle en la Figura A.6

A.2.3. Encoder incremental

El encoder incremental situado en el extremo de la segunda articulación es el modelo *EL58* del fabricante Eltra. En la Figura A.7 se muestran sus principales características electrónicas.

Características Electrónicas Serie EL	
Resolución	De 1 a 10000 impulsos por rev.
Tensión de alimentación	5 + 28 Vdc Nota: LINE DRIVER, disponible solamente con alimentación 5 Vdc o 8 a 24 Vdc.
Absorción en vacío	80 mA
Máx. Corriente conmutable	50 mA por canal 20 mA por canal LINE DRIVER
Configuración Electrónica de salida	NPN / NPN OPEN COLLECTOR / PUSH PULL / LINE DRIVER
Máx frecuencia de uso	Max 300 KHz $F = \frac{\text{RPM} \times \text{Resolución}}{60}$

Figura A.7: Características del encoder incremental

A.3. Servoactuadores

Los servoactuadores utilizados para ambos motores son los *ServoStar 600* de la firma Kollmorgen. Estos dispositivos permiten el control preciso del par aplicado a los motores. Sus principales características son:

- Conexión RS232.
- Conexión bus CAN.
- 2 entradas analógicas.
- 2 salidas analógicas.
- 4 entradas digitales programables.
- 2 salidas digitales programables.

- Control digital:
 - Corriente.
 - Velocidad.
 - Posición.
- Evaluación de señal de resolver.
- Evaluación de señal de encoder seno-coseno.
- Salida de simulación de encoder.

En la Figura A.8 se muestra el aspecto del servo, mientras que en la Figura A.9 se ve el esquema general de componentes auxiliares del servo. Por último en la Figura A.10 se muestra el diagrama de conexiones, donde se detallan las entradas y salidas del servoactuador.

A.4. Tarjetas

Se dispone asimismo de un par de tarjetas de lectura de valores de encoder y de entrada/salida analógicas de la firma National Instruments. En concreto se utiliza la tarjeta *NI 6601* para la lectura de los valores de encoder, y la tarjeta *NI 6024E* para las entradas/salidas analógicas.

La tarjeta *NI 6601* posee la capacidad de leer señales de hasta dos encoders absolutos, y teniendo en cuenta que cada servoactuador posee una salida simulada de encoder absoluto, esta tarjeta resulta adecuada para la conexión. La justificación del uso de señales diferenciales puede encontrarse en su mejor funcionamiento ante el ruido propio de entornos industriales.

Por su parte, la tarjeta *NI 6024E* posee, entre otras, 2 salidas analógicas (DAC0OUT y DAC1OUT) y 16 entradas analógicas. De este modo, se utiliza cada una de las señales de salida analógica como señal de control: DAC0OUT para el codo y DAC1OUT para el hombro.

A.5. Sistema de control

El sistema de control del robot de tres grados de libertad está basado en el sistema operativo en tiempo real *xPC Target* de Mathworks. Este es un entorno de alto rendimiento, que permite conectar modelos de Simulink o Stateflow con un sistema real y ejecutarlos en tiempo real en un hardware PC compatible.

El diseño de controladores para el sistema es sencillo, ya que *xPC Target* permite añadir bloques de entrada/salida a los modelos de Simulink. Para ello dispone de una amplia librería de dispositivos de entrada/salida, que es capaz de gestionar de forma transparente al usuario.

Los principales componentes del sistema de control son los siguientes:



Figura A.8: Kollmorgen Seidel Servostar 600

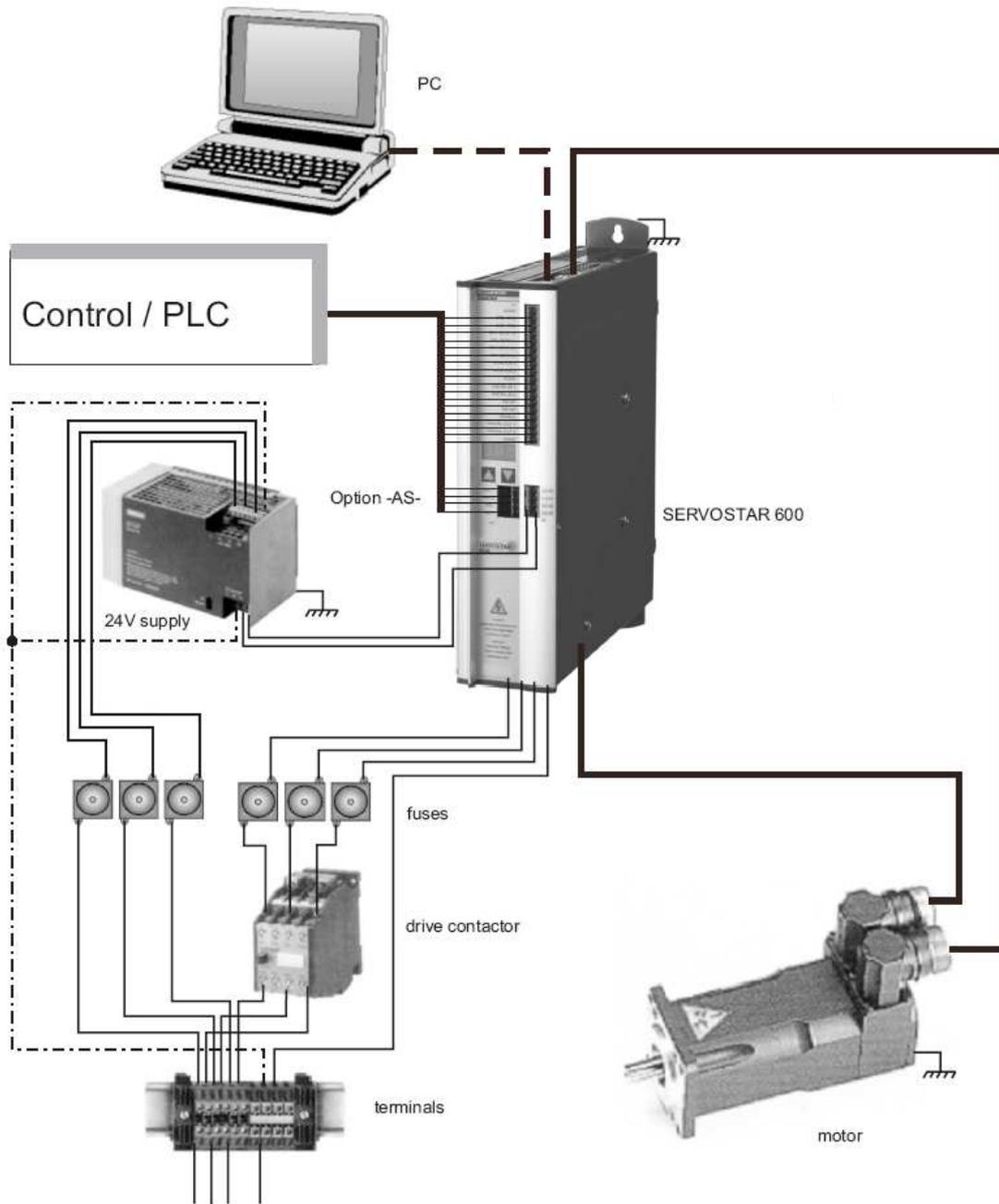


Figura A.9: Componentes del sistema del servoactuador

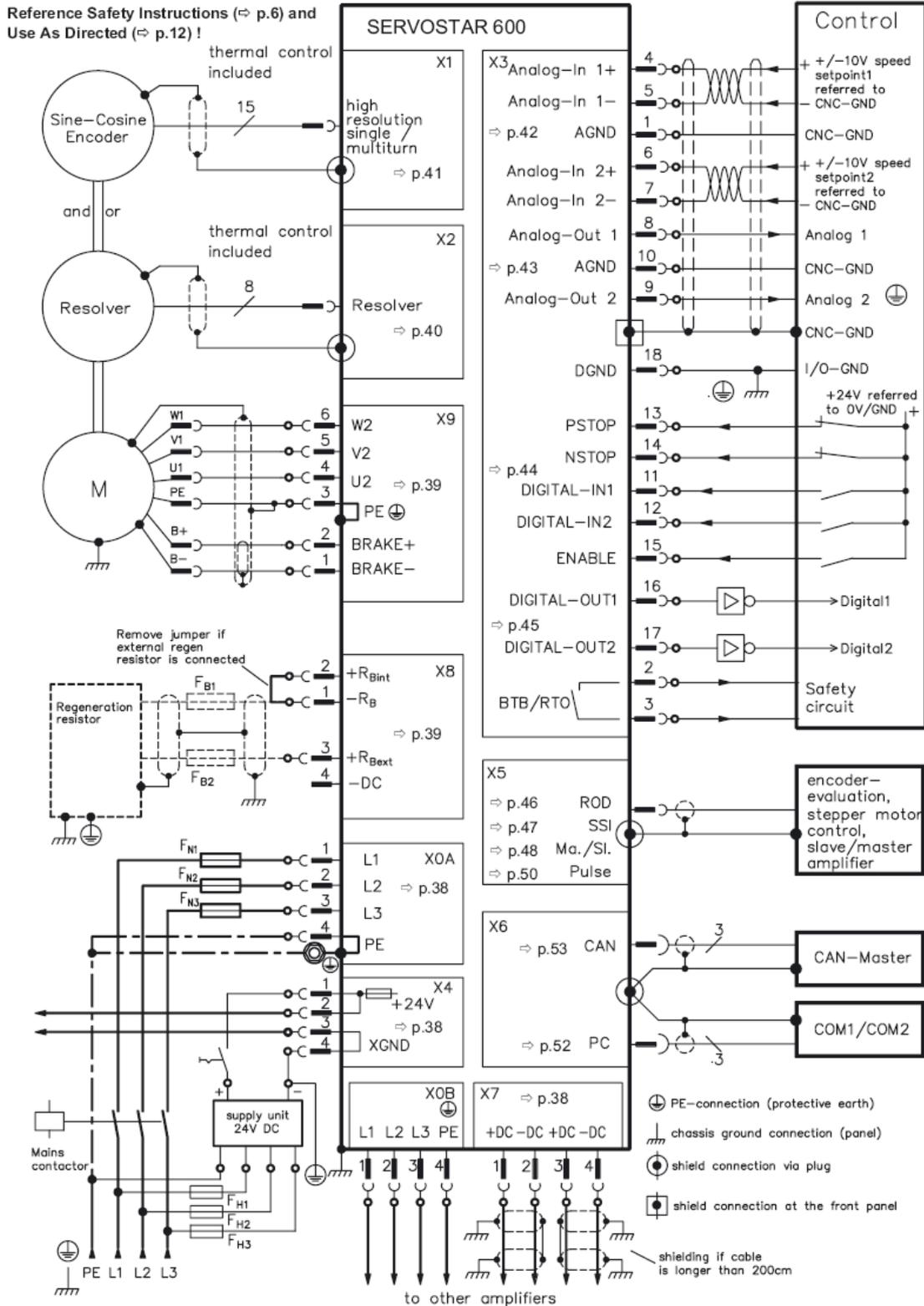


Figura A.10: Diagrama de conexiones del ServoStar 600

- PC Host: en él se modela en Simulink el sistema de control. Una vez realizado el modelo se compila y se transmite mediante TCP/IP al PC Target.
- PC Target: posee el sistema operativo en tiempo real xPC Target. Recibe vía TCP/IP el código necesario para ejecutar el modelo diseñado en el PC Host. En este PC se encuentran las tarjetas que permiten la comunicación con el sistema real, en concreto las descritas anteriormente y una tarjeta de comunicación con el bus CAN.

El esquema general del sistema se muestra en la Figura A.11.

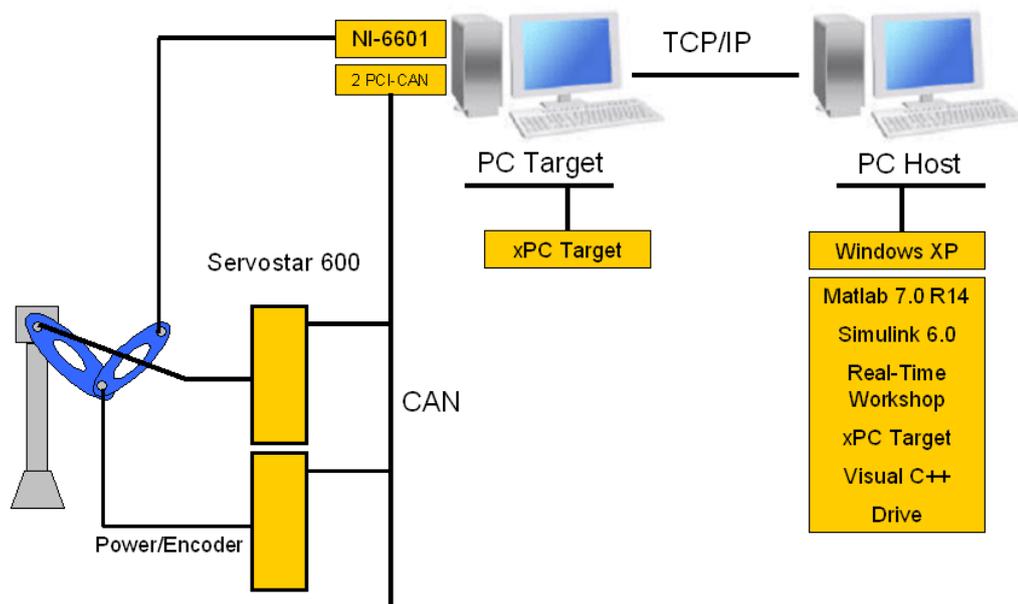


Figura A.11: Esquema del sistema de control

A continuación se procederá a describir de forma más minuciosa cada uno de los componentes del sistema de control.

A.5.1. PC Host

Este PC está dedicado a la creación de los modelos o controladores que luego van a ejecutarse en tiempo real en el PC Target y sobre el robot real. Los modelos pueden ser creados en la herramienta gráfica Simulink. Para ello debe disponerse del siguiente software:

- MATLAB 7.0 R14 o superior: con el correspondiente Toolbox de xPC Target. Permite la creación del disquete de arranque del PC Target y contiene las librerías de Simulink de las tarjetas de entrada/salida compatibles con el sistema.
- Simulink 6.0 o superior.
- Real Time Workshop.

- Visual C++: necesario para compilar el modelo de Simulink para transferirlo posteriormente al PC Target.
- Drive: programa de configuración del servoactuador ServoStar 600. Permite tanto configurar distintos parámetros como implementar varios controles en modo par, velocidad y posición, entre otras funciones.
- Conexión RS232.

A.5.2. PC Target

En este equipo se ejecuta la aplicación en tiempo real. A él están conectadas las siguientes tarjetas de adquisición de datos:

- Softing CAN-AC2-PCI: tarjeta PCI con 2 buses CAN independientes.
- NI 6024E: tarjeta de entrada/salida analógica.
- NI 6601: tarjeta de Counters/Timers con modo de funcionamiento de lectura de encoder.

A esta última tarjeta le acompañan los siguientes accesorios para facilitar la conexión del encoder incremental:

- R6868 Low-Cost, Unshielded Ribbon Cable, 1 m.
- CB-68LP Low-Cost, Unshielded I/O Connector Block.

El PC Target se carga con el sistema operativo xPC Target mediante un disquete, así que no necesita ningún programa en especial, simplemente la configuración para que arranque con el sistema operativo encontrado en la disquetera al iniciarse.

A.5.3. Bus CAN

Para el intercambio de información entre el ServoStar 600 y el sistema en tiempo real se utiliza el protocolo de aplicación CANopen sobre el bus CAN. La conexión mediante el bus CAN se utiliza como alternativa a la conexión tradicional para eliminar el cableado, puesto que con la interfase CANopen puede realizarse tanto la configuración inicial como la comunicación necesaria para el control. Esta comunicación mediante el bus CAN se establecerá entre una tarjeta controladora instalada en el PC Target, modelo DS1103 PPC, y el servoactuador.

A.5.4. Tarjeta controladora DS1103 PPC

La tarjeta DS1103 PPC se encuentra alojada en el PC Target y se encarga de realizar el control de los motores a través de los servoactuadores. La tarjeta dispone de un microprocesador PowerPC a 400 MHz con un controlador DSP de la firma Texas Instruments TMS320F240 integrado como un subsistema.

El software asociado a la tarjeta controladora permite crear una interfaz gráfica para la monitorización y control de las señales de forma sencilla. Además, dicho software está

integrado con otras herramientas como MATLAB, que se utilizará en el desarrollo de controladores para el robot, gracias a que como ya se ha comentado se puede utilizar con lenguajes de programación como C. Así, el controlador desarrollado mediante MATLAB y compilado en C, puede descargarse a la tarjeta y comprobar su funcionamiento mediante la interfaz gráfica.

A.5.5. Software de configuración de ServoStar 600 (Drive)

Este software permite configurar de forma sencilla los servoactuadores. Debe estar instalado en el PC Host, y puesto que el PC Host estará conectado al servoactuador mediante uno de sus puertos, se puede establecer comunicación entre ellos para la configuración. También puede configurarse desconectado.

Los parámetros más interesantes de cara a la experimentación con el robot son la resolución de los encoders en cuanto a número de pulsos por vuelta, y el modo seleccionado para el control. La resolución puede elegirse entre 256 y 524.288 líneas por vuelta. Respecto al modo de control, el más adecuado para la aplicación es el modo de par.

Es importante precisar que al disponer de dos servoactuadores, es necesario seleccionar a uno de ellos como el master de la aplicación. En este caso se ha seleccionado el servoactuador del codo como master, se puede apreciar en la Figura A.13.

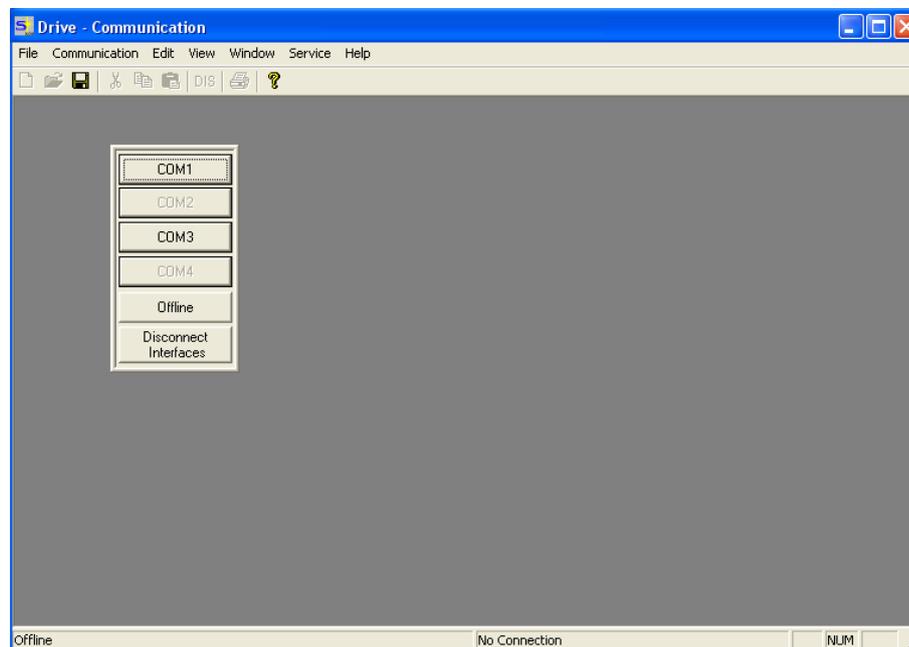


Figura A.12: Software Drive: menú para elegir el puerto

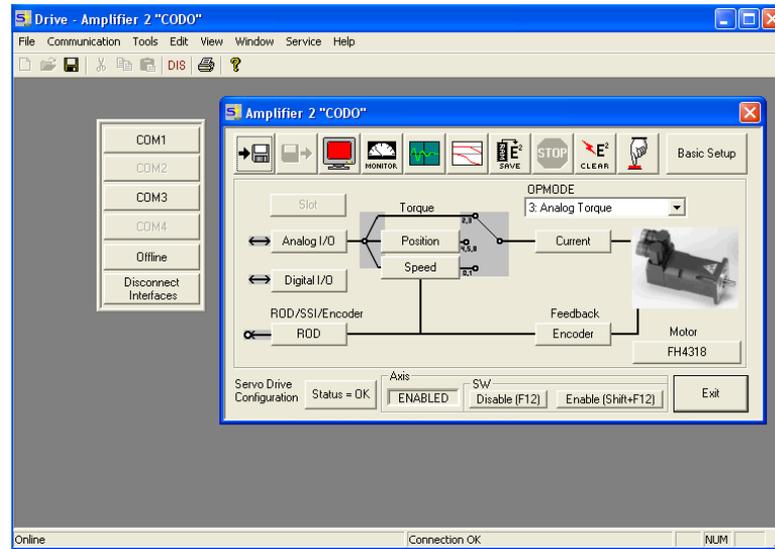


Figura A.13: Software Drive: menú para manipular cada articulación

A.6. Imágenes de los distintos elementos del sistema

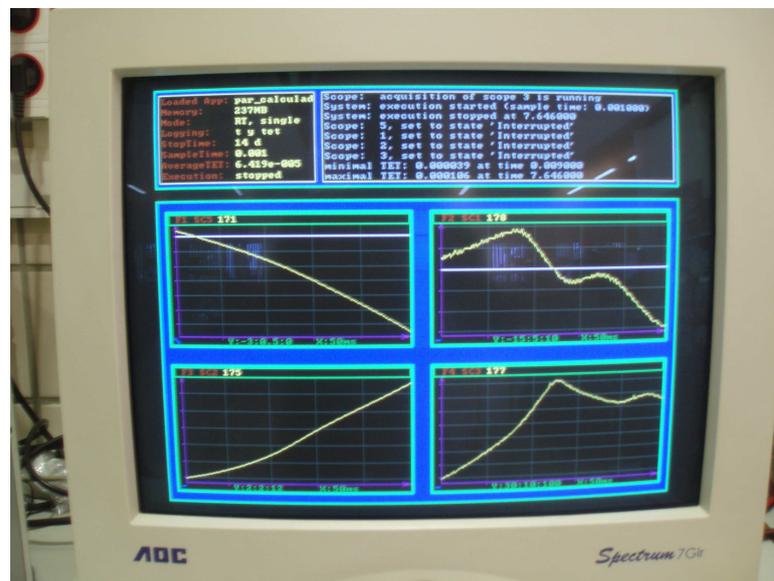


Figura A.14: PC Target e interfaz gráfica

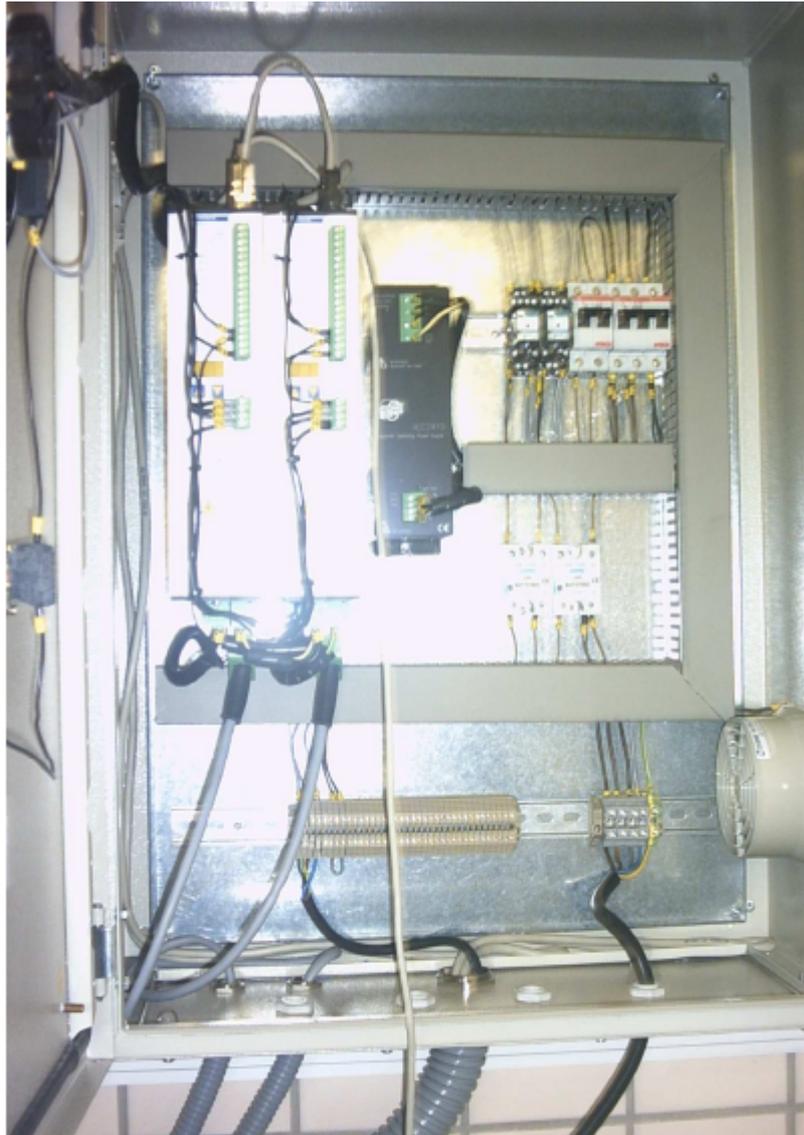


Figura A.15: Servoactuadores y cuadro eléctrico



Figura A.16: Cierre del cuadro eléctrico