



Trabajo Fin de Máster

Simulación del Control Exoesqueleto Mediante el Paradigma Assist-As-Needed

Autor

Adrián Rivero Pérez

Director

Luis Montano Gella

Escuela de Ingeniería y Arquitectura
Curso 2011-2012

Simulación del Control Exoesqueleto Mediante el Paradigma Assist-As-Needed

RESUMEN

Este proyecto simula el comportamiento de un exoesqueleto acoplado a un brazo humano según el paradigma *Assist-as-needed* (asistencia bajo demanda). Este paradigma consiste en que durante los ejercicios de rehabilitación el robot no realiza todo el trabajo de movimiento sino que asiste, en la medida necesaria, el movimiento residual del paciente, permitiendo que éste pueda recuperar parte del movimiento del brazo sano. Con este propósito se ha diseñado un sistema de control del exoesqueleto de tal forma que el par del robot se regule según aquel que da el paciente.

El sistema de control ofrece tres modos de funcionamiento pensados para adecuarse a las distintas terapias de rehabilitación. El modo asistencia es el que se adecua al paradigma *assist-as-needed* y puede ser regulado entre 0% y 100% de manera manual o automática según las condiciones del paciente. Un nivel 100 de asistencia significa que el robot realiza, por si mismo, los movimientos de rehabilitación preprogramados cargando con el brazo del paciente. Éste no realiza ningún esfuerzo. A medida que disminuye la asistencia el robot hace un porcentaje menor de fuerza. En consecuencia, el paciente tendrá que ejercer un esfuerzo cada vez mayor consiguiendo así una mejora. En el nivel de asistencia 0 el robot entra en el modo seguidor. Aquí el paciente es el que marca los movimientos mientras que el robot se limita a seguirle. Por último, el modo resistencia es aquel en el que el robot ofrece una cierta fuerza en contra del movimiento prefijada por un nivel de resistencia, obligando al paciente a ejercer sus músculos.

Se han realizado simulaciones para validar todos los modos diseñados y se muestran resultados que las apoyan. Para ello se han programado distintos ejercicios de rehabilitación en los que se pueden observar los distintos parámetros dinámicos a los que están sometidos el robot y el paciente.

Del mismo modo se ha llevado a cabo una sesión de experimentación con un brazo robótico Kuka perteneciente al centro IBEC de Barcelona, donde se ha encontrado una relación entre el trabajo que realiza el paciente y los niveles de asistencia y resistencia.



ÍNDICE

CAPÍTULO 1.- MOTIVACIÓN	2
CAPÍTULO 2.- OBJETIVOS	3
2.1.- SISTEMA	3
2.2.- MODO DE ASISTENCIA	4
2.3.- MODO SEGUIMIENTO	5
2.4.- MODO RESISTENCIA	6
CAPÍTULO 3.- IMPLEMENTACIÓN	7
3.1.- INTRODUCCIÓN	7
3.2.- PARÁMETROS ROBOT	7
3.3.- MATRICES EJERCICIO	8
3.4.- MODO DE ASISTENCIA Y MODO SEGUIDOR	8
3.4.1.- <i>Inicialización y entrada</i>	8
3.4.2.- <i>Sistema</i>	8
3.5.- MODO RESISTENCIA	11
3.5.1.- <i>Inicialización y entrada</i>	11
3.5.2.- <i>Sistema</i>	11
3.5.3.- <i>Salidas</i>	12
3.5.4.- <i>Control</i>	12
CAPÍTULO 4.- RESULTADOS	14
4.1.- MODO ASISTENCIA - MANUAL	14
4.2.- MODO SEGUIMIENTO	16
4.3.- MODO ASISTENCIA – AUTOMÁTICO	18
4.4.- MODO RESISTENCIA	19
4.5.- CONCLUSIONES Y COMENTARIOS SOBRE LOS RESULTADOS	19
4.6.- CONTROL	19
BIBLIOGRAFÍA	23



CAPÍTULO 1.- Motivación

La presente tesis forma parte del proyecto “**dispositivos híbridos neuroprotésicos y neurorrobóticos para compensación funcional y rehabilitación de trastornos del movimiento**” normalmente llamado **proyecto Hyper**[3]. Este proyecto mostrará un importante avance en la investigación de dispositivos neurorrobóticos (NR) y neuroprotésicos (MNP) en una interacción cercana con el cuerpo humano y enfocados a la rehabilitación y a la compensación funcional de trastornos motores en las actividades de la vida diaria. El enfoque de este proyecto estará encaminado a desarrollar, de una manera científico tecnológica, configuraciones innovadoras de NR y MNP que conjuntamente potenciarán y ayudarán a restablecer las capacidades latentes en las personas que, afectadas de lesiones cerebrovasculares, parálisis cerebral o lesión medular, presentan trastornos motores en miembro inferior o superior.

Los principales objetivos del proyecto Hyper son la de restaurar la función motora en pacientes con lesión medular a través de la compensación funcional y promover el reaprendizaje del control motor en pacientes afectados por accidente cerebro vascular y parálisis cerebral, por medio de un uso integrado de NR y MNP.

Este proyecto validará funcional y clínicamente el concepto de sistemas neurorrobóticos-neuroprotésicos híbridos para la rehabilitación y compensación funcional de trastornos motores bajo el paradigma assist-as-needed (asistencia bajo demanda). El proyecto Hyper asume que el avance en las terapias de rehabilitación física depende de la obtención de una comunicación más transparente entre los pacientes y las máquinas por lo que se explorarán diferentes niveles de actividad neural humana.

Hyper plantea preguntas fundamentales en diferentes disciplinas tecnológicas y científicas. Estas preguntas plantean seis líneas de investigación: biomecánica, control neuromotor, tecnologías de control, tecnologías de sensores, actuadores y energía, interfaces multimodales cerebro máquina y adaptación de sistemas híbridos a escenarios de aplicación además de todas las interacciones horizontales.

- Los sistemas tratarán la variabilidad en las estructuras neuromusculares humanas, con adaptaciones dinámicas acordes a las capacidades latentes de los usuarios.
- La aproximación propuesta promoverá el reentrenamiento del control sensorial y motor, y la plasticidad cerebral. Esto podrá conducir a potenciales soluciones de rehabilitación de pacientes con apoplejía y parálisis cerebral, y a soluciones para la compensación funcional de pacientes de con lesión medular.

En concreto, la presente tesis forma parte de la línea de investigación de tecnologías de control aplicada solo al miembro superior. Esta tesis pretende simular los sistemas brazo paciente y brazo robot actuando conjuntamente y en todos los casos que requiere el proyecto Hyper. El sistema creado no solo será integrado finalmente en el proyecto Hyper, sino que los resultados permitirán el correcto diseño y desarrollo de la prótesis robótica que asista al miembro superior.



CAPÍTULO 2.- Objetivos

El objetivo de esta tesis es el de simular el comportamiento de la ortesis robótica de miembro superior acoplada al brazo de un paciente humano. Con esto se pretende establecer un sistema de control apropiado para la correcta interacción máquina – humano además de obtener resultados numéricos que permitan el diseño de un robot físico real. Este sistema estará orientado a la rehabilitación, potenciación y restablecimiento de las capacidades latentes del miembro superior de los pacientes con trastornos motores en los mismos. Es por esta razón que el sistema deberá permitir diferentes modos de funcionamiento para diferentes tipos de terapia de rehabilitación.

Se distinguen tres modos:

- **Modo de asistencia**, que entra dentro del **paradigma assist-as-needed** (ver capítulo 1: motivación),
- **Modo de seguimiento**.
- **Modo de resistencia**.

.- En el **modo de asistencia** el exoesqueleto deberá ayudar al paciente a completar un ejercicio. El exoesqueleto ayudará más o menos al paciente según se marque en el nivel de asistencia α que variará entre 0 y 1. Si $\alpha = 1$ será el robot el que haga todo el movimiento “tirando” a la vez del brazo del paciente. Si $0 < \alpha < 1$ el paciente deberá ejercer una cierta fuerza y el robot suministrará el resto, indicado por el valor de α .

.- El **modo seguimiento** es el modo de asistencia cuando $\alpha = 0$. En este caso el robot deberá moverse solo a si mismo siguiendo las órdenes de movimiento que realice el paciente. No habrá interacción de fuerzas entre los dos sistemas, robot y paciente, sino que cada uno se ocupará de realizar el par necesario para moverse a si mismo.

.- Por último en el **modo resistencia**, el robot ejerce cierta resistencia al movimiento indicada por un nivel de resistencia β . Este modo será el equivalente de un entrenamiento con pesos, de forma que el paciente recupere o adquiera una mayor fuerza muscular. Es el caso contrario al modo de asistencia, donde el robot ayuda al paciente; aquí el paciente ayuda al robot.

El sistema deberá simular tanto el brazo robótico como el brazo del paciente y calculará todos los pares que sean necesarios y que permitan una posterior construcción del sistema real. Además el sistema deberá de admitir pequeñas variaciones para poder utilizarlo en el robot real.

2.1.- Sistema

El robot simulado en este sistema tendrá **5 grados de libertad** para simular un exoesqueleto lo más parecido al brazo humano. Un exoesqueleto de este tipo dispondrá de **3 articulaciones en el hombro** que permitirán la flexión, abducción y rotación y **2 articulaciones en el codo** que permitirán la rotación y la flexión. Una simulación con tantos grados de libertad permitirá una mejor adaptación en robots que tengan menos grados de libertad. El peso del robot y la distancia entre sus articulaciones serán parámetros de entrada adaptables a cualquier tipo de robot. Los parámetros del paciente también podrán ser modificados para hacer el sistema adaptable a cualquier tipo de persona.

La figura 2.1 resume las entradas y salidas que deberá tener el sistema. La entrada “sensor de fuerza” será un sensor que indique al sistema cuanta fuerza está ejerciendo el paciente y es la manera en la que este se comunica con el exoesqueleto. La entrada “ejercicio de rehabilitación” solo tendrá importancia en el modo de asistencia donde el paciente, por si solo, no puede ejecutar un ejercicio al completo y para que el robot pueda guiarle, necesitará, como datos los movimientos a seguir. En los otros dos modos, donde el paciente ya tiene fuerza para moverse a si mismo, será él, a través del sensor de fuerza, el que guíe al sistema. En el modo de funcionamiento podrá elegirse entre los tres indicados: asistencia, seguimiento y resistencia. Esto es muy importante puesto que el sistema funcionará de diferente manera en cada modo. El par a la salida del sistema y que será la entrada a los motores de las articulaciones, será distinto en cada modo de funcionamiento.

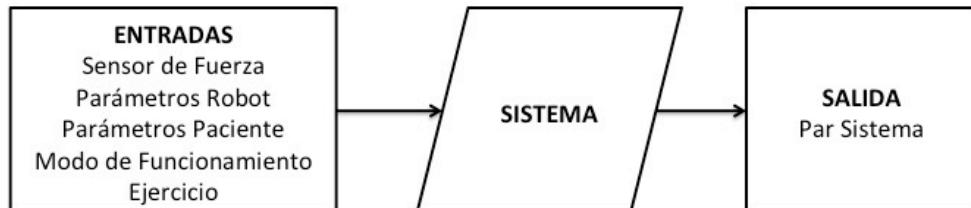


Figura 2.1: Resumen Entradas y Salidas del Sistema

2.2.- Modo de Asistencia

Este modo de funcionamiento permitirá la recuperación gradual del movimiento del brazo del paciente. Cuanto mayor sea el nivel de asistencia α menos fuerza tiene que hacer el paciente y más hará el robot. La figura 2.2 detalla el bloque “Sistema” de la figura 2.1 en el modo de asistencia.

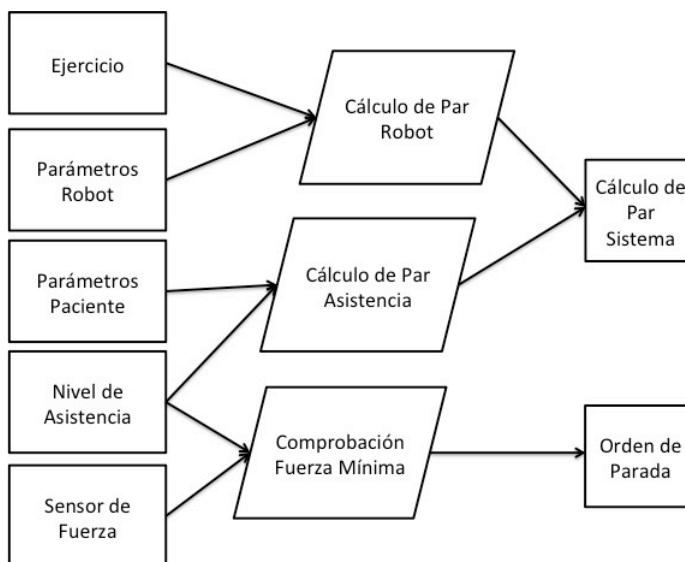


Figura 2.2: Detalle del Sistema en el Modo Asistencia

En primer lugar hay que diferenciar en este sistema, tres tipos de pares:

- ☒ **Par Robot (τ_{min}).**
- ☒ **Par de Asistencia (τ_{assist}).**
- ☒ **Par del Sistema ($\tau_{sistema}$).**

.- El **Par Robot** es el par que necesitaría el robot para poder moverse si no tuviera anclado al paciente, es el par articular mínimo necesario.

.- El **Par de Asistencia** es el par extra que ha de ejercer el exoesqueleto para ayudar al paciente y que vendrá determinado por el nivel de asistencia y las características físicas de cada paciente. Este par, sumado al que hace el paciente deberá ser el par que haría un paciente sano.

.- Por último el **Par del Sistema** es el par, en este modo, que deben de hacer los motores y que será la suma de los otros dos, es así una entrada directa al robot. El cálculo del par de asistencia será proporcional al nivel de asistencia α y al par que haría el paciente si estuviera sano τ_{sano} :

$$\tau_{assist} = \alpha \tau_{sano}$$

Ecuación 2.1

Debido a esto, se ha de cumplir que:

$$\tau_{paciente} \geq (1 - \alpha) \tau_{sano}$$

Ecuación 2.2

Siendo $\tau_{paciente}$ el par que ejerce el paciente. Esto es necesario ya que, si el paciente no ejerce el par que establece el nivel de asistencia, el exoesqueleto necesitará un $\tau_{sistema}$ mayor al que se le ha introducido en la entrada para poder moverse. Si esto llega a ocurrir, se envía al sistema una orden de parada que detendrá la ortesis en el lugar que esté. Esto es necesario para que el conjunto paciente-robot no caiga por su propio peso, algo que podría dañar no solo al paciente sino también a la estructura del robot.

En resumen se ha de cumplir que:

$$\tau_{sano} = \tau_{paciente} + \tau_{assist}$$

Ecuación 2.3

Si no el sistema entrará en parada. Esto puede ser útil en otro tipo de ejercicio de rehabilitación donde el paciente ha de lograr mover su brazo y la prótesis. Es decir, tendrá que esforzarse en conseguir ejercer una fuerza mayor a la que establece el nivel de asistencia. Cuando lo consiga, el robot se moverá el tiempo en el que el paciente pueda mantener este par.

El sistema también tendrá la opción de elegir el nivel de asistencia de forma manual o automática. En el caso manual, es el fisioterapeuta, encargado de la rehabilitación del paciente, el que lo establece. En el caso automático, se le pedirá al paciente que intente hacer un cierto movimiento. Utilizando el sensor de fuerza, se compara esta con la fuerza que haría esta persona en condiciones sanas (a partir de una base de datos y un percentil). El porcentaje de fuerza que hace el paciente respecto a las condiciones sanas nos permite calcular el nivel de asistencia α que necesita ese paciente en concreto.

$$\alpha = \frac{\tau_{sano} - \tau_{paciente}}{\tau_{sano}}$$

Ecuación 2.4

2.3.- Modo Seguimiento

El sistema entra en este modo cuando los niveles de asistencia y resistencia son cero. La figura 2.3 detalla el bloque sistema de la figura 1 cuando está activo el modo seguimiento.

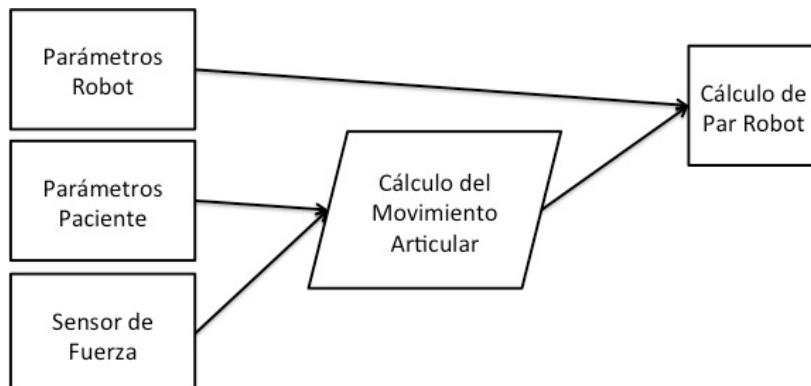


Figura 2.3: Detalle del Sistema en modo seguimiento

Este sistema leerá en cada instante la fuerza que ejerce el paciente en la muñeca. A partir del modelo del paciente se calculan los movimientos necesarios en cada articulación. Por último, a partir de esto se calcula el par que deben hacer los motores para seguir el movimiento que quiere hacer el paciente. En el sistema ideal el paciente moverá el brazo y automáticamente el robot copiará el movimiento, el paciente debería poder mover el brazo como si no tuviese la prótesis. Obviamente, en el caso real, habrá cierto retardo entre que el paciente aplica la fuerza y el robot se mueve. En este modo de funcionamiento el par del sistema será el mismo que el par robot ya que el par de asistencia es nulo.



2.4.- Modo Resistencia

Este modo de funcionamiento se activará cuando el nivel de asistencia sea nulo pero el nivel de resistencia sea mayor a cero. Este modo de funcionamiento, detallado en la figura 4, obliga al paciente a ejercer una fuerza mayor de la que normalmente requiere para mover el brazo. Para conseguir esto el par de entrada a los motores del robot será menor al que necesitaría el para moverse a si mismo. Este par será:

$$\tau_{motor} < \tau_{min} \quad \text{Ecuación 2.5}$$

$$\tau_{motor} = \tau_{min} - \beta \tau_{sano} \quad \text{Ecuación 2.6}$$

Donde el segundo término de la ecuación 2.6 es el extra que debe hacer el paciente para que haya algún movimiento en el conjunto paciente-robot. Matemáticamente, el par del paciente es:

$$\tau_{paciente} = \tau_{sano}(1 + \beta) \quad \text{Ecuación 2.7}$$

Es decir, proporcional a lo que haría estando sano.

Este modo de funcionamiento es el equivalente de usar pesos o mancuernas para ejercitar los músculos del brazo del paciente y permite la utilización del brazo robótico no solo en terapias de rehabilitación sino además como herramienta para ejercitar los brazos de cualquier persona.



CAPÍTULO 3.- Implementación

3.1.- Introducción

En este capítulo se detallará la programación del sistema explicado en el capítulo anterior. El sistema ha sido programado utilizando Matlab y el Toolbox de Simulink y el “Robotics Toolbox” una herramienta gratuita para Matlab que incluye distintos bloques utilizados en robots.

Todos estos bloques, y la matemática que llevan detrás, se explican en el Anexo 1 mientras que los programas de Matlab utilizados están en el Anexo 2. Además en el Anexo 4 se muestra una alternativa a este diseño que finalmente fue rechazada por problemas con los bloques pero que puede ser útil en otros proyectos.

Tal y como se explicó en el Capítulo 2, el sistema tiene tres modos de funcionamiento distintos: **modo de asistencia, modo de resistencia y modo seguidor**. Se explicarán primero el modo de asistencia y el modo seguidor por ser la base de todo el sistema y luego se explicarán los cambios que se han hecho para poder ajustar el sistema al modo de resistencia. De esta manera se enumerarán las entradas, configuraciones y funcionamiento del sistema. Para todos los modos existen cuatro partes bien diferenciadas:

- ▣ .- **Los parámetros** que definen a los robots y que cambian según el modo.
- ▣ .- **La matriz del ejercicio** a simular,
- ▣ .- Un **archivo de inicialización**, que cargará estos parámetros, el movimiento y además nos permite seleccionar el modo, cambiando los parámetros según la selección
- ▣ .- Por último **los bloques Simulink** encargados de llevar a cabo la simulación.

Además, en este capítulo, también se detallará un sistema de control de par para el futuro montaje real del robot. Este sistema se ha diseñado aparte, ya que no tiene cabida en la simulación donde los cálculos son ideales. Este sistema se encargará de compensar los pequeños errores que aparecen en un sistema real: retrasos, inconsistencias y ruidos provenientes de vibraciones.

3.2.- Parámetros Robot

El brazo del paciente se ha simulado como un brazo robótico, por lo que tendrá un archivo de parámetros muy parecidos a los de la ortesis. Estos archivos de parámetros lo que hacen es, a partir de las propiedades del brazo y antebrazo (masas, inercias y longitudes) crearán un objeto robot con estas características. Además, como se trata de la simulación de un robot real se deben añadir otros valores como son: la fricción y relación de los ejes, y la fricción viscosa del motor. Puesto que se desconocen los parámetros del robot real, se han utilizado parámetros parecidos a los del **robot puma560** cuyos datos vienen dentro del “**robotics toolbox**” en el archivo “**mdl_puma560.m**”. Estos datos fueron ligeramente modificados, para permitir una correcta simulación del brazo robótico.

Cada objeto robot se caracteriza con los parámetros “**Denavit-Hartenberg**” o comúnmente denominados **DH**. Estos constituyen la matriz que especifica las articulaciones del robot con unas coordenadas relativas al punto anterior. De esta manera se tendrá un objeto robot con tres articulaciones en el hombro: flexión, abducción y rotación, y dos articulaciones en el codo: rotación y flexión. Además se ha colocado el efecto final al principio de la muñeca. Puesto que se tienen varias articulaciones en un mismo punto, los parámetros físicos que caracterizan las uniones entre ella, como son la inercia, longitud y masa son cero ya que, realmente no existen. Las matrices DH y el resto de parámetros de ambos objetos robot pueden ser vistos en el Anexo 3.

Este archivo de parámetros permitirá la adaptación del sistema a distintos pacientes y a distintos robots, cambiando el peso y las longitudes de sus segmentos. Otra aproximación que se ha hecho en esta simulación es la de asemejar tanto el brazo como el antebrazo a dos cilindros con su centro de gravedad justo en su centro. Así la inercia de cada segmento y las coordenadas de su centro de gravedad son las de cilindros. De cara a la simulación real, deberían darse valores reales para estos parámetros.



Por último se supone que el brazo robótico y el brazo del paciente están en la misma posición. Como físicamente esto no es posible, cuando el sistema se simule en la realidad deberá añadirse el desfase que existe entre los dos.

3.3.- Matrices Ejercicio

Los movimientos que definen cada ejercicio vienen descritos en una matriz q dentro de diferentes programas y está compuesta por las posiciones angulares de cada articulación en radianes. Así pues, para simular un ejercicio en concreto, basta con escribir el nombre del archivo que lo define en la parte correspondiente del archivo de inicialización. Dentro de cada archivo de ejercicio se puede simular rápidamente el movimiento, un brazo robot realizará el ejercicio al completo para que el usuario pueda tener una previsualización del mismo antes de realizar la simulación y los cálculos, que suelen tardar varios minutos.

3.4.- Modo de Asistencia y Modo Seguidor

3.4.1.- Inicialización y entrada

Los parámetros de inicialización de este modo de funcionamiento serán: el **nivel de asistencia α** , el **tiempo de movimiento t** , el **tiempo de muestreo o paso** y la **selección de automático o manual**.

Este submodo, que se activa haciendo **automático = 1**, sirve para poder elegir un nivel de asistencia apropiado para el paciente, es decir el paciente hará una fuerza para realizar un movimiento y el sistema lo comparará con la fuerza que debería hacer este paciente en condiciones sanas, de la comparación resultará un nivel de asistencia recomendado. En este caso, para simular este efecto, se multiplica el par que ejerce el paciente por un número aleatorio (de 0 a 1) que indica su grado de discapacidad. El nivel de asistencia y el grado de discapacidad siempre han de sumar 1, ya que si no el robot no le dará al paciente la ayuda que necesita y este último no logrará mover su brazo.

Con **automático = 0**, el nivel de asistencia debe ser elegido a priori. Si el nivel de asistencia es igual a cero, el sistema entrará en **modo seguidor**. Normalmente el nivel de asistencia, como se dijo antes, debe suprir las carencias del paciente. Es decir, si la persona solo puede hacer un 25% de la fuerza que tendría en condiciones sanas, el otro 75% lo hará el robot. No obstante el fisioterapeuta encargado de las tareas de rehabilitación puede elegir poner un nivel de asistencia menor al 0,75 que se necesitaría en este caso para que el paciente se esfuerce en conseguir realizar el movimiento.

El **tiempo de movimiento t** es el tiempo que se tarda en hacer cada uno de los movimientos de los ejercicios. Es decir, si un ejercicio tiene n movimientos, el tiempo total de simulación será de nt segundos. Cuanto menor sea t más rápido se hará el movimiento, con los que las velocidades y aceleraciones serán mucho más grandes y, en consecuencia, pares debido a coriolis a la inercia serán mucho mayores.

Una vez elegidos todos estos parámetros de entrada, el archivo de inicialización cargará los parámetros del objeto robot llamado sistema. Los parámetros de este objeto son los mismos que los del robot pero, además, se le añade un porcentaje de los parámetros del paciente. Este porcentaje será el **nivel de asistencia α** . En modo seguidor el objeto robot y el objeto sistema serán una copia exacta, y cuando el nivel de asistencia esté al 100% sus parámetros serán una suma del objeto robot y el objeto paciente.

3.4.2.- Sistema

Los archivos de Simulink se ejecutan dentro de un bucle que se repetirá tantas veces como movimientos hayan dentro de un ejercicio. Es decir, si la matriz de movimiento q tiene n movimientos, la primera vez que se ejecute el bucle el modelo Simulink efectuará los cálculos para moverse de la posición inicial **$q(1,:)$ hasta la posición final $q(2,:)$** , la segunda vez lo hará desde la posición inicial **$q(2,:)$ hasta $q(3,:)$** y esto se repetirá hasta que la posición final sea la n de la matriz q , es decir **$q(n,:)$** . Esto se debe hacer debido a que el bloque que calcula la trayectoria (figura 3.1) entre las posiciones

iniciales y finales en cada movimiento solo permite dos entradas: q_i donde comienza el movimiento y q_f donde acaba. Con el bucle se consigue simular el ejercicio seguido, movimiento a movimiento.

La figura 3.1 muestra el **sistema completo**. Como se puede ver, está dividido en tres bloques bien diferenciados: un **bloque de entrada “trayectoria”** (rojo), un **bloque para el sistema paciente** y otro para el **sistema robot**.

El primer **bloque “trayectoria”** será el encargado de dar, según el tiempo de paso y tiempo de simulación, todas las posiciones, velocidades y angulares en cada instante de tiempo para pasar de una posición inicial q_i a una posición final q_f , que serán dos movimientos dentro de un mismo ejercicio.

El **bloque paciente** es el encargado de simular la fuerza que está ejerciendo el paciente y las posiciones que este sigue, que deben ser las mismas que las del robot. En el sistema real se tendrá una medida de **EMG** o un sensor de fuerza en la muñeca del paciente que indicará la magnitud y dirección de la fuerza que está ejerciendo este.

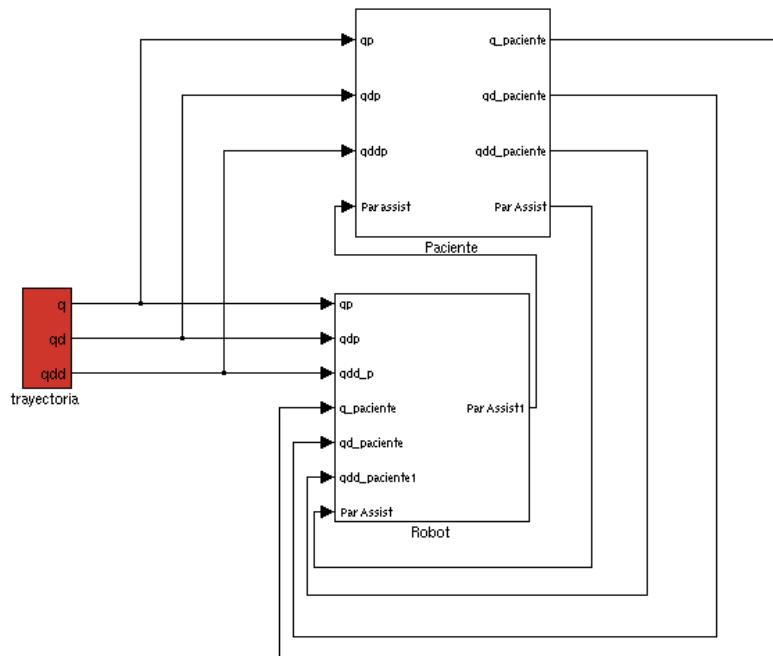


Figura 3.1: Sistema

La figura 3.2 muestra el detalle del bloque Paciente. Las entradas 1-3 son directamente las consignas de movimiento que calcula el bloque trayectoria y entran en el bloque “Modelo Paciente Sano” encargado de calcular el par que haría una persona en condiciones sanas. Esto se multiplica por una ganancia ($1 - \alpha$) que simulará la discapacidad del paciente. Este par se suma a la entrada 4: Par assist, que es la ayuda que le proporciona el robot al paciente y que se calcula en el bloque del robot. Esta suma se introduce en el bloque “Paciente con ayuda” para visualizar el movimiento del paciente y además saca la trayectoria que sigue el brazo del paciente en las salidas 1-3. La salida 4: Par assist, es el par necesario que le falta al paciente para hacer el movimiento. En los modos de asistencia/seguidor será directamente el par del paciente sano multiplicado por α .

En el caso de que el nivel de asistencia sea nulo, el par simulado del “modelo paciente sano” y el que se introduce en “paciente con ayuda” es el mismo. En estos casos la entrada y salida 4 son nulas.

La figura 3.3 detalla y describe el **bloque robot**. Este bloque recibe 6 entradas de movimiento: las **entradas 1-3** son las provenientes del bloque trayectoria mientras que las **entradas 4-6** provienen de la **salidas 1-3** del bloque paciente y se calculan de acuerdo al par que este ejerza. En el caso del robot, la trayectoria de entrada no simula ningún comportamiento real, es decir, las **entradas 1-3** se introducirán en el robot y este, mediante los modelos, calculará el par que debe aplicar. Las diferentes entradas se seleccionan según el nivel de asistencia mediante un interruptor: si

α es distinto de 0, el interruptor colocará en la salida las entradas 1-3 mientras que si $\alpha = 0$, en modo seguidor, el robot tendrá como patrón el seguir la cinemática del brazo del paciente, es decir, las entradas 4-6.

Una vez se seleccioné la trayectoria a seguir, esta se introduce en el bloque “**Modelo_Robot_solo**” que calculará el par que necesita el robot para moverse. Es decir, sería el par que necesitaría el brazo robótico si no tuviera que ayudar al paciente. Este par se suma con la entrada 7 “par assist” proveniente del bloque robot y ambos se introducen en el bloque “**Modelo Sistema**”. Este bloque simula el comportamiento del brazo robótico y además del brazo del paciente en cierta medida: según el nivel de asistencia. En el modo seguidor, con $\alpha = 0$, este sistema simplemente simulará el comportamiento del robot. El par que le entra a este bloque, es el mismo que le entraría al brazo robótico en la realidad. Una vez se calcule la trayectoria del sistema se pasa de esta a otro bloque para que calcule de nuevo el par. A este par se le resta el par que utiliza el robot para moverse a si mismo y se saca la salida 1 Par assist que irá al bloque del paciente. Este último paso, donde se vuelve a calcular el par que hace el sistema, pero esta vez utilizando la cinemática, es necesario para calcular el par de asistencia. Si no se hace así no habrá interacción entre los dos bloques y podría ocurrir que un problema con el bloque robot no sea percibido por el bloque paciente.

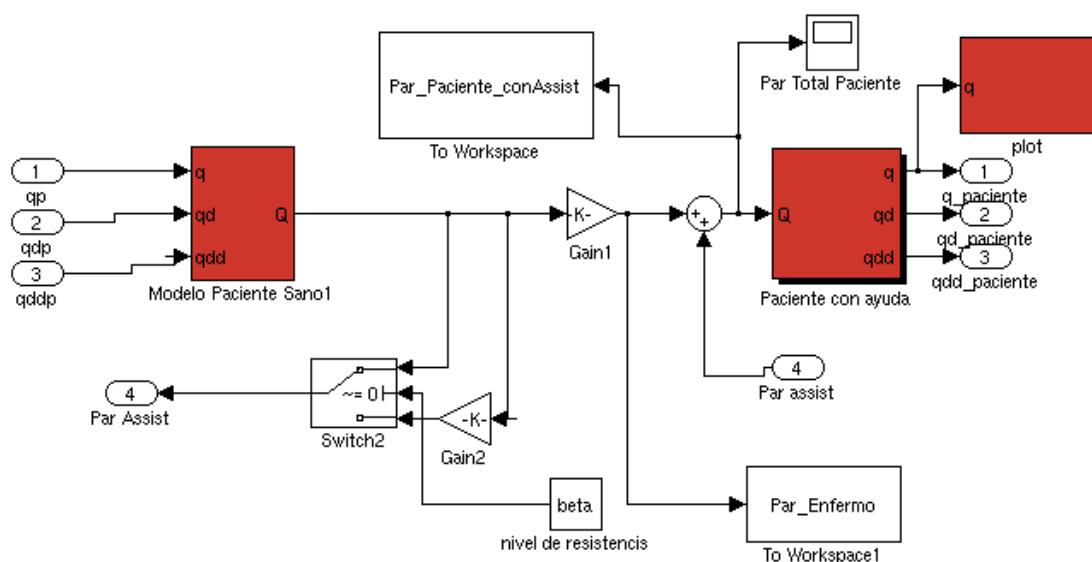


Figura 3.2: Bloque Paciente

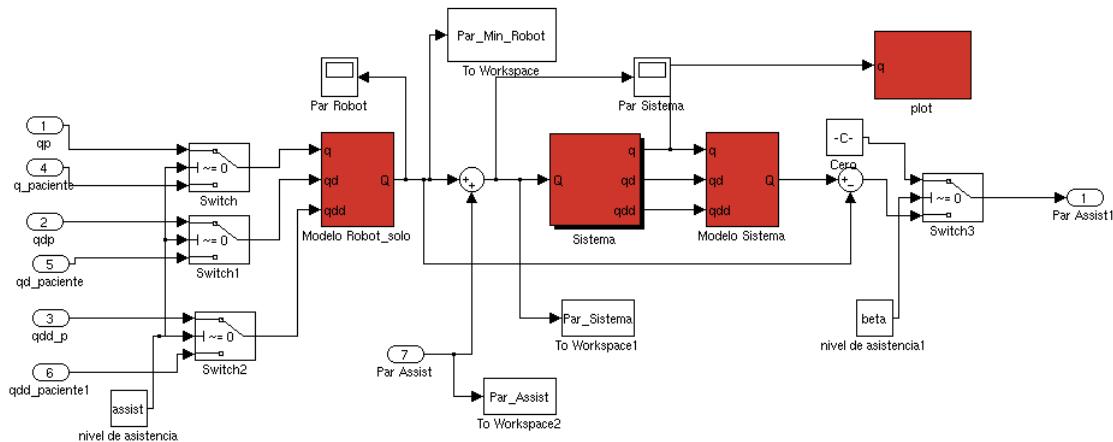


Figura 3.3: Bloque Robot

3.5.- Modo Resistencia

3.5.1.- Inicialización y entrada

Este modo tendrá prácticamente las mismas entradas que el modo asistencia pero en vez de tener un nivel de asistencia α se tiene un **nivel de resistencia β** , además de los parámetros de los tres objetos robot: **paciente, robot y sistema, cambian**. Cabe señalar que el nivel de resistencia solo se propagará a estos parámetros si $\alpha = 0$. Esto se comprobará en el archivo de inicialización. Para utilizar el modo de seguimiento basta con hacer $\beta = 0$.

En cuanto a los cambios en los parámetros de los robot, en este modo es el paciente el que tiene que hacer más fuerza de lo normal, como si un entrenamiento de pesas se tratase. Por tanto se le añade a los parámetros de masas, inercias y fricciones del paciente un porcentaje extra establecido por β mientras que en los parámetros del brazo robótico se le restará este término y el término propio del paciente. La suma de los dos, será el par necesario que necesitaría el robot para moverse a si mismo.

3.5.2.- Sistema

El sistema es el mismo que se ha descrito en las figuras 1-3, sin embargo en el apartado anterior no se describieron algunos elementos que solo tienen sentido en este modo. Para empezar en el bloque paciente, figura 3.2, a la salida 4 “par assist” le precede un interruptor (detalle en la figura 3.4). Si $\beta = 0$ significa que se está en **modo asistencia**, o en **modo seguidor**. En este caso el interruptor dejará salir la simulación del par de asistencia que necesita.

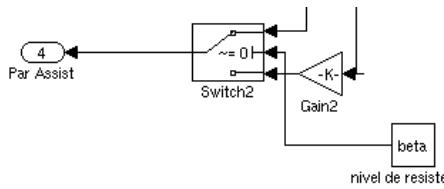


Figura 3.4: Detalle Interruptor Modo Resistencia Bloque Paciente

En caso de que β no sea cero, el par del paciente será nulo, pues $\alpha = 0$, y por tanto el “Par assist” será esta vez el par que está haciendo el paciente, que será el de su propio peso más el extra que se ha indicado con el parámetro β . Esto es necesario puesto que el robot no tiene la fuerza suficiente para sostenerse a si mismo, por lo que necesita la aportación de este par de más que está ejerciendo el paciente.

En el bloque robot el mismo interruptor es utilizado de otra manera. Al igual que en el caso anterior el interruptor se pone antes de la salida “Par assist” pero esta vez, cuando $\beta \neq 0$ esta salida será una constante nula.

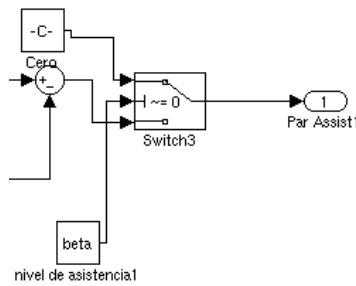


Figura 3.5: Interruptor Modo Resistencia Bloque Robot

Esto es debido a que la salida 1, en los modos de asistencia y seguimiento, calculaba la diferencia entre el **robot sistema** y el **robot normal** que es, en ese caso, el par de asistencia que ayudará al paciente. En el caso de resistencia, tal y como se han definido los parámetros, esta diferencia no es más que el extra de fuerza que debe hacer el paciente, la resta, a parte de ser negativa no interesa que se le sume al paciente debido a la entrada 4 del bloque paciente. A modo de simulación esta salida se hace cero en el modo resistencia.

3.5.3.- Salidas

En primer lugar se debe aclarar que existen dos modelos Simulink distintos para el mismo sistema. El que aparece en las figuras 3.2 y 3.3 es el sistema “sistema_f.mdl” que cuando se ejecute dibujará dos brazos robóticos representando el brazo del paciente y el brazo del robot/sistema (figura 3.6). Debido a que la carga de computación de este sistema es bastante grande, se ha diseñado otro sistema llamado “sistema_noPlot.mdl” que arranca el mismo sistema pero sin dibujar.

El resto de salidas del sistema son las gráficas con los pares involucrados en el sistema. Así se tienen como salidas: el **par que ejerce el paciente enfermo**, el **par que necesita el brazo robótico para moverse a si mismo**, el **par que le falta al paciente para poder moverse**, el **par que hace el sistema** y por último el **par que hace el paciente sumado con aquel de ayuda** que le proporciona la ortesis. Todas estas salidas se grafican con respecto al tiempo durante el tiempo completo que dure la simulación (*nt*). La figura 3-7 muestra un ejemplo de estas gráficas.

Cabe destacar que se pueden obtener otras salidas del sistema como son: la **trayectoria articular**, **velocidades** y **aceleraciones articulares** para los tres robots además de lo mismo en coordenadas cartesianas. Sin embargo se han elegido los pares ya que estos demuestran el correcto funcionamiento del sistema en conjunto.

Como diferencia entre los modos de asistencia y resistencia, en el modo de asistencia se graficará el par que necesita el paciente y hace el robot de más. En el modo resistencia esto no tiene sentido por lo que no se grafica.

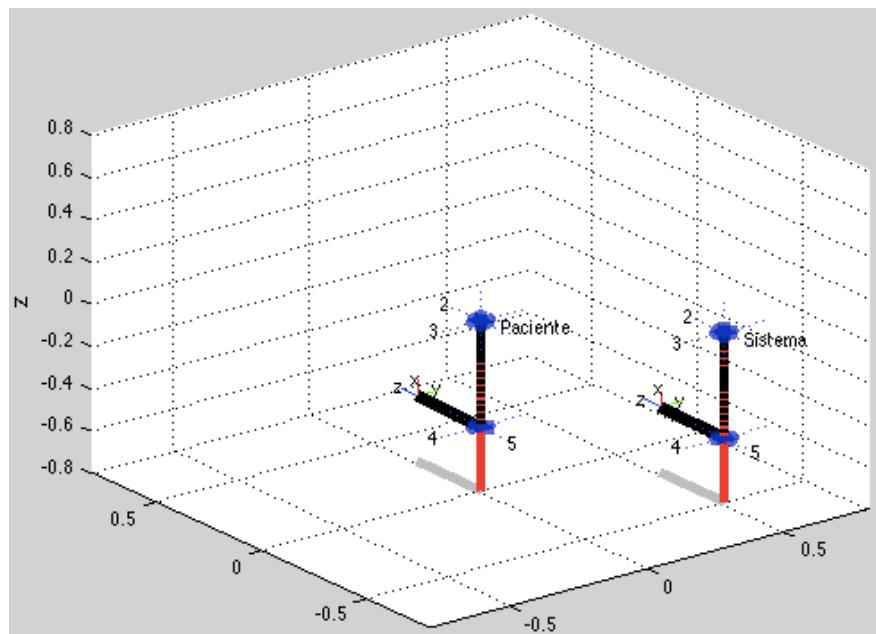


Figura 3.6: Ejemplo del dibujo de ambos brazos

3.5.4.- Control

Como se dijo en la introducción a este capítulo, en el sistema pueden aparecer diferentes errores: **perturbaciones del robot o paciente**, **errores en el modelado** o **errores en la aplicación del modelo** al sistema real. La figura 3.8 muestra el diseño del circuito de control. Para simular las interferencias o posibles incongruencias con el par que se requiere en la entrada del robot, se le añade, al par calculado, una señal de ruido blanco de potencia 1 con una frecuencia de muestreo igual a la del sistema. Además se le añade un retraso justo antes de la entrada.

El control se pone a la salida de las del bloque “Sistema” que comparará la posición y velocidad del robot actual con aquella que debería tener en realidad. Este error se pasa por dos

controladores, uno para la posición, otro para la velocidad cuya salida se suma al par que entra al bloque “sistema”. Como se ve en la figura 3.8, el par “sistema” es la suma de tres: par mínimo del robot, par de asistencia del paciente y este par de control.

Ambos controladores son del tipo Proporcional, los parámetros y el programa que inicializa esta simulación del control se detalla en el anexo correspondiente. Se ha elegido este control ya que un controlador proporcional integral/diferencial no mejoraba el error cometido entre el valor de par

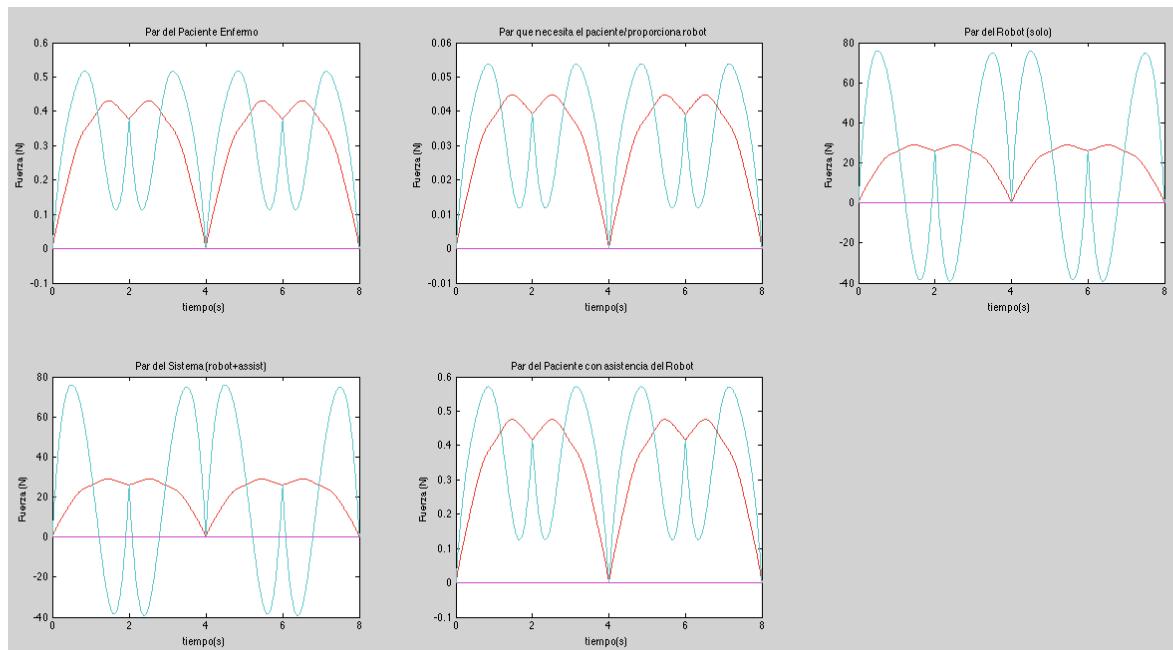


Figura 3.7: Gráficas ejemplo de salida en el sistema

esperado y el que se obtenía a la salida.

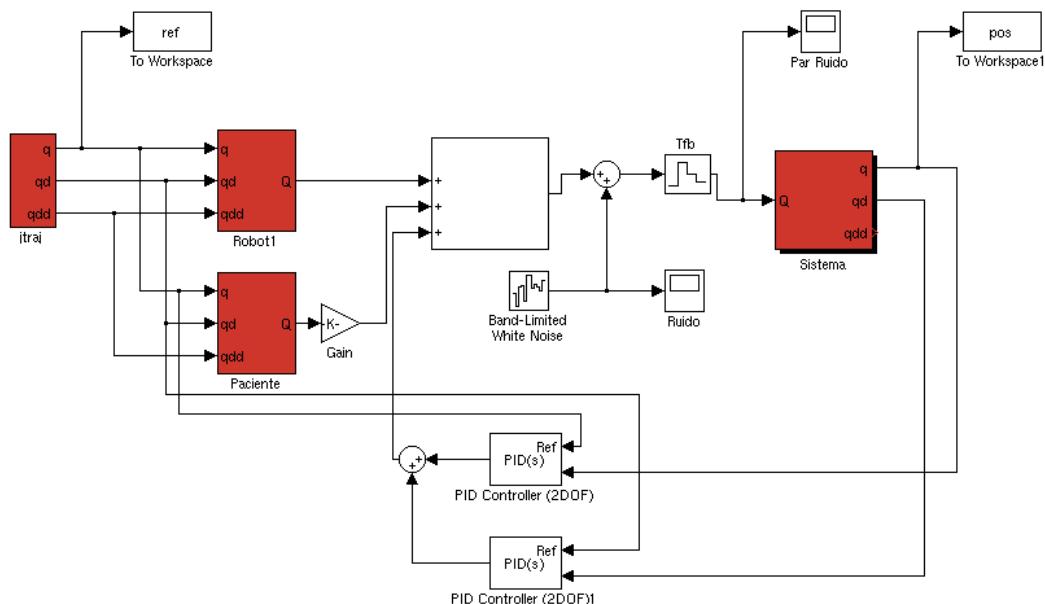


Figura 3.8 Sistema de Control

CAPÍTULO 4.- Resultados

En este capítulo se mostrarán y comentarán algunos resultados del sistema en los diferentes modos. Estos resultados son solo un ejemplo de todas las posibles combinaciones posibles pero abarcarán como mínimo cada uno de los modos. De esta manera habrán resultados para el modo asistencia, manual y automático, modo seguimiento y por último modo resistencia.

También se mostrarán algunos resultados característicos del sistema de control donde se verá la entrada ruidosa que se le introduce al sistema, la salida esperada y la salida real, además del error que se comete.

4.1.- Modo Asistencia - Manual

En esta simulación se simuló el **ejercicio de flexión donde el paciente flexiona el bíceps dos veces**:

- ▣ Primero se mostrarán los resultados en el submodo manual con un nivel de asistencia prefijado del 50% y un tiempo de movimiento de dos segundos con un paso de 0.01 segundos.
- ▣ Posteriormente se repetirá la simulación incrementando el tiempo de movimiento a 10 segundos para poder observar y comentar las diferencias de par debido a las fuerzas de coriolis y fuerzas iniciales.

Las figuras 4.1 y 4.2 muestran las dos posiciones entre las que se moverá el sistema en este ejercicio, algunas también coinciden con otro movimiento simulado de alcance descrito posteriormente.

La figura 4.3 muestra los resultados de esta simulación. Debido al poco tiempo entre movimientos, las fuerzas iniciales y de coriolis ejercen una gran influencia sobre al par motor de cada articulación.

Esto es normal teniendo en cuenta que el sistema sufre una aceleración máxima de 3,4 rad/s² y una velocidad de 2,21 rad/s.

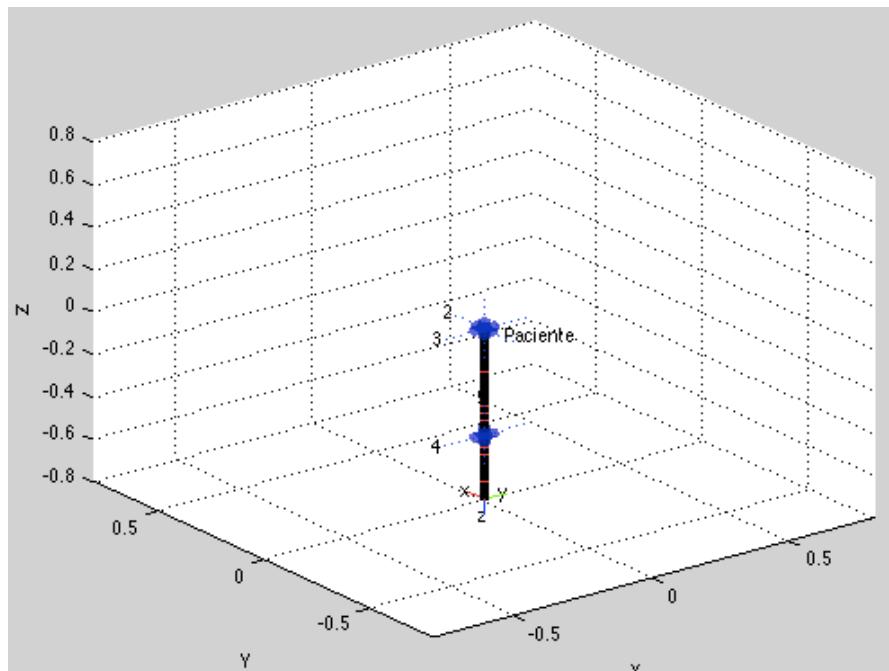


Figura 4.1: Posición Reposo inicial $q(1,:)$ y final de los movimientos Alcance y Flexión

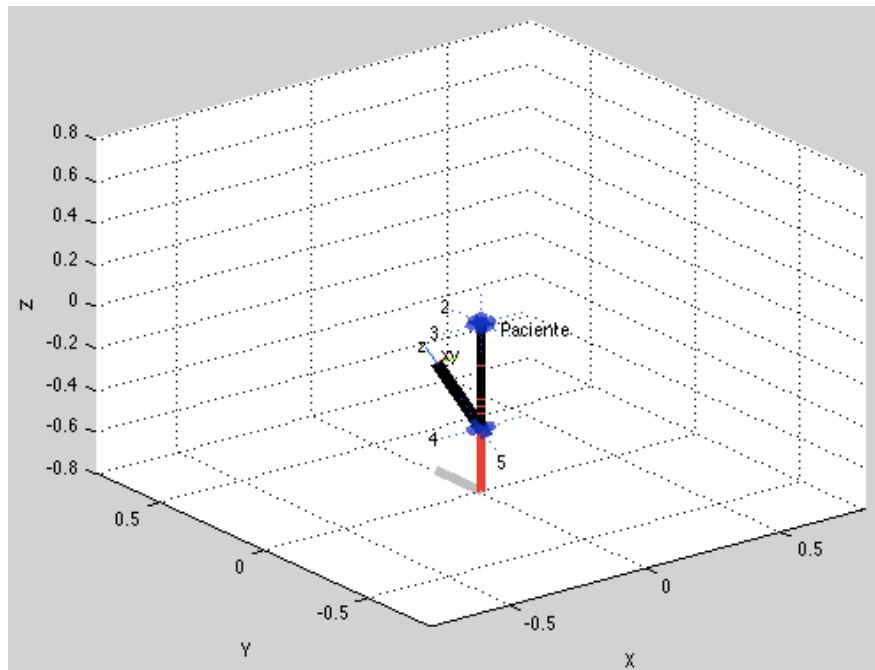


Figura 4.2: Flexión de Bíceps posición $q(2,:)$ y $q(4,:)$ del movimiento Flexión

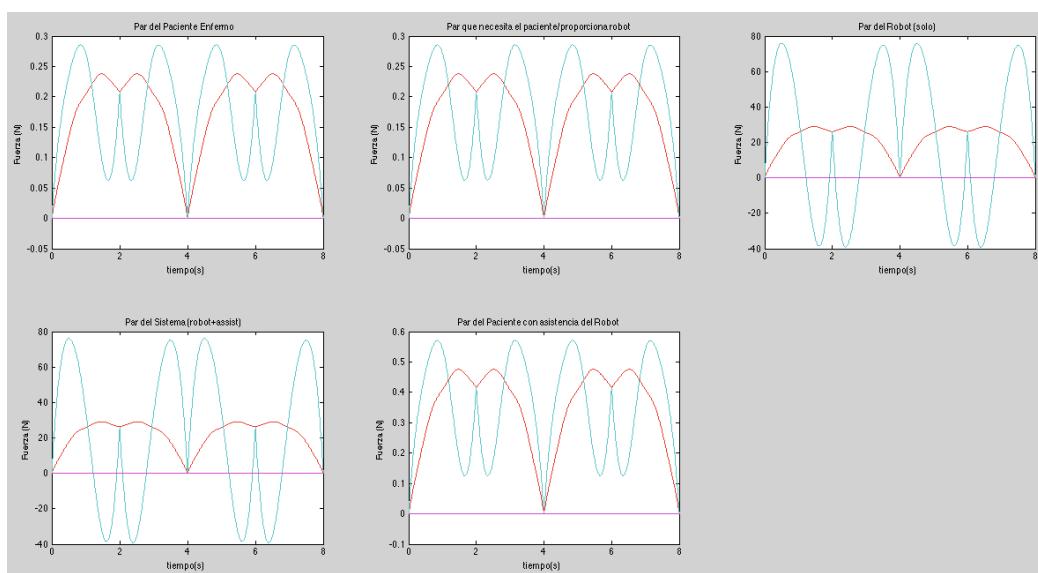


Figura 4.3: Resultados Simulación Flexión. $t = 2s$, paso = 0.01s, $\alpha = 0.5$

Repetiendo el experimento pero para $t = 7$ segundos, en la figura 4.4 se muestra como cambian los resultados. Para empezar las gráficas tienen menos saltos, son más suaves. Esto es debido a que los términos inerciales y de coriolis ya no tienen tanto efecto, y se puede comprobar al observar la aceleración máxima que es de: 0.278 rad/s^2 y la velocidad máxima de: 0.63 rad/S , muchísimo menores que en el caso anterior. Además se tienen menos pares máximos, antes el robot llegaba casi a los 80 Nm mientras que ahora apenas a los 35 Nm : el robot solo tiene que compensar, prácticamente, el efecto de la gravedad. Además, en el caso del par del sistema se observa la existencia de pares negativos durante la primera simulación, cosa que desaparece al hacer la simulación más lenta.

Analizando por otro lado los pares, se ve que una persona necesita llegar hasta un par de 0.55 Nm en la articulación 4 (línea turquesa) para realizar el movimiento de flexión y que solo hace la mitad, indicado por el nivel de asistencia. El robot tiene que hacer, solo para moverse a si mismo un par máximo de 35 Nm en su articulación 4 y luego se le añade un par de 0.275 Nm que es el 50% de lo que haría una persona sana.

Puede resultar confuso que a pesar de que en este movimiento solo se mueva una articulación, la 4, para realizar la flexión del codo en las gráficas aparezcan dos articulaciones realizando par. La línea roja representa la articulación 3, paralela a la articulación 4 pero situada en el hombro y que controla la flexión del mismo. Cuando el robot/paciente tiene el brazo estirado, en posición reposo (0,4 y 8 segundos) no hay ningún par en la articulación 3, sin embargo cuando el brazo está flexionado se ve como la articulación 3 tiene que hacer un par. Esto es debido a que, al flexionar el brazo se produce un par articular en la articulación 3 que impide que el hombro gire hacia atrás, ya que la distancia perpendicular entre el centro de gravedad del antebrazo y la articulación 3 deja de ser cero. Este par, por tanto, está contrarrestando el efecto de la gravedad.

Otro efecto que se puede visualizar en estas gráficas es el de la fricción. Estrictamente el par máximo que deberían hacer las articulaciones 3 y 4 sería justo cuando el antebrazo y el brazo estuviesen a noventa grados, es decir, cuando la distancia perpendicular entre la articulación y el centro de gravedad del antebrazo fuese máxima. El par determinado por esta distancia, justo la mitad el antebrazo (0,15 m) y el peso del antebrazo para el robot (25 kg) sería de:

$$\tau = F \cdot L = 25 \cdot g \cdot 0.15 = 36,79 \text{ Nm}$$

Ecuación 4.1

Siendo g la aceleración debido a la gravedad ($9,81 \text{ m/s}^2$). Sin embargo, en los resultados, se observa que el par que ejerce la articulación 3 (la 4 aún muestra algunos efectos de las iniciales/coriolis) es de 36,3 Nm. Esos 0,5 Nm de diferencia son debido a la fricción de las articulaciones que ayuda a mantener el brazo.

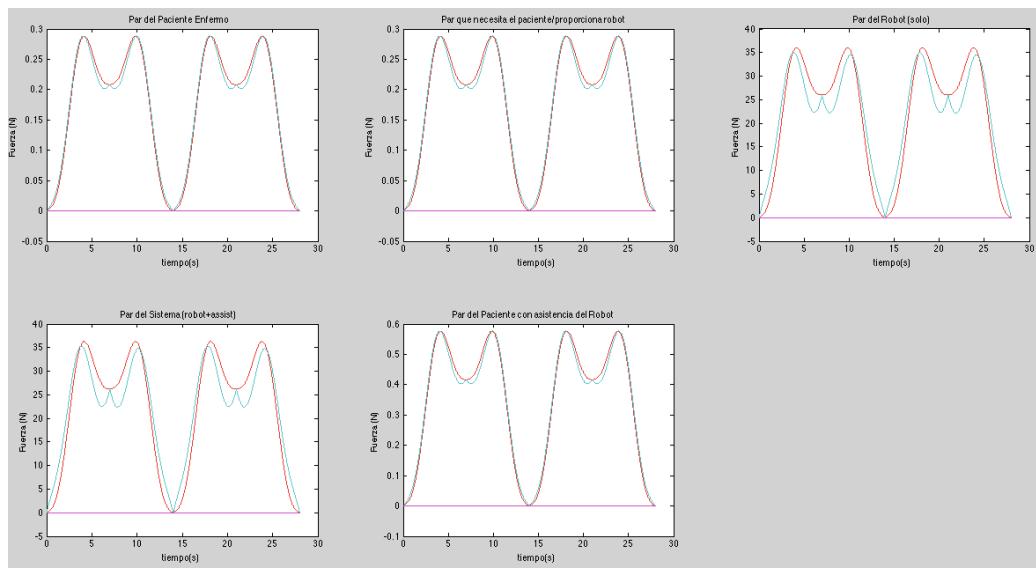


Figura 4.4: Resultados Simulación Flexión. $t = 7\text{s}$, $\text{paso} = 0.01\text{s}$, $\alpha = 0.5$

4.2.- Modo Seguimiento

Los siguientes resultados son para una simulación en modo seguimiento ($\alpha = 0$, $\beta = 0$) de un ejercicio de alcance donde el paciente intentará coger un objeto situado a cierta distancia con el brazo estirado. Las figuras 4.1, 4.5 y 4.6 muestran las posiciones características de este ejercicio y entre las que se moverán los brazos.

Para esta simulación el tiempo entre movimientos es de 6 segundos con un paso de 0.1

La figura 4.7 muestra los resultados obtenidos en esta simulación. Al ser modo seguimiento se comprueba que el sistema funciona correctamente: el par que necesita el paciente y que proporciona el robot es nulo. Del mismo modo el par que hace el robot para mantenerse a si mismo y el par sistema coinciden al igual que el par del paciente enfermo y el par del paciente con la ayuda del robot. El sistema funciona tal y como se ha descrito el modo seguimiento: cada parte del sistema aguanta sus propias fuerzas.

Los pares máximos para este movimiento son bastante mayores que en el de flexión ya que esta vez se mueve el brazo al completo. De esta manera el paciente ejerce un par máximo de 3,3 Nm y el robot 203,9 Nm. Debido a que el movimiento no es demasiado rápido, se tiene una aceleración máxima de 0,26 rad/s² y una velocidad angular máxima de 0,5 rad/s lo que implica que prácticamente todo el par es debido a la fuerza de la gravedad.

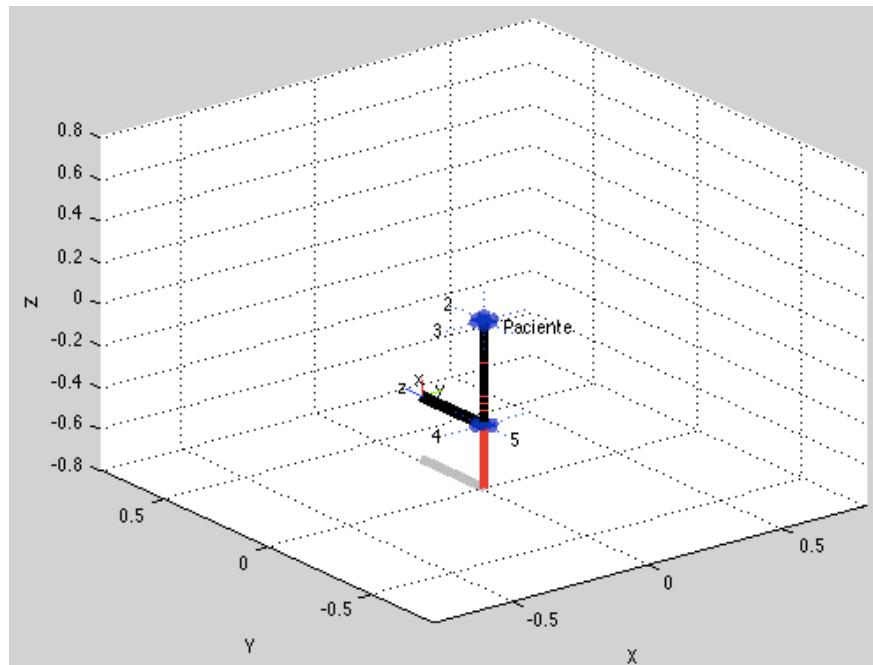


Figura 4.5: Flexión de Codo posición $q(2,:)$ y $q(4,:)$ del movimiento Alcance

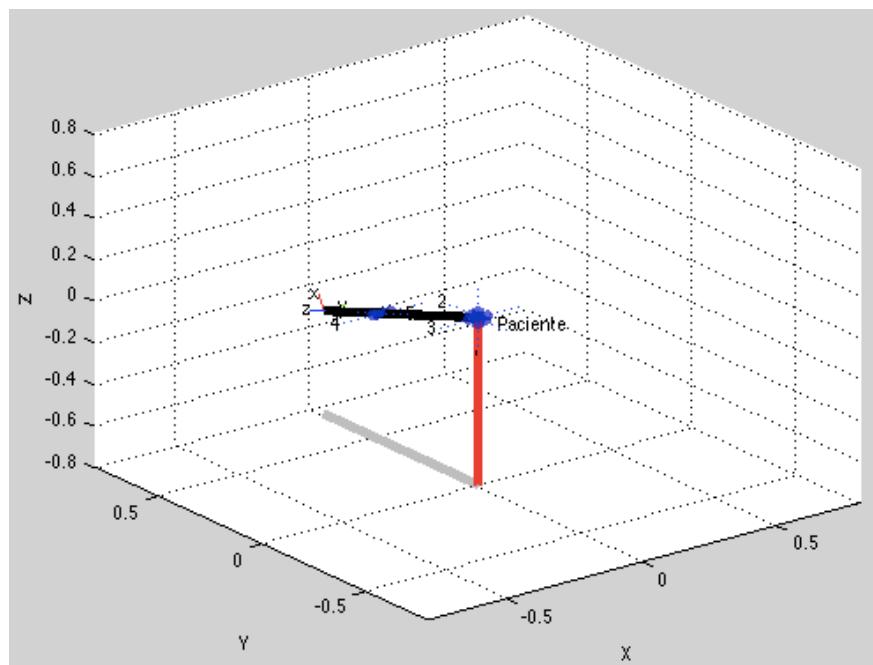


Figura 4.6: Alcance posición $q(3,:)$ del movimiento Alcance

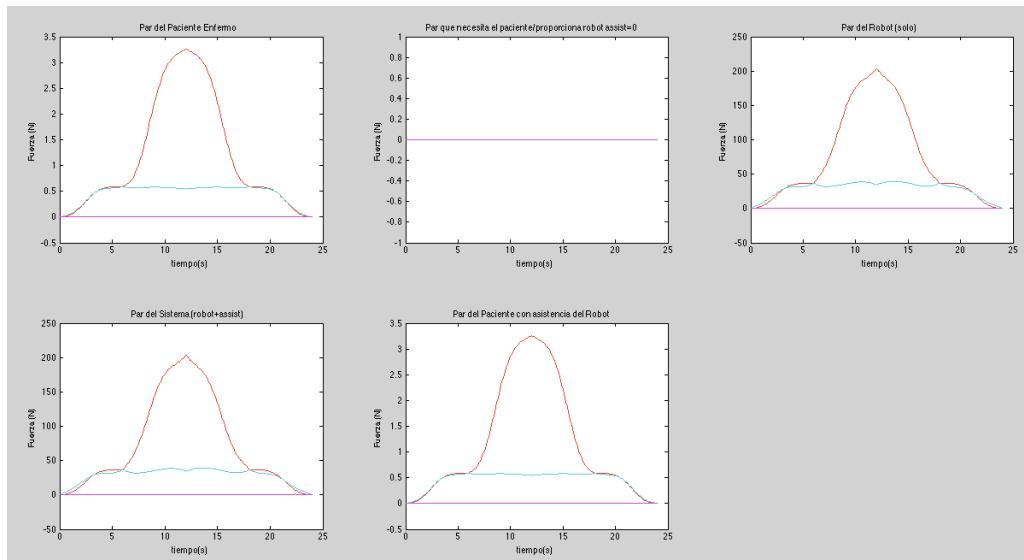


Figura 4.7: Resultados Simulación Alcance $t = 6s$, paso = 0.1s. Modo Seguimiento

4.3.- Modo Asistencia – Automático

Para estos resultados se ha repetido la simulación anterior pero utilizando el sub modo automático de asistencia donde el sistema elegirá la asistencia adecuada según la enfermedad del usuario (dado un número aleatorio entre 0 y 1). Estos resultados se compararán con los anteriores, donde se sabe que el paciente está haciendo el 100% del par, para comprobar que efectivamente funciona este modo.

La figura 4.8 resume los resultados de esta simulación. El nivel de asistencia es de 0,18528 y la enfermedad de 0,8147. La suma de ambos es $0,99998 \approx 1$. Con los resultados de la figura 4.7 se ve cuento par máximo hace un paciente sano: 3,3Nm. Si se le aplica el nivel de asistencia:

$$\tau_{asistencia} = \tau_{sano} * \alpha = 3,3 * 0,185 = 0,611Nm$$

Ecuación 4.2

Se comprueba que el par máximo de la segunda gráfica es de 0,6045Nm. Por lo que existe un error de 1%. El par máximo que debe hacer el robot para sostenerse a si mismo es de 203,9 Nm, mientras que el par que debe hacer para ayudar al paciente con este nivel de asistencia es de 204,5 Nm. Ocupando el 0,6 Nm que necesita el paciente. Por último, el par que realiza el paciente con ayuda del robot es de 3,26 Nm, 0,04 Nm menos que un paciente sano debido al pequeño error.

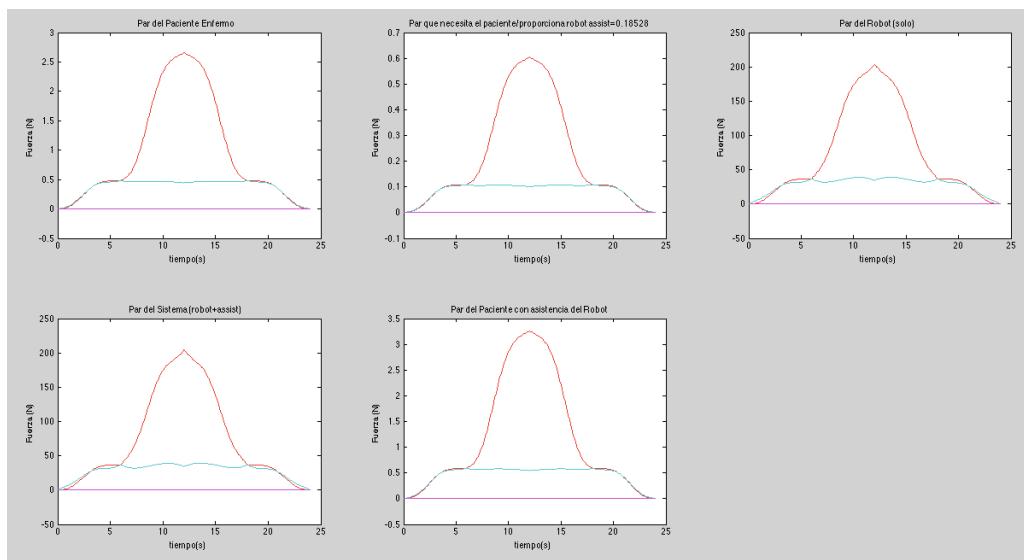


Figura 4.8: Resultados Simulación Alcance, $t=6s$, paso=0.1s, Modo Asistencia Automático asistencia=0,19

4.4.- Modo Resistencia

Los últimos resultados se harán para el modo resistencia. Se repetirá el ejercicio anterior para poder comparar las diferencias. El nivel de resistencia $\beta = 1$.

La figura 4.9 muestra los resultados para esta simulación. Lo primero que destaca es que, en comparación con la figura 4.7, la persona ha pasado de hacer, aproximadamente, el doble de lo que hacía antes (6,5 Nm). Básicamente esto es parecido a lo que haría la persona si agarra una pesa de 1kg. El robot, por su parte, hace 197,4 Nm menos que los 203,9 Nm que necesitaría para realizar el movimiento. Esto significa que el robot ejercerá una fuerza contraria al movimiento sobre el paciente. El par sistema es exactamente 203,9 Nm que es lo que necesita el robot para moverse.

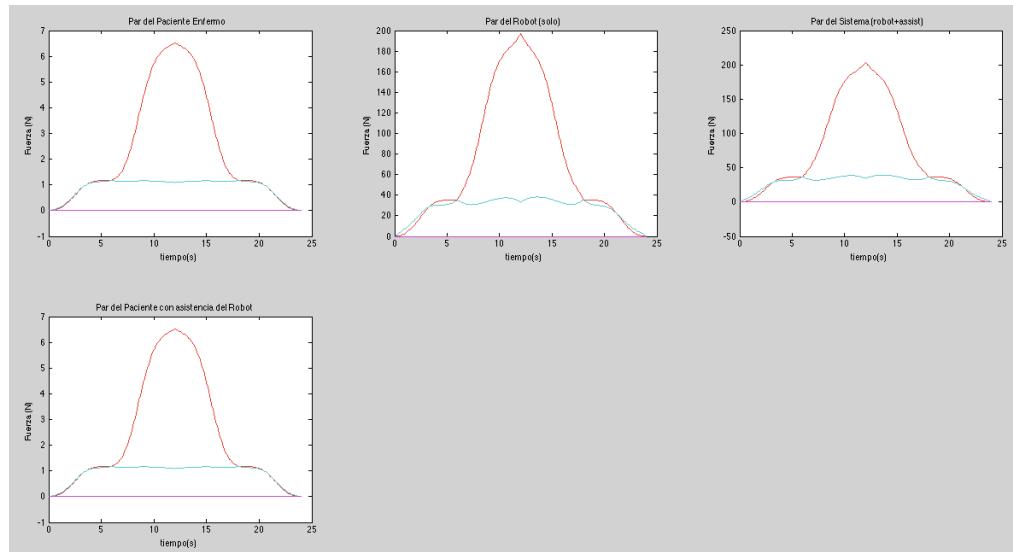


Figura 4.9: Resultados Simulación Alcance $t=6$, $\text{paso}=0.1$, modo resistencia

4.5.- Conclusiones y comentarios sobre los resultados

Los resultados anteriores se han obtenido de la simulación de todos los modos posibles para comprobar el correcto funcionamiento del sistema. A pesar de que solo se han simulado dos ejercicios (flexión y alcance) el sistema es capaz de funcionar en estos modos y de calcular los pares necesarios sea cual sea el caso. Estos resultados pueden ser importantes a la hora de programar nuevos ejercicios o de establecer la duración de los mismos. Los pares de coriolis e iniciales, si el movimiento es muy rápido, pueden influenciar al par necesario para mover el robot lo que podría sobrepasar el par máximo admisible por los motores o incluso llegar a poner en riesgo el sistema y el paciente. Una simulación previa puede determinar el tiempo t mínimo de cada ejercicio. Durante el ejercicio de asistencia automática apareció un pequeño error de un 1% que podría cambiar los resultados pero este error se atribuye al redondeo del número aleatorio que se utiliza para simular el nivel de enfermedad.

4.6.- Control

La simulación del sistema de control se hace sobre el ejercicio de alcance. El nivel de asistencia se ha puesto al 50% aunque esto no afecta para nada al control. La primera simulación se hace para un tiempo de 7 segundos.

La figura 4.10 muestra en primer lugar la posición articular, en radianes, de las distintas articulaciones. En línea discontinua aparecen las posiciones que debería seguir en robot y en línea continua la que realmente sigue. Como muestran las figuras no se pueden distinguir diferencias entre ambas. Sin embargo, la figura 4.11 muestra el error existente entre ambas que se ve es de, como máximo, de -4×10^{-3} radianes.

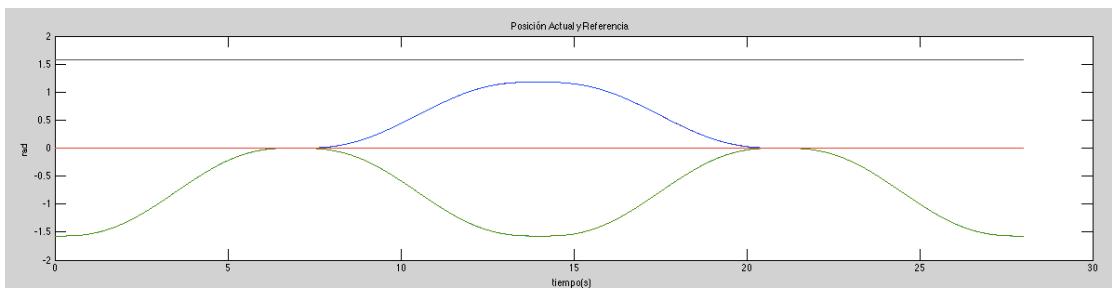


Figura 4.10: Posiciones Articulares Reales y de Referencia en un sistema con control

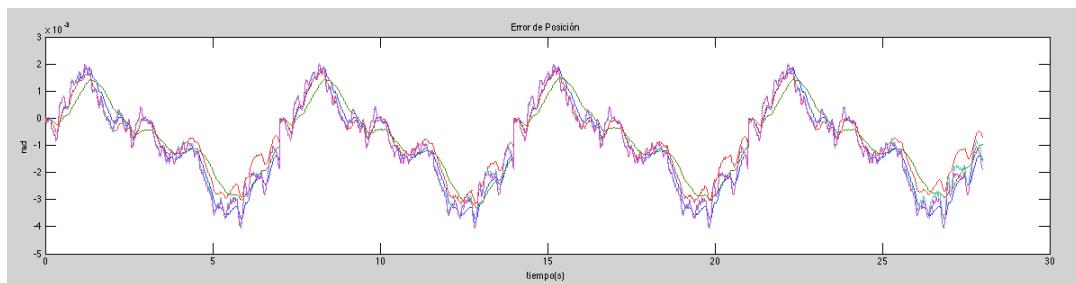


Figura 4.11: Error en las posiciones articulares en un sistema sin control

Las figura 4.12 y 4.13 muestran las diferencias que existen entre el par calculado y el par tras la suma del ruido blanco. Debido al control, el par calculado muestra un pequeño rizado, una muestra de que el par se está regulando.

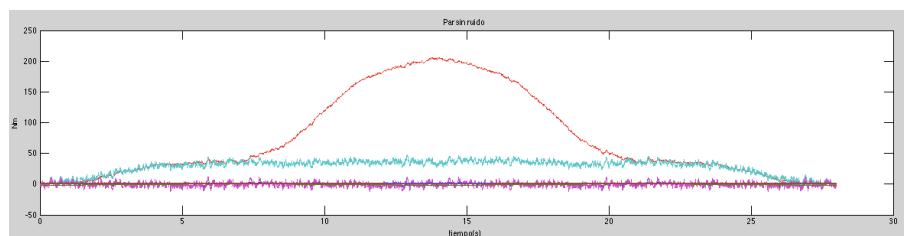


Figura 4.12: Par calculado y Par de control

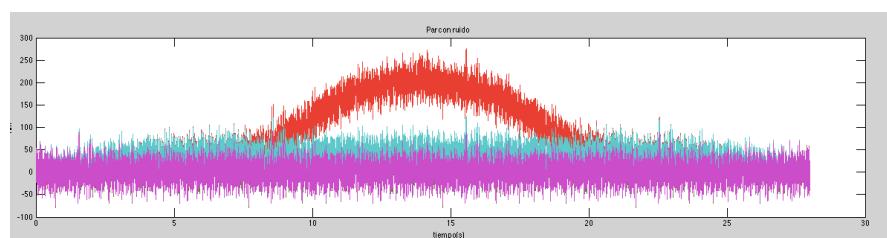


Figura 4.13: Par Calculado y Ruido Blanco

Para una mejor comparativa, la figura 4.14 muestra las posiciones articulares y reales pero sin el control para una simulación con el mismo tiempo de 7 segundos por movimiento.

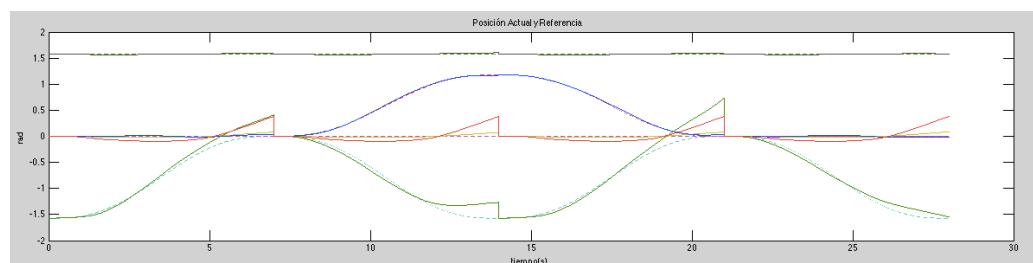


Figura 4.14: Posiciones Articulares Reales y de Referencia en un sistema sin control

Como se visualiza en la figura 4.14, las diferencias ahora son más notables. Se observa además que cada 7 segundos, el tiempo que tarda cada movimiento del ejercicio, hay un salto. Esto es debido a que la simulación reinicia los parámetros de la posición en cada ejercicio. Al final de cada movimiento, el error es cada vez más grande pero al reiniciarlo, vuelve a 0.

En la figura 4.15 se muestra el detalle del error que se comete. Esta vez se llega hasta 0.5 radianes, 100 veces superior al error con el control.

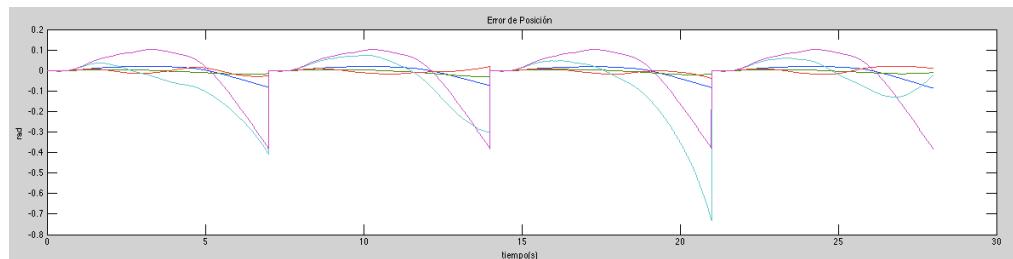


Figura 4.15: Error en las posiciones articulares en un sistema sin control

Por último, la figura 4.16 muestra el par de control que inyectan los controladores en el sistema. Esto es necesario para ver si este tipo de control es posible: si los controladores inyectan un par demasiado grande para compensar los errores este sistema podría ser inviable físicamente, donde los motores no podrían generar cambios tan bruscos de par.

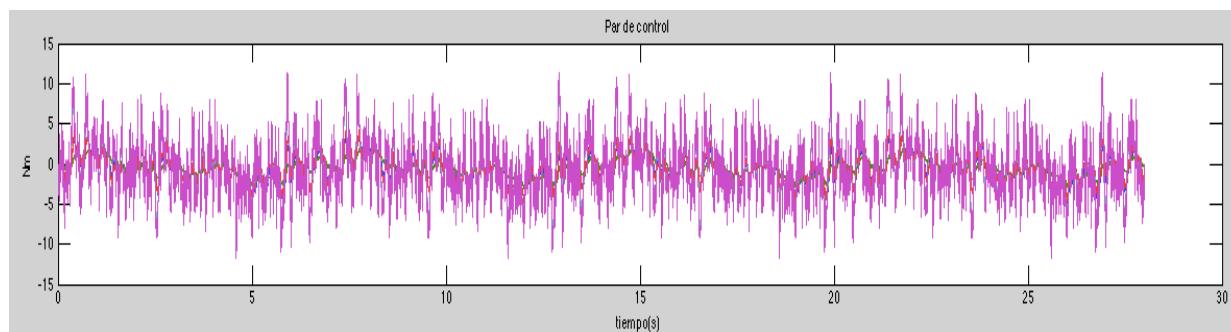


Figura 4.16: Par de Control

Como muestra la figura, el par de control llega, como máximo a 10 Nm, un par que cualquier motor puede suministrar con facilidad.

4.7.- Resultados Experimentales

A continuación se presentan algunos de los resultados finales obtenidos en la sesión de experimentación realizada en el instituto de Bioingeniería de Cataluña (IBEC). Esta sesión de experimentación queda detallada en el Anexo 05.

Las siguientes gráficas muestran la relación obtenida entre el trabajo realizado por el paciente, en Julios, y los niveles de asistencia (figura 4.17) y resistencia (figura 4.18). Otros resultados y las conclusiones obtenidas de esta experimentación quedan escritas en el Anexo 05.

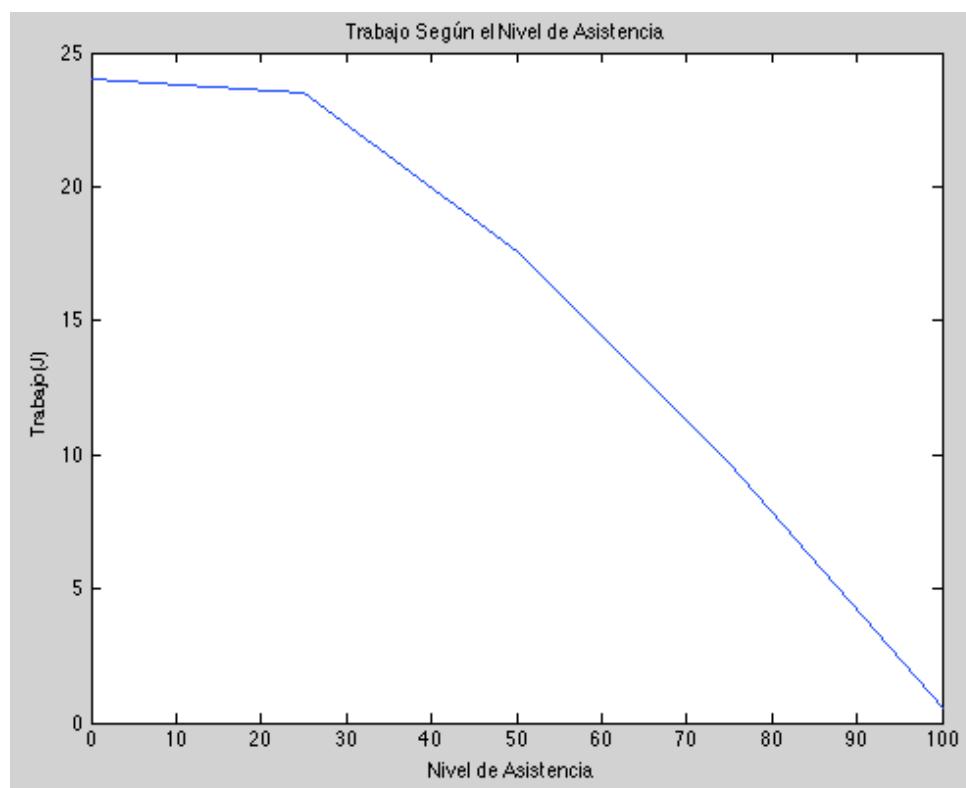


Figura 4.17: Trabajo medio realizado según el nivel de asistencia

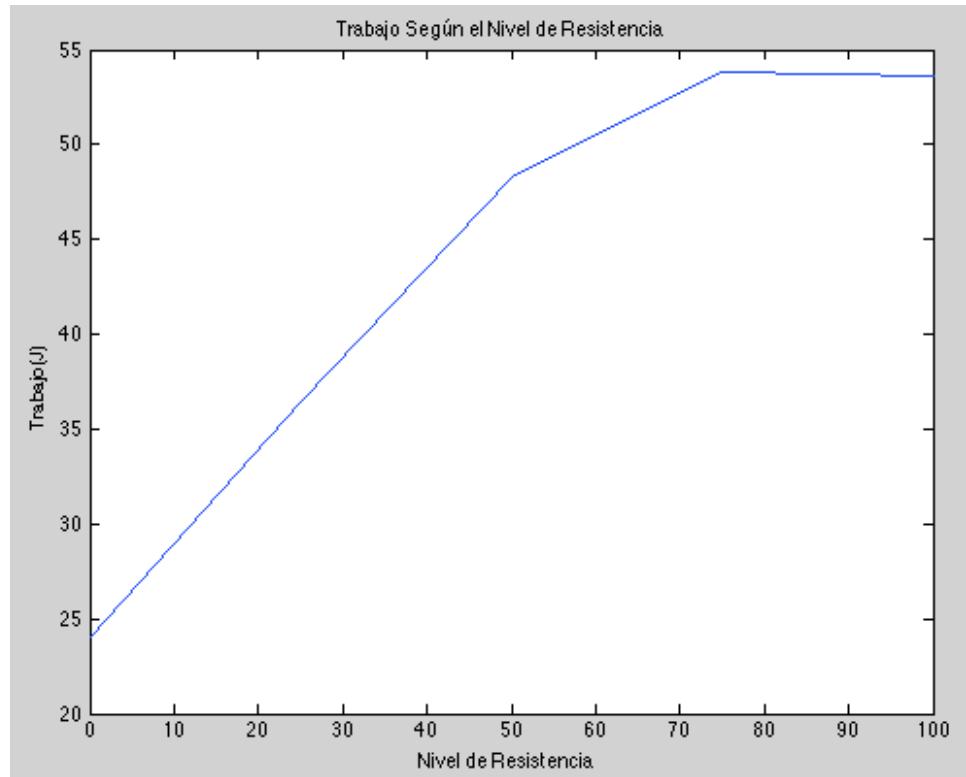


Figura 4.7: Trabajo medio realizado según el nivel de resistencia



Bibliografía

- [1] Peter Corke, 2011, Robotics, Vision and Control. Fundamental Algorithm in Matlab; Heidelberg, Alemania, Springer.
- [2] Xavier Giralt, Luis Amigo, Alicia Casals, Josep Amat, Pendiente de Publicación, Robotic Platform to Evaluate the Assistance and Assessment on the Rehabilitation Loop, Barcelona, España.
- [3] Hyper project, “Hybrid Neuroprosthetic and Neurorobotic Devices for Functional Compensation and Rehabilitation of Motor Disorders”. CSD2009-00067.



ÍNDICE DE FIGURAS

Figura 2.1: Resumen Entradas y Salidas del Sistema	4
Figura 2.2: Detalle del Sistema en el Modo Asistencia	4
Figura 2.3: Detalle del Sistema en modo seguimiento	5
Figura 3.1: Sistema	9
Figura 3.2: Bloque Paciente	10
Figura 3.3: Bloque Robot	10
Figura 3.4: Detalle Interruptor Modo Resistencia Bloque Paciente	11
Figura 3.5: Interruptor Modo Resistencia Bloque Robot	11
Figura 3.6: Ejemplo del dibujo de ambos brazos	12
Figura 3.7: Gráficas ejemplo de salida en el sistema	13
Figura 3.8 Sistema de Control	13
Figura 4.1: Posición Reposo inicial $q(1,:)$ y final de los movimientos Alcance y Flexión	14
Figura 4.2: Flexión de Bíceps posición $q(2,:)$ y $q(4,:)$ del movimiento Flexión	15
Figura 4.3: Resultados Simulación Flexión. $t = 2s$, $paso = 0.01s$, $\alpha = 0.5$	15
Figura 4.4: Resultados Simulación Flexión. $t = 7s$, $paso = 0.01s$, $\alpha = 0.5$	16
Figura 4.5: Flexión de Codo posición $q(2,:)$ y $q(4,:)$ del movimiento Alcance	17
Figura 4.6: Alcance posición $q(3,:)$ del movimiento Alcance	17
Figura 4.7: Resultados Simulación Alcance $t = 6s$, $paso = 0.1s$. Modo Seguimiento	18
Figura 4.8: Resultados Simulación Alcance, $t=6s$, $paso=0.1s$, Modo Asistencia Automático asistencia=0,19	18
Figura 4.9: Resultados Simulación Alcance $t=6$, $paso=0.1$, modo resistencia	19
Figura 4.10: Posiciones Articulares Reales y de Referencia en un sistema con control	20
Figura 4.11: Error en las posiciones articulares en un sistema sin control	20
Figura 4.12: Par calculado y Par de control	20
Figura 4.13: Par Calculado y Ruido Blanco	20
Figura 4.14: Posiciones Articulares Reales y de Referencia en un sistema sin control	20
Figura 4.15: Error en las posiciones articulares en un sistema sin control	21
Figura 4.16: Par de Control	21
Figura 4.17: Trabajo medio realizado según el nivel de asistencia	22
Figura 4.18: Trabajo medio realizado según el nivel de resistencia	22