



Anexos Proyecto Fin de Carrera

Simulación del Control de un Exoesqueleto para la
Rehabilitación de miembro superior y/o inferior

Autor

Luis Bernués Bergua

Directora

Diana Ruiz Bueno

Ponente

Luis Montano Gella

Escuela de Ingeniería y Arquitectura

2013/2014

Anexo I: Robotic Toolbox

Esta librería recoge casi todas las funciones que han permitido la simulación y el estudio de la cinemática y dinámica de tanto el objeto robot como del objeto paciente.

Estos objetos se han definido como una serie de articulaciones sucesivas, denominadas *links*, unidas por medio de segmentos en forma de barras, mediante el uso de estructuras *serial-link*.

En el caso particular de los parámetros geométricos del objeto, se utiliza un convenio que define sucesivamente las coordenadas de las articulaciones que lo forman respecto de la anterior, mediante el uso de transformadas, tanto de rotación como de traslación dependiendo de la posición de una respecto de la otra, y el tipo de grado de libertad de cada *link*. Este método se denomina de Denavit-Hartenberg (*DH*). De esta manera, los parámetros anteriormente mencionados y los restantes que completan la descripción de la dinámica del robot se encapsulan dentro de esta estructura, donde estos datos quedan registrados y almacenados, formándose así, un objeto compacto con el que más sencillo y ordenado trabajar.

En cuanto a las ecuaciones de movimiento que nos permiten proponer soluciones a los objetivos que se plantean en este documento, la Robotic Toolbox también ofrece una serie de funciones para solucionar tanto la cinemática como la dinámica, en ambos casos, de forma directa o inversa. Además, contiene también múltiples funciones adicionales que permiten un análisis más detallado de la dinámica del objeto a lo largo de una trayectoria específica. Como por ejemplo, la descomposición de la aportación de cada uno de los términos que participan de la ecuación de movimiento, es decir, que parte de la energía resultante en cada articulación para satisfacer una determinado movimiento, es resultado de la acción de la gravedad sobre el objeto, cual es el resultado de los términos relacionados con la fricción, tanto seca como viscosa, y de otros términos que se comentan en apartados sucesivos.

Parámetros de la estructura serial-link

Estos parámetros son estándar para cualquier modelo que se quieran generar con esta toolbox, ya sean en el caso de este documento en particular, objeto robot u objeto paciente, y se ajustan según describan mejor los parámetros dinámicos reales del objeto a estudiar.

Estos parámetros se definen en función de las características de cada articulación. Para ello, se generan unas subestructuras, denominadas *links*, que recogen la información dinámicas de cada grado de libertad contenido en una articulación. Todas estas propiedades se almacenarán

finalmente en una estructura *serial-link*. El siguiente diagrama representa le método de almacenamiento de propiedades para generar la estructura.

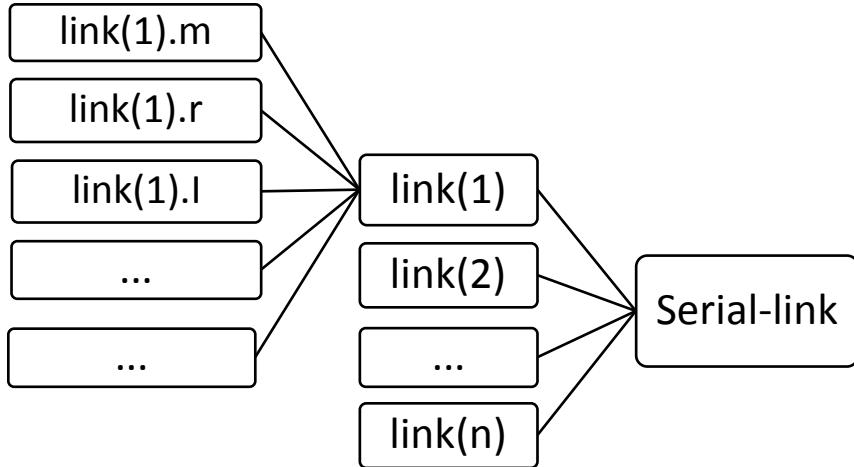


Figura 1 - Esquema de formación de la estructura serial-link

El primer aspecto que se ha de generar, es la geometría del objeto. Para ello, como se ha mencionado anteriormente, se utiliza el método representativo de Denavit-Hartenberg. Para cada *link i*, se definen los 5 parámetros necesarios para la utilización del método: la rotación alrededor del eje z_{i-1} (θ_i), la traslación a lo largo del eje z_{i-1} (d_i), la traslación a lo largo de x_i (a_i), la rotación alrededor del eje x_i (α_i) y por último el tipo de articulación, de revolución o de traslación, que indica el grado de libertad desarrollado. Como se observa, el método se basa en índices, por lo que cada link precisa de la referencia del anterior componente de la estructura objeto.

A partir de estos parámetros, es posible desarrollar un modelo cinemático que permita determinar la posición y orientación del efector final respecto de la base del objeto.

El objeto se define como una serie de links, donde cada cual desarrolla un único grado de libertad. Para un objeto de N uniones, numeradas de 1 a N , se considera que existe un *link 0* que generalmente se encuentra fijo y alojado en la base del objeto, mientras que el efector final se carga en el extremo de la barra generada por el último *link N*.

Entre los *links* se definen barras que se comportan como sólidos rígidos. Bajo esa premisa es posible definir la relación de posición entre dos *links* consecutivos.

Estableciendo un sistema coordenado diestro y unitario como marco de referencia para cada *link*, se definen las matrices de transformación homogénea A_{i-1}^i para obtener la relación de posición de un *link* respecto del *link* anterior:

$$A_{i-1}^i = \begin{pmatrix} R & a_i \cos \theta_i \\ 0 & a_i \sin \theta_i \\ 0 & d_i \\ 0 & 1 \end{pmatrix} \quad (39)$$

Donde R_{i-1}^i es la matriz de rotación:

$$R_{i-1}^i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{pmatrix} \quad (40)$$

Una vez determinadas las matrices de transformación homogénea de todos los *links* que conforman el objeto respecto del anterior, se puede determinar la posición relativa entre *links* no consecutivos mediante el producto de las matrices de transformación homogénea que los unen. Para determinar esta relación entre la base del objeto y su efecto final:

$$A_0^N = A_0^1 A_1^2 \cdots A_{N-1}^N \quad (41)$$

Una vez la geometría ha sido constituida, se asigna a cada subestructura *link(i)* los siguientes parámetros que definirán su comportamiento dinámico:

Masa, *link(i).m*: Para simplificar la dinámica de los sistemas a estudio, se aproxima la masa de cada segmento que forma el sistema como una masa puntual situada a lo largo del mismo en función del centro de gravedad correspondiente. Como pueden existir distintos links en cada articulación, se asignará este valor puntual únicamente al último link definido antes de que el segmento lo une con otra articulación, mientras que a los otros *links* que comparten la primera articulación, se les asociará una masa nula. (Por defecto, el parámetro *link(i).m* no puede ser negativo. Si lo fuese, el sistema no respondería correctamente).

Distancia al centro de gravedad (CoG), $link(i).r$: como en el caso anterior, se define únicamente para el último link, e indica la posición [x y z] del CoG, de la masa asociada en $link(i)$ considerando los ejes de $link(i)$ y utilizando como origen, en este caso en particular, el extremo de la barra definida por $link(i).a$. Es importante que queden claros como se transforman los ejes entre los links.

Momentos de inercia, $link(i).I$: define el momento asociado a la inercia rotacional de un cuerpo, y por lo tanto depende de la forma y el peso que este tenga. A falta de trabajar con especificaciones concretas de robots reales, se ha modelado la forma de las barras que constituyen el objeto robot como si fueran cilindros, utilizando las fórmulas de cálculo basadas en la dinámica de sólido rígido (la primera para el cálculo de la inercia del sólido en su dirección longitudinal y la segunda en la dirección transversal):

$$I_L = \frac{1}{2}mR^2 \quad (42)$$

$$I_R = \frac{1}{4}mR^2 + \frac{1}{12}mL^2 \quad (43)$$

Donde:

- m : masa del cilindro
- R : radio del cilindro
- L : longitud del cilindro

Inercia del motor, $link(i).Jm$: asociada al motor alojado en cada link para desarrollar la potencia necesaria para realizar un determinado movimiento por parte del objeto. Este parámetro precisa de conocer las especificaciones del robot real que se pretende estudiar. La elección de un motor real, a cuyas especificaciones se refiere este parámetro, limitará la potencia máxima a desarrollar en ese link dependiendo del tipo de motor alojado en la articulación.

Gear ratio, $link(i).G$: este parámetro establece un ratio acerca de la velocidad angular entre *links* consecutivos, y se calcula directamente a partir del número de dientes de los engranajes que unen los *links*. Utilizando el método de los trabajos virtuales, se concluye

que esta relación también establece una transmisión de momento entre los *links* que definen el objeto. Por lo tanto, en el caso de que este factor sea distinto de la unidad, la potencia relativa que se transmiten entre unos y otros no es la total. La ecuación del ratio de velocidad viene dada por:

$$R = \frac{\omega_A}{\omega_B} = \frac{N_B}{N_A} = \frac{T_B}{T_A} \quad (44)$$

Donde:

- ω_A : velocidad angular del eslabón *A*
- N_A : número de dientes del eslabón *A*
- T_A : par desarrollado por el eslabón *A*

Fricción viscosa, *link(i).B*: este parámetro define la interacción de los sistemas mecánicos, entre ellos mismos y el entorno de trabajo, relacionados en este caso con la lubricación existente entre *links*.

Fricción de Coulomb, *link(i).Tc*: indica el intervalo (2 valores positivo y negativo respectivamente) asociado al coeficiente de fricción estática, entre los que se produce una fuerza de contacto entre sistemas mecánicos debido a la fricción estática. Estos valores son proporcionales al coeficiente de fricción estática, que limita la fuerza a partir de la cual comienza a producirse movimiento relativo entre dos sistemas. Una vez que se rebasan estos límites, empieza el fenómeno de fricción dinámica. Ambos tipos de fricción son función de múltiples factores, como las características de los materiales de contacto y la temperatura de operación, y por lo tanto, se modelan en base a aproximaciones lineales del comportamiento real de este fenómeno.

Offset kinematic, *link(i).offset*: redefine, a partir de las relaciones de posición entre *links* establecidas por los parámetros de Denavit-Hartenberg, la posición del objeto donde los ángulos, en este caso en particular, son nulos (también la traslación sería nula si el objeto no fuese de revolución). No indica una situación de reposo del sistema.

Joint variable limits, *link(i).qlim*: indica los límites, máximo y mínimo, constructivos o de seguridad teóricos de los grados de libertad del objeto. Se han definido como teóricos,

ya que en la práctica, las funciones de cálculo de la dinámica directa e inversa no tienen en ningún momento estos límites en cuenta, y por lo tanto, pueden llegar a ser rebasados si las condiciones así lo requieren. Todos los parámetros mencionados anteriormente son ajustables a distintos modelos de robot, conocidas las especificaciones técnicas y de funcionamiento del mismo.

Finalmente, una vez que cada *link* está correctamente definido, se procede a encapsular toda esta información en una única estructura objeto, que represente lo mejor posible el robot con el que se quiere trabajar. Al mismo tiempo que se forma este objeto, otra serie de propiedades asociadas a este conjunto de *links* pueden ser gestionadas: como la gravedad que afecta al modelo completo (dirección, sentido y magnitud), como especificar también una serie de opciones gráficas asociadas al modelo, de manera que la visualización del mismo sea adecuada, y como otras propiedades adicionales que se explican más detalladamente en otras referencias.

Dinámica inversa (*ID*)

Para evaluar la dinámica inversa de los modelos, la Robotic Toolbox de Matlab propone una función denominada *rne.m*, basada en el algoritmo recursivo de Newton-Euler. Este método calcula las fuerzas generalizadas requeridas en las uniones para un conjunto de coordenadas, velocidades y aceleraciones generalizadas. La recursión directa propaga información cinemática desde el primer *link* hasta el efector final. Por otro lado, la recursión inversa propaga las fuerzas y los momentos que se desarrollan en los *links*, desde el efector final hasta la base. La función *rne.m* depende de los siguientes términos para el cálculo de la dinámica articular del modelo *Tau*:

$$\text{Tau} = \text{rne}(\text{Robot}, q, \dot{q}, \ddot{q}, \text{grav}, \text{Fext}) \quad (45)$$

Donde:

- *Robot*: modelo sobre el que se pretende calcular la *ID*
- *q*: posición articular del modelo
- *\dot{q}* : velocidad articular del modelo
- *\ddot{q}* : aceleración articular del modelo
- *grav*: gravedad que afecta al modelo
- *Fext*: fuerzas exteriores que afectan al modelo sobre el efector final

Dinámica directa (**FD**)

Para solucionar la dinámica directa, la Robotic Toolbox de Matlab propone una función *slaccel.m*, la cual calcula la aceleración a partir del par *Tau*, y la velocidad \dot{q}_0 y la posición articular q_0 del instante anterior (como condiciones de contorno).

Primero utiliza la función *rne.m* mencionada en el apartado anterior, definiendo la posición del instante anterior, una velocidad articular nula, una aceleración articular unitaria y estableciendo la gravedad nula. De esto modo se calcula el término asociado a la inercia del sistema M de la ecuación dinámica de movimiento para una aceleración unitaria.

Segundo, se utiliza de nuevo la misma función, pero ahora se define la posición y velocidad articular del instante anterior, una aceleración articular nula, la gravedad correspondiente que afecta al modelo y las fuerzas exteriores si las hubiese. El resultado es la cinética articular del modelo asociada a los términos derivados del acoplamiento de la velocidad, de la fricción, de la gravedad y de las fuerzas exteriores *Torque*.

Finalmente, se puede calcular la aceleración articular del modelo a partir de los resultados anteriores:

$$\ddot{q} = (\text{Tau} - \text{Torque}) / M' \quad (46)$$

Para completar la cinemática articular utiliza 2 etapas consecutivas de integración. El esquema de bloques que modela estas operaciones en el siguiente:

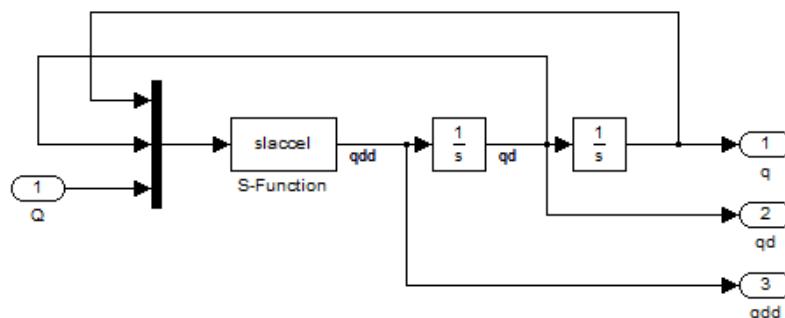


Figura 2 – Esquema Simulink para resolver FD

Matrices de transformación homogénea

Miembro superior

Considerando que todos los *links* son de revolución, θ_i representa la posición articular de cada *link*, $offset_i$ indica el offset aplicado sobre la geometría inicial que genera el método de Denavit-Hartenberg modificando entonces la referencia donde el ángulo es nulo, y L_1 y L_2 son la longitud de antebrazo y brazo respectivamente:

$$A_0^1 = \begin{pmatrix} \cos(\theta_1 + offset_1) & 0 & \sin(\theta_1 + offset_1) & 0 \\ \sin(\theta_1 + offset_1) & 0 & -\cos(\theta_1 + offset_1) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (47)$$

$$A_1^2 = \begin{pmatrix} \cos(\theta_2 + offset_2) & 0 & \sin(\theta_2 + offset_2) & 0 \\ \sin(\theta_2 + offset_2) & 0 & -\cos(\theta_2 + offset_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (48)$$

$$A_2^3 = \begin{pmatrix} \cos(\theta_3 + offset_3) & -\sin(\theta_3 + offset_3) & 0 & -L_1 \cos(\theta_3 + offset_3) \\ \sin(\theta_3 + offset_3) & \cos(\theta_3 + offset_3) & 0 & -L_1 \sin(\theta_3 + offset_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (49)$$

$$A_3^4 = \begin{pmatrix} \cos(\theta_4 + offset_4) & 0 & \sin(\theta_4 + offset_4) & 0 \\ \sin(\theta_4 + offset_4) & 0 & -\cos(\theta_4 + offset_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (50)$$

$$A_4^5 = \begin{pmatrix} \cos(\theta_5 + offset_5) & -\sin(\theta_5 + offset_5) & 0 & 0 \\ \sin(\theta_5 + offset_5) & \cos(\theta_5 + offset_5) & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (51)$$

Las matrices anteriormente mencionadas son resultado de la aplicación de los parámetros *DH* que parecen en la siguiente tabla:

Link/Parámetro	θ	d	a	α	RP
1	0	0	0	$\pi/2$	R
2	0	0	0	$\pi/2$	R
3	0	0	$-L_1$	0	R
4	0	0	0	$\pi/2$	R
5	0	L_2	0	0	R

Tabla 1 - Denavit- Hartenberg (*DH*) parámetros para extremidad superior derecha

Miembro superior

Considerando que todos los *links* son de revolución, θ_i representa la posición articular de cada *link*, $offset_i$ indica el offset aplicado sobre la geometría inicial que genera el método de Denavit-Hartenberg modificando entonces la referencia donde el ángulo es nulo, y L_1 , L_2 y L_3 son la longitud del muslo, pierna y pie respectivamente:

$$A_0^1 = \begin{pmatrix} \cos(\theta_1 + offset_1) & 0 & \sin(\theta_1 + offset_1) & 0 \\ \sin(\theta_1 + offset_1) & 0 & -\cos(\theta_1 + offset_1) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (52)$$

$$A_1^2 = \begin{pmatrix} \cos(\theta_2 + offset_2) & 0 & \sin(\theta_2 + offset_2) & 0 \\ \sin(\theta_2 + offset_2) & 0 & -\cos(\theta_2 + offset_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (53)$$

$$A_2^3 = \begin{pmatrix} \cos(\theta_3 + offset_3) & -\sin(\theta_3 + offset_3) & 0 & L_1 \cos(\theta_3 + offset_3) \\ \sin(\theta_3 + offset_3) & \cos(\theta_3 + offset_3) & 0 & L_1 \sin(\theta_3 + offset_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (54)$$

$$A_3^4 = \begin{pmatrix} \cos(\theta_4 + offset_4) & -\sin(\theta_4 + offset_4) & 0 & L_2 \cos(\theta_4 + offset_4) \\ \sin(\theta_4 + offset_4) & \cos(\theta_4 + offset_4) & 0 & L_2 \sin(\theta_4 + offset_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (55)$$

$$A_4^5 = \begin{pmatrix} \cos(\theta_5 + offset_5) & 0 & \sin(\theta_5 + offset_5) & 0 \\ \sin(\theta_5 + offset_5) & 0 & -\cos(\theta_5 + offset_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (56)$$

$$A_5^6 = \begin{pmatrix} \cos(\theta_6 + offset_6) & -\sin(\theta_6 + offset_6) & 0 & L_3 \cos(\theta_6 + offset_6) \\ \sin(\theta_6 + offset_6) & \cos(\theta_6 + offset_6) & 0 & L_3 \sin(\theta_6 + offset_6) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (57)$$

Las matrices anteriormente mencionadas son resultado de la aplicación de los parámetros DH que parecen en la siguiente tabla:

Link/Parámetro	θ	d	a	α	RP
1	0	0	0	$\pi/2$	R
2	0	0	0	$\pi/2$	R
3	0	0	L_1	0	R
4	0	0	L_2	0	R
5	0	0	0	$\pi/2$	R
6	0	0	L_3	0	R

Tabla 2 - Denavit- Hartenberg (DH) parámetros para extremidad inferior derecha

Anexo II: Parámetros extremidad superior del paciente

Este es el código pacparlowerLimb.m de Matlab que carga los parámetros que conforman el objeto para la simulación de modelos de miembro superior:

```
%% Human parameters:

% Human masses, 1=upper arm 2=forearm
% (reference weight: male and female total body mass:100 kg)

mRef = 100;
% mC0 = 79; % COXX human mass (check BP-1HNP excel document)

m1h = 2.8*mC0/mRef; % upper arm
% m1h=1.864572; % OpenSim humerus data
m2h = 2*mC0/mRef; % forearm
% m2h =1.534315; % OpenSim radius data

clear mRef;clear mC0;

% "Cylinder" radius which simulate upper arm and forearm:
Rh1=0.05;
Rh2=0.03;

% "Cylinder" lengths:
L1= 0.28;
L2= 0.15;

% Inertial moments (L:longitudinal, R:Radial):
ILh1=m1h*(Rh1^2)/2;
IRh1=(m1h/4)*((Rh1^2)+(L1^2)/3);
ILh2=m2h*(Rh2^2)/2;
IRh2=(m2h/4)*((Rh2^2)+(L2^2)/3);

clear Rh1;clear Rh2;

%Denavit Hartenberg (DH) parameters (Links):

%
th d   a   alpha RP: joint type (R:revolute,P:prismatic)
LH(1) = Link([ 0  0  0  pi/2  0]); % shoulder inner rotation/outer rotation
LH(2) = Link([ 0  0  0  pi/2  0]); % shoulder abduction/aduction
LH(3) = Link([ 0  0 -L1  0      0]); % shoulder flexion/extension
LH(4) = Link([ 0  0  0  pi/2  0]); % elbow flexion/extension
LH(5) = Link([ 0  L2  0  0      0]); % elbow supination/pronation

%% Links Dynamic parameters:

% Masses
LH(1).m = 0; %several links at the same joint so mass is only settled on the
LH(2).m = 0; %last one
LH(3).m = m1h;
LH(4).m = 0;
LH(5).m = m2h;
```

```

% Distance to CoG (center of gravity):
LH(1).r = [ 0      0      0];
LH(2).r = [ 0      0      0];
LH(3).r = [ 0.9*L1 0      0]; % 0:elbow, 1:Shoulder (CoG location)
LH(4).r = [ 0      0      0];
LH(5).r = [ 0      0    -0.2*L2]; % 0:wrist, -1:elbow (CoG location)

% Note: CoG aligned with joints linker...

clear L1;clear L2;

% Inertia moments: [Ix Iy Iz Ixy Iyz Ixz]
LH(1).I = [0 0 0 0 0 0];
LH(2).I = [0 0 0 0 0 0];
LH(3).I = [ILh1 IRh1 IRh1 0 0 0];
LH(4).I = [0 0 0 0 0 0];
LH(5).I = [IRh2 IRh2 ILh2 0 0 0];

% Motor Inertia (motor referred)
LH(1).Jm = 100e-6;
LH(2).Jm = 100e-6;
LH(3).Jm = 100e-6;
LH(4).Jm = 15e-6;
LH(5).Jm = 15e-6;

% Gear ratio
LH(1).G = -1;
LH(2).G = 1;
LH(3).G = -1;
LH(4).G = 1;
LH(5).G = 1;

% Viscous friction (motor referred)
LH(1).B = 1.48e-10;
LH(2).B = .817e-10;
LH(3).B = 1.38e-10;
LH(4).B = 71.2e-10;
LH(5).B = 82.6e-10;

% Coulomb Friction
LH(1).Tc = [0 0];
LH(2).Tc = [0 0];
LH(3).Tc = [0 0];
LH(4).Tc = [0 0];
LH(5).Tc = [0 0];

% Offset kinematic: joint variable offset:
LH(1).offset=0;
LH(2).offset= pi/2;
LH(3).offset= 0;
LH(4).offset= -pi/2;
LH(5).offset= 0;

% Joint variable limits(max/min):
LH(1).qlim=[pi/2 pi];
LH(2).qlim=[0 3*pi/2];
LH(3).qlim=[-pi/2 pi/2];
LH(4).qlim=[0 3*pi/4];
LH(5).qlim=[-pi/2 pi/2];

```

Anexo III: Parámetros extremidad inferior del paciente

Este es el código pacparlowerLimb.m de Matlab que carga los parámetros que conforman el objeto para la simulación de modelos de miembro inferior:

```
%% Human lower limb parameters:

% Human masses, 1=thigh 2=leg 3=foot
% m1p=8; %from hip to knee
m1p=9.3014; %femur openSim data
% m2p=3.5; %knee to ankle
m2p= 3.7075; %tibia openSim data
% m3p=0.5; %ankle to toes
m3p=1.3; %talus,calcaneus and toes openSim data

% Cylinder radius which simulate thigh, leg and foot:
% Rh1=0.15;
% Rh2=0.10;
Rh3=0.025;

% Cylinder lengths:
L1=0.37;
L2=0.35;
L3=0.22;

% Inertial moments (L:longitudinal, R:Radial):
% ILh1=m1p*(Rh1^2)/2;
% IRh1=(m1p/4)*((Rh1^2)+(L1^2)/3);
% ILh2=m2p*(Rh2^2)/2;
% IRh2=(m2p/4)*((Rh2^2)+(L2^2)/3);
ILh3= m3p*(Rh3^2)/2;
IRh3=(m3p/4)*((Rh3^2)+(L3^2)/3);

clear Rh3;

% Denavit Hartenberg (DH) parameters (Links):

%
th d a alpha RP: joint type (R:revolute,P:prismatic)
LH(1) = Link([ 0 0 0 pi/2 0]); % Hip inner(+) - outer(-) rotation
LH(2) = Link([ 0 0 0 pi/2 0]); % Hip Adduction(+) - Abduction(-)
LH(3) = Link([ 0 0 L1 0 0]); % Hip Flexion(+) - Extension (-)
LH(4) = Link([ 0 0 L2 0 0]); % Knee Extension(+) - Flexion(+)
LH(5) = Link([ 0 0 0 pi/2 0]); % Ankle Dorsal Extension(+) - Plantar Flexion(-)
LH(6) = Link([ 0 0 L3 0 0]); % Ankle supination(+) - pronation(-)

%% Link Dynamic parameters:

% Masses
LH(1).m = 0;
LH(2).m = 0;
LH(3).m = m1p;
LH(4).m = m2p;
LH(5).m = 0;
LH(6).m = m3p;

clear m1p;clear m2p;clear m3p;
```

```

% Distance to CoG(center of gravity):
LH(1).r = [ 0      0      0];
LH(2).r = [ 0      0      0];
LH(3).r = [ -0.7*L1 0      0];
LH(4).r = [ -0.7*L2 0      0];
LH(5).r = [ 0      0      0];
LH(6).r = [ -0.5*L3 0      0];

clear L1;clear L2;clear L3;

% Inertia moments: [Ix Iy Iz Ixy Iyz Ixz]
LH(1).I = [0 0 0 0 0 0];
LH(2).I = [0 0 0 0 0 0];
LH(3).I = [0.0351 0.1412 0.1339 0 0 0]; % from OpenSim Femur
% LH(3).I = [ILh1 IRh1 IRh1 0 0 0];
LH(4).I = [0.005 0.0504 0.0511 0 0 0]; % from OpenSim Tibia
% LH(4).I = [ILh2 IRh2 IRh2 0 0 0];
LH(5).I = [0 0 0 0 0 0];
%As this last joint is a set of main bones, talus, calcaneus, toes...
LH(6).I = [ILh3 IRh3 IRh3 0 0 0]; % because longitudinal axis of the
% cylinder corresponds
% with x reference of the last joint...
clear ILh3;clear IRh3;

% Motor Inertia (motor referred):
LH(1).Jm = 0;
LH(2).Jm = 0;
LH(3).Jm = 0;
LH(4).Jm = 0;
LH(5).Jm = 0;
LH(6).Jm = 0;

% Gear ratio:
LH(1).G = -1;
LH(2).G = 1;
LH(3).G = -1;
LH(4).G = 1;
LH(5).G = 1;
LH(6).G = 1;

% Viscous friction (motor referred):
LH(1).B = 1;%1.48e-3;
LH(2).B = 1;% .817e-3;
LH(3).B = 1;% 1.38e-3;
LH(4).B = 1;%1.2e-6;
LH(5).B = 1;%82.6e-6;
LH(6).B = 1;%36.7e-6;

% Coulomb Friction:
LH(1).Tc = [0 0];
LH(2).Tc = [0 0];
LH(3).Tc = [0 0];
LH(4).Tc = [0 0];
LH(5).Tc = [0 0];
LH(6).Tc = [0 0];

% Offset kinematic: joint variable offset:
LH(1).offset= pi/2;
LH(2).offset= -pi/2;
LH(3).offset= 0;
LH(4).offset= 0;
LH(5).offset= pi/2;
LH(6).offset= 0;

```

Anexo IV: Análisis fuerza de reacción durante ejercicio de caminar

En este anexo se aporta información adicional acerca del efecto de las fuerzas de reacción durante ejercicios de caminar de manera que comprendan mejor la evolución de las mismas durante el tiempo, y se entienda, en caso de que se revisen los modelos en Matlab, el código implementado para describir este efecto.

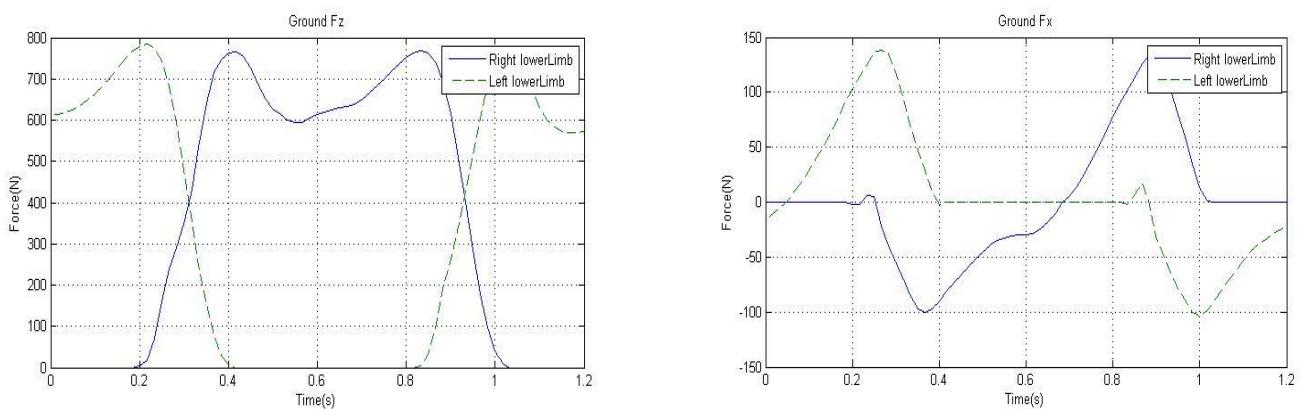


Figura 3 - Fuerzas externas sobre extremidades inferiores durante ejercicio completo de caminar

Se observa en la figura 42 cómo se reparte la acción del peso del sujeto en cada extremidad según evoluciona el ejercicio. Atendiendo a la extremidad derecha, que es para la que se muestran los resultados, la simulación comienza con la extremidad en fase de balanceo, por lo que la extremidad izquierda es la que soporta todo el peso sobre la superficie de contacto planta-superficie del ejercicio. La siguiente figura muestra la posición de las extremidades en el momento de comenzar la fase de apoyo de la extremidad derecha:

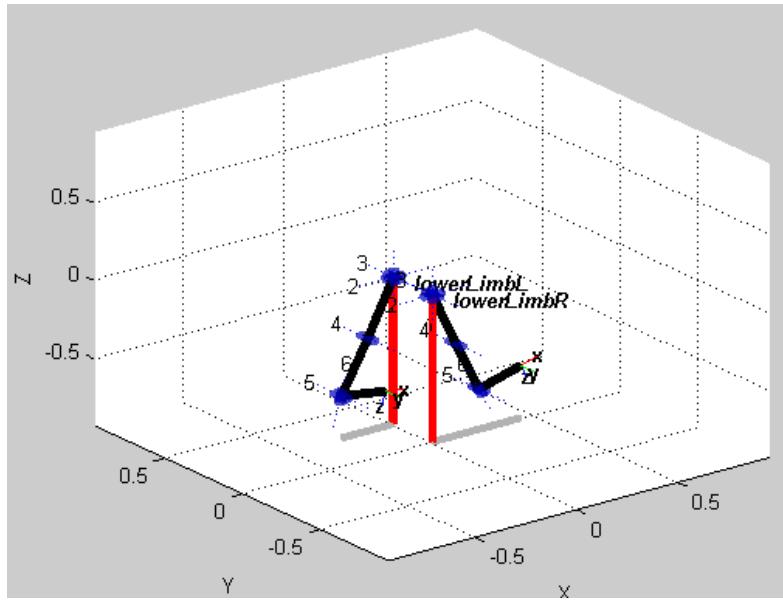


Figura 4 - Posición de extremidades inferiores durante ejercicio de caminar cuando la extremidad derecha está a punto de alcanzar la fase de apoyo

Cuando la fuerza sobre el eje z deja de ser nulo, el talón de la extremidad derecha contacta con la superficie, por lo que parte de la fuerza derivada del peso del sujeto se distribuye gradualmente hasta que alcanza un máximo, que coincide con la fase de balanceo de la extremidad izquierda.

Los picos que aparecen son resultado de que en los primeros instantes de la fase de apoyo, la superficie de contacto es más reducida por lo que la fuerza se distribuye menos, y el efecto es más puntual, generando esos máximos. El mismo efecto ocurre al despegar la punta del pie cuando finaliza la fase de apoyo de la extremidad, y comienza de nuevo la fase de balanceo.

De manera inversa, el valle que se genera durante la mitad de la fase de apoyo, implica una mayor superficie de contacto, y por lo tanto las fuerzas se distribuyen y su efecto se reparte. Además hay que considerar, que en ejercicios reales y con modelos no tan simplificados, aparecen componentes sobre el eje perpendicular al plano sagital, que no han sido introducidos en los cálculos del modelo al que atiende este apartado. De todos modos, el efecto de estas componentes es despreciable respecto de las demás.

En cuanto a los valores positivos y negativos que presenta esta componente, como ya se ha mencionado anteriormente, depende del sentido del movimiento. Durante el inicio de la fase de apoyo, como se representa en la figura 43, la fuerza de rozamiento se opone al movimiento, ya que la tendencia del talón sería que, una vez apoyado, deslizase hacia adelante en el sentido

del ejercicio de caminar. Por otro lado, aun en la fase de apoyo, pero cuando la extremidad se encuentra retrasada respecto al tronco, el efecto es positivo sobre el eje x e impulsa al sujeto. Este instante se representa en la siguiente figura:

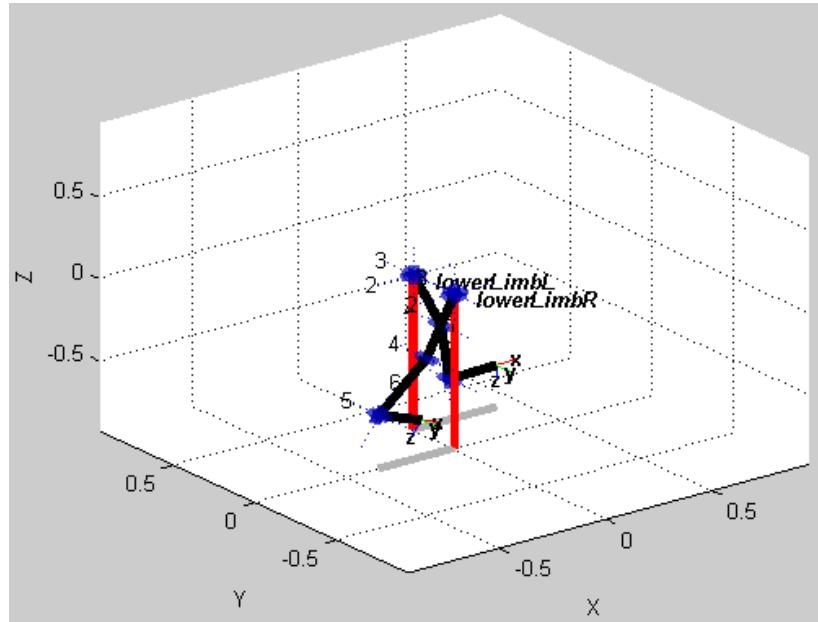


Figura 5 - Posición de extremidades inferiores durante ejercicio de caminar donde sobre la extremidad derecha se desarrolla una fuerza de rozamiento que coincide con el sentido de su trayectoria

Simulación evolución de aplicación de fuerzas de reacción

Para simular este efecto, se detecta en que instante se apoya el talón, que coincide con el que las fuerzas de reacción de la extremidad considerada dejan de ser nulas, y en cual se despega el pie para que comience la fase de balanceo. Entre esos dos instantes, se actualiza el parámetro *link(6).a* perteneciente al *link(6)* del objeto “*lowerLimb*”, que corresponde a la longitud del pie, durante la fase de apoyo de la simulación. Entonces, el Jacobiano se actualiza en cada instante de la trayectoria, de modo que la fuerza se aplique en cada instante al end-effector.

Anexo V: Resultados de la Dinámica directa del humano

Miembro superior

Mediante el control de aceleración propuesto, los resultados son los siguientes:

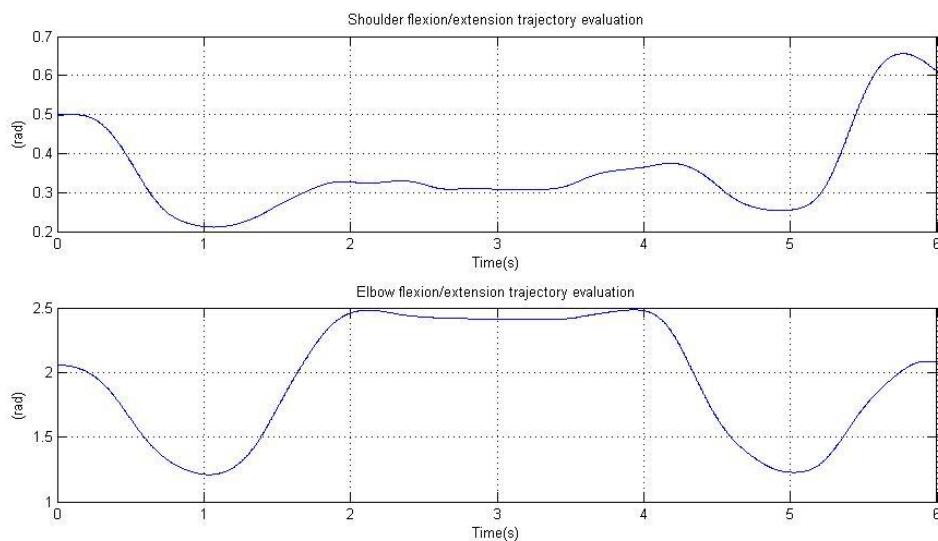


Figura 6 - Resultados de la dinámica directa para extremidad superior para un ejercicio de caminar con el sujeto 5

Sujeto/Error	RMSe(hombro) [rad]	RMSe(codo) [rad]	R^2(hombro)	R^2(codo)
1	0.0031	0.0125	0.9999	0.9999
2	0.0017	0.0048	1	1
3	0.0029	0.0135	0.9997	0.9999
4	0.0031	0.0018	0.9999	1
5	0.0021	0.006	1	1
6	0.0034	0.013	0.9997	0.9992

Tabla 3 - Resultados RMSe y R² de la dinámica directa de superior para ejercicios de beber con pacientes reales

Miembro inferior

El esquema de bloques es muy parecido al modelo diseñado para la evaluación de la dinámica directa de extremidad superior, salvo que en este se han de controlar 3 grados de libertad y hay que considerar las fuerzas externas ya comentadas previamente.

Los resultados del modelo calculados respecto de los datos del modelo en OpenSim son los siguientes para los movimientos de flexión/extensión de la cadera y de extensión/flexión de rodilla y tobillo:

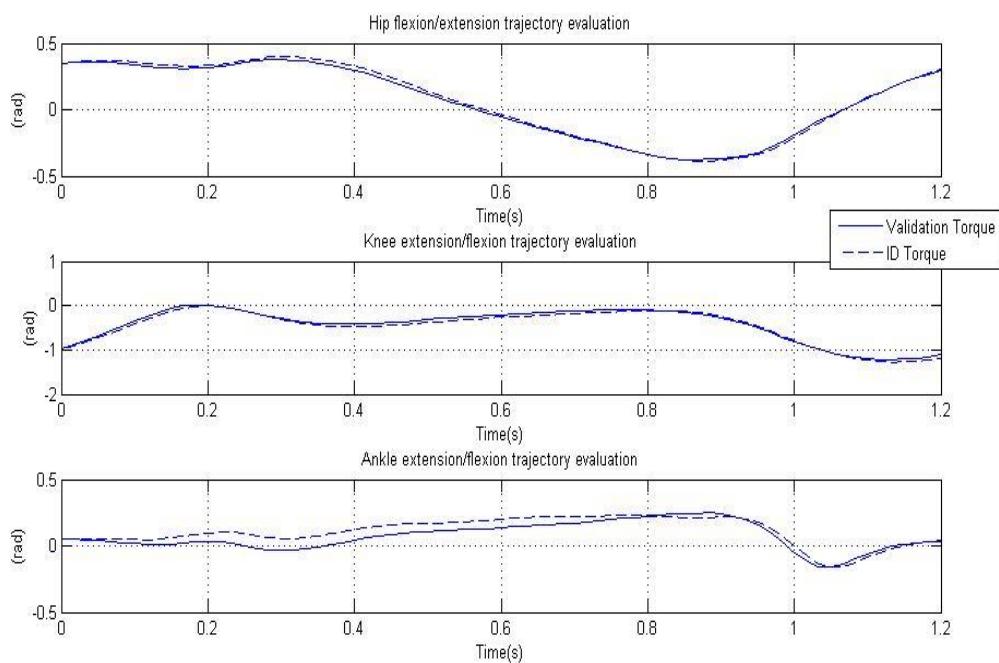


Figura 7 - Resultados de la dinámica directa para extremidad inferior derecha para un ejercicio de caminar con el dataset de OpenSim

RMSe(cadera) [rad]	RMSe(rodilla) [rad]	RMSe(tobillo) [rad]	R^2(cadera)	R^2(rodilla)	R^2(tobillo)
0.0164	0.047	0.0512	0.9993	0.9964	0.9304

Tabla 4 - Resultados RMSe y R^2 de la dinámica directa para extremidad inferior derecha para un ejercicio de caminar con el dataset de OpenSim (incluye fuerzas de contacto)

Se observa que la tendencia de la dinámica que genera el modelo coincide con la de referencia. En cuanto a la magnitud del error y la correlación, el desajuste se debe a la falta de precisión en algunos parámetros dinámicos que definen las extremidades, difíciles de precisar o calcular para cada persona.

Anexo VI: Tutorial ficheros Matlab

Como normal general a la hora de trabajar con los ficheros de simulación para todos los modelos implementados, se recomienda ejecutar los mismos dentro de la carpeta donde está alojado ese fichero de inicialización del modelo en particular, ya que algunos de los ficheros que contienen los parámetros dinámicos o funciones adicionales comparten el mismo nombre pero contienen modificaciones de manera que se optimice el funcionamiento de cada modelo y el “workspace” de Matlab esté lo más ordenado posible y muestre únicamente los resultados o parámetros de interés. Además, todos los datos de entrada están alojados dentro de las carpetas donde se encuentran los ficheros de inicialización.

Para la validación de los modelos de extremidad superior e inferior del humano durante el cálculo de la dinámica inversa, se utilizan los ficheros de inicialización “upperLimbIDevaluation.m” y “lowerLimbIDevaluation.m” respectivamente, que cargan los datos dinámicos del sujeto, la cinemática del ejercicio y la cinética real asociada a ese cinemática que se utiliza como referencia, y simulan los modelos en Simulink “upperLimbID.mdl” o “lowerLimbID.mdl” respectivamente, que calcula la cinética articular del modelo. Las características dinámicas de la extremidad del sujeto están alojadas dentro de los ficheros “pacparUpperLimb.m” y “pacparLowerLimb2”, y como se ha comentado previamente, permite el ajuste de los parámetros según el sujeto. Este último fichero “pacparLowerLimb2” almacena información de las 2 extremidades inferiores de la persona. También se ha propuesto la validación del modelo de extremidad superior únicamente durante la fase de balanceo del ejercicio de caminar, por lo que se han modificado los modelos anteriores de extremidades inferiores de manera que se desprecia el pie del sujeto, los ficheros son “lowerLimbIDevaluationSwing.m”, “lowerLimbIDSwing.mdl” y “pacparLowerLimbHipKnee”, que funcionan de manera análoga a los comentados previamente.

Para el análisis durante el cálculo de la dinámica inversa de la contribución particular de cada fenómeno (gravedad, inercia, fricción...), se ha implementado un fichero “IDanalysis.m” que representa los resultados en forma de gráficas.

Para la validación de los modelos de extremidad superior e inferior del humano durante el cálculo de la dinámica directa, se procede de igual manera que en el caso anterior, con los ficheros “upperLimbFDevaluation.m”, “upperLimbFD.mdl” para extremidad superior, y con “lowerLimbFDevaluation.m” y “lowerLimbFD.mdl” para extremidades inferiores. Los ficheros para cargar las características dinámicas del sujeto son los mismos que en para el cálculo de la

dinámica inversa. La única diferencia es que el fichero de inicialización también contiene los parámetros de los *PID*'s necesarios para este cálculo como se ha discutido previamente en otros capítulos de este documento.

Para inicializar la simulación de la terapia de rehabilitación asistida por robot para extremidad superior, se accede a la carpeta (“Current Folder”) en Matlab que contiene el fichero “RobotAssistedTherapy.m”, el cual carga los parámetros de los elementos que participan del ejercicio: paciente “pacparUpperLimb.m”, robot “robparUpperLimb.m”, y de la combinación de los mismos actuando conjuntamente “sysparUpperLimb.m”, y simula el modelo “RobotAssistSystem.mdl”.

Anexo VII: Parámetros del modelo Hill para sEMG

En este apartado se presenta información adicional para completar cómo se utilizan las señales EMG dentro de las terapias asistidas de rehabilitación.

Glosario

α	Porcentaje de fibras rápidas del elemento contráctil
ANC	Anconeus
BIC long	Biceps long head
BICshort	Biceps short head
BRA	Brachialis
BRD	Brachioradialis
ECRL	Extensor carpi radialis longus
E_{rms}	error cuadrático medio
$f_{_ANC}$	factor de escala de la activación de músculo ANC
$f_{_BIC}$	factor de escala de la activación de músculo BIC long
$f_{_BRA}$	factor de escala de la activación de músculo BRA
F_{CE}	fuerza del elemento contráctil
F_{CEmax}	fuerza máxima del elemento contráctil
FCR	Flexor Carpi Radialis
$f_{_geom}$	factor geométrico del modelo
F_i	Fuerza muscular total del músculo i
f_l	relación entre fuerza muscular y longitud de músculo
F_{PE}	fuerza del elemento paralelo
F_{PEmax}	fuerza máxima del elemento paralelo
F_{SE}	fuerza del elemento en serie
F_{SEmax}	fuerza máxima del elemento en serie
f_v	relación entre fuerza muscular y velocidad de contracción
L_{CEO}	longitud de fibra óptima
ΔL_{CE}	variación de longitud del elemento contráctil con respecto al reposo

ΔL_{CEmax}	variación máxima de la longitud del elemento contráctil
L_m	longitud de músculo
\widetilde{L}_m	longitud de músculo-fibra normalizada
ΔL_{PE}	variación de longitud del elemento paralelo con respecto al reposo
ΔL_{PEmax}	variación máxima de la longitud del elemento paralelo
ΔL_{SE}	variación de longitud del elemento en serie con respecto al reposo
ΔL_{SEmax}	variación máxima de la longitud del elemento en serie
L_{TS}	longitud del tendón en reposo
M_{neto}	momento neto en la articulación
MVC	maxima contracción voluntaria
PT	Pronator teres
S_{PE}	factor de forma del elemento paralelo
S_{SE}	factor de forma del elemento contráctil
TRIIlat	Triceps lateral head
TRIIlong	Triceps long head
TRImed	Triceps medial head
U	activación muscular normalizada
V_{CEO}	máxima velocidad elemento contráctil con nivel de activación máximo
V_{CE}	velocidad de contracción del elemento contráctil
V_{CEmax}	velocidad máxima del elemento contráctil con

Calibración del modelo

Este modelo tiene un enfoque muy biomecánico y hay parámetros que es muy difícil de hallar para cada persona, haciendo necesaria una calibración individual para cada sujeto. Durante el proceso de calibración, se utiliza la dinámica inversa obtenida mediante OpenSim que nos proporciona el valor teórico de los pares articulares a partir de la cinemática del movimiento, como referencia de par con la que comparar. En cada iteración de la calibración la estimación obtenida mediante el modelo de Hill a partir del *EMG*, la cinemática y el nivel de fatiga detectado se compara el valor con la dinámica inversa y el error entre ambas se utiliza para realimentar y reajustar el sistema. El ajuste se realiza mediante un procedimiento de optimización que proporciona los parámetros óptimos del modelo.

Se utiliza un algoritmo no lineal ‘trust-region-reflective’ [15] disponible en el *Matlab Optimization toolbox* para resolver este problema de ajuste de curvas mediante mínimos cuadrados ('lsqcurvefit "función de Matlab").

Con un modelo completo experimental de la extremidad superior [11,12], se obtuvieron los parámetros genéricos, para el percentil 50 de un varón de 170 cm de altura, de L_{\max} , L_{C0} , L_{TS} , y F_{cemax} ,

Muscle	1	2	3	4	5	6	7
	L_{\max} [cm]	L_{C0} [cm]	L_{TS} [cm]	F_{cemax} [N]	α [%]	S_{PE}	S_{SE}
1 DELT3	19.9392	13.67	3.8	259.88	58	9	2.8
2 DELT2	21.9046	10.78	10.95	1142.6	58	9	2.8
3 DELT1	38.99	9.76	9.3	1142.6	58	9	2.8
4 PECM1	16.1934	14.42	0.28	364.41	58	9	2.8
5 BIClong	45.53	11.57	27.23	624.3	56	9	2.8
6 BICshort	41.6755	13.21	19.23	435.56	56	9	2.8
7 TRIlat	21.47	11.38	9.8	624.3	66	10	2.3
8 TRIlong	32.57	13.4	14.3	798.52	66	10	2.3
9 TRImed	20.233	11.38	9.08	624.3	66	10	2.3
10 ECRL	34.49	8.1	22.4	304.89	50	8	3
11 FCR	32.84	6.28	24.4	73.96	58	6	3
12 BRD	30.6	17.3	13.3	261.3	56	9	2.8
13 FCU	31.6	5.1	26.5	128.9	58	6	3
14 INFRA	10.7	7.6	3.1	1210.8	58	9	2.8

Tabla 5 - Parámetros genéricos obtenidos (Marcados en verde los parámetros a optimizar)

La Tabla 2 representa los parámetros del modelo que se optimizan y los límites utilizados. Los parámetros α , S_{PE} , y S_{SE} fueron tomados de [14], al igual que algunos de los valores sobre los límites del intervalo inferior y superior de los parámetros. En total nuestro modelo contiene 72

parámetros que optimizar, 5 para cada músculo y 2 factores globales (el factor de activación del músculo BICshort y un factor de escala geométrico).

Parámetro	Límites	Característica
Lce_o	[0.8,1.2]	por músculo
Fce_{max}	[0.5,1.5]	por músculo
α	[0.5,1.5]	por músculo
S_{PE}	[0.8,1.2]	por músculo
S_{SE}	[0.8,1.2]	por músculo
f_BIC	[0.8,1.2]	Global
f_geom	[0.8,1.2]	Global

Tabla 6 - Límites superiores e inferiores impuestos en el proceso de optimización

Una vez calibrado, el modelo pasa a funcionar en bucle abierto, dejando de utilizar la dinámica inversa de OpenSim para realimentar el sistema, y utilizando únicamente el modelo de Hill.

Anexo VII: Esquema Simulink del modelo de terapia para rehabilitación asistida por robot

El esquema representado en la siguiente página corresponde al modelo de Simulink con el que se ha trabajado para la simulación del funcionamiento de la terapia previamente explicada en el capítulo 5 de este documento representado en la figura 19.

Como se puede apreciar, en el modelo de Simulink, no se realimenta la cinemática del ejercicio al bloque donde el modelo de Hill debería estar integrado, ni tampoco se trabaja con la señal EMG directamente, sino que toda esa información ha sido procesada previamente y no durante la simulación del ejercicio de terapia.

Los bloques en amarillo representan los inputs que se introducen al sistema, tanto la cinética que se generaría mediante el modelo de Hill, como la cinemática de referencia del ejercicio. También se introduce desde el workspace de Matlab el valor de la K correspondiente a la etapa proporcional.

Los bloques en verde son las variables que almacenan los resultados cinéticos de los elementos que participan de la terapia (del humano; del exoesqueleto para desplazar su propia dinámica; del exoesqueleto para asistir al sujeto, tanto antes como después de las etapas de saturación y proporcional; y también la cinemática que realmente desarrolla el conjunto del sistema).

Los bloques en rojo representan el comportamiento dinámico del robot y del sistema conjunto humano-robot, mediante el cálculo de la dinámica inversa y de la dinámica directa respectivamente.

Por último, los bloques en cian recogen, primero la etapa de control de posición y velocidad, y también, las etapas de saturación y proporcional.

