

Proyecto Fin de Máster Máster en Ingeniería Industrial

Analisis Dinámico Inverso de la Marcha Humana

Autor: Eduardo Maza Ortega

Tutor: Joaquín Ojeda Granja

**Dep. Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2019



Proyecto Fin de Máster
Máster en Ingeniería Industrial

Analisis Dinámico Inverso de la Marcha Humana

Autor:

Eduardo Maza Ortega

Tutor:

Joaquín Ojeda Granja

Profesor Contratado Doctor Interino

Dep. Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Proyecto Fin de Máster: Análisis Dinámico Inverso de la Marcha Humana

Autor: Eduardo Maza Ortega

Tutor: Joaquín Ojeda Granja

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Este trabajo no ha sido fácil y no hubiera sido posible completarlo sin el apoyo de mi familia y mis amigos. A todos ellos, gracias.

Resumen

En el presente trabajo se desarrolla un programa en Matlab para resolver un problema dinámico inverso aplicado a la marcha humana. El objetivo de este problema es obtener el valor de los momentos en cadera, rodilla y tobillo durante la marcha. Para ello se toma como referencia el software de OpenSim pero intentando llevar a cabo un código mas sencillo y que permita al usuario libertad para modificarlo sin necesidad de amplios conocimientos de programación. Destacan el uso de calculo simbólico en Matlab y de un conjunto mínimo de coordenadas de manera que no existan restricciones algebraicas sobre ellas [18].

En los dos primeros capítulos se desarrolla el trasfondo teórico necesario para comprender el problema. En el tercer capítulo se explica el modelo dinámico utilizado. Los capítulos cuarto y quinto suponen el núcleo del trabajo ya que desarrollan el problema cinemático y dinámico inverso presentando los resultados obtenidos en Matlab y su comparación con OpenSim. Por último, en el sexto capítulo se obtienen conclusiones y se comentan aspectos a mejorar.

Abstract

The present work develops a Matlab program to solve an inverse dynamics problem applied to human gait. The goal of this problem is to get the torques in hip, knee and ankle during gait. For this goal the OpenSim software is taken as reference but trying to develop an easier code that lets more freedom for the user without a broad programming knowledge to change it. Use of symbolic math in Matlab and a minimal set of coordinates so there are no algebraic restrictions on them is notable [18].

The first two chapters develop the theoretical background needed to comprehend the problem. In the third chapter the dynamic model used is explained. Fourth and fifth chapters are the work's core as they develop the kinematic and dynamic problems introducing the results in Matlab and its comparison with OpenSim. At last, the sixth chapter comes to conclusions and aspects to improve are commented.

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	3
2 Conceptos Teóricos	5
2.1 Ejes y planos anatómicos	5
2.2 Ecuaciones cinemáticas	7
2.2.1 Movimiento de un sólido	7
Ángulos de Tait-Bryan	7
Parámetros de Euler	7
Ecuación cinemática fundamental	8
2.2.2 Movimiento de un conjunto de sólidos	8
2.3 Ecuaciones dinámicas	9
2.4 El ciclo de marcha	11
3 Modelo Utilizado	15
3.1 Modelo Original	16
3.1.1 Cuerpos	16
3.1.2 Articulaciones	16
3.1.3 Coordenadas	17
3.1.4 Datos de entrada del modelo	20
Marcadores	20
Parámetros cinemáticos	22
Parámetros dinámicos	22
3.2 Modelos simplificados	22
4 Cinemática Inversa	23
4.1 Formulación	23
4.2 Modelado en Matlab	26
4.3 Presentación de resultados	27
4.3.1 Modelo original Delp	27
4.3.2 Modelo original con bloqueo	29
4.3.3 Modelo sin splines	30

4.3.4	Modelo monosegmento sin splines	32
5	Dinámica Inversa	35
5.1	Formulación	35
5.2	Modelado en Matlab	35
5.3	Presentación de resultados	37
6	Conclusiones	41
	<i>Bibliografía</i>	43

1 Introducción

1.1 Motivación

En el presente trabajo se ha llevado a cabo uno de los procedimientos más habituales en el análisis de la marcha humana como es la resolución de un problema dinámico inverso. Este estudio consiste en obtener las fuerzas y momentos en las articulaciones conociendo el movimiento del sujeto. La principal motivación para calcular estos valores reside en que son difíciles de medir directamente de forma experimental y son parámetros de diseño importantes, por ejemplo, en el diseño de prótesis [14]. Por otra parte, previamente a la obtención de los momentos se obtienen los ángulos en las articulaciones. Si se comparan los valores obtenidos en la marcha de un paciente con los valores de referencia para una persona sana se puede establecer la severidad de la enfermedad. Esta clasificación se lleva a cabo en enfermos de parálisis cerebral [8] y resulta una mejora de los métodos tradicionales de clasificación que se basaban en la mera observación de la marcha del paciente.

De forma específica, los estudios de análisis de la marcha consideran que el aparato locomotor se puede modelar como un conjunto de sólidos rígidos unidos entre sí mediante articulaciones. El estudio de Seireg y Arvikar [17] presenta un modelo matemático del sistema musculoesquelético capaz de evaluar fuerzas musculares y momentos en las articulaciones para diferentes posturas estáticas. En él se modelan los principales músculos de las extremidades inferiores y se aplican ecuaciones de equilibrio para cada uno de los segmentos considerados (pelvis, fémur, tibia y fíbula y pie). El problema tiene más incógnitas que ecuaciones, por lo que se utilizan criterios de minimización para obtener las fuerzas musculares. Al tratarse de un sistema de ecuaciones lineal se resuelve mediante un algoritmo simplex. Los resultados obtenidos para diferentes grados de inclinación del tronco y de flexión de cadera y rodilla muestran una correlación entre las señales EMG (electromiográficas) medidas experimentalmente y las fuerzas obtenidas analíticamente.

Por su parte, el estudio de Winter [20] estudia el valor de un momento de soporte neto calculado como la suma de los momentos en tobillo, rodilla y cadera. La suposición inicial de que dicho momento debe ser positivo durante toda la fase de apoyo de la marcha se hace evidente en los resultados obtenidos. Para obtenerlos se utilizan marcadores reflectivos colocados en determinados puntos anatómicos del sujeto que camina, mientras se utiliza una cámara de cine (de frecuencia 50 Hz) para obtener su posición. Simultáneamente se registran las fuerzas de reacción con el suelo verticales y horizontales en la dirección de la marcha. Los momentos se calculan usando el programa cinético para segmentos unidos por pares estándar, desarrollado por Bresler y Frankel [3]. De esta manera se obtiene una curva para el momento de soporte neto con una forma representativa para toda la variedad de sujetos que se examinan, incluido un paciente sometido a cirugía en la rodilla.

En el artículo de Pandey y Berme [13] se presenta una técnica general para simular el movimiento del miembro inferior durante la marcha. Dicha técnica se basa en una formulación recurrente de

Newton-Euler para resolver el problema dinámico inverso. La principal innovación de este trabajo radica en que presenta una técnica general que permite no tener que derivar manualmente las ecuaciones del movimiento como se había hecho hasta ahora, por ejemplo, en el trabajo de Winter [20]. En este sentido también hay que destacar el artículo de Riemer *et al.* [15], en el que se modela el cuerpo humano como un conjunto de trece segmentos articulados entre sí para calcular los momentos en el plano sagital de las diferentes articulaciones. Para ello se resuelven las ecuaciones de equilibrio de cada segmento de forma sucesiva, empezando por el pie, por lo que es necesario conocer las fuerzas de reacción con el suelo. Una de las principales limitaciones de este modelo es que es bidimensional y por tanto solo permite calcular los momentos de flexión-extensión.

En el trabajo de Damsgaard *et al.* [4] se hace una revisión del software de simulación AnyBody Modeling System, el cual se utiliza para resolver el problema dinámico inverso de modelos musculoesqueléticos del cuerpo humano. Para ello se hace uso de una formulación general basada en la dinámica de sistemas multicuerpo en la que se usan coordenadas cartesianas y parámetros de Euler como coordenadas generalizadas. Dichas coordenadas están sujetas a ecuaciones de restricción que es necesario resolver antes de plantear las ecuaciones de equilibrio.

El artículo de Delp *et al.* [6] presenta un modelo gráfico del miembro inferior y se puede considerar el antecesor de OpenSim. En el se modela el miembro inferior como un sistema multicuerpo formado por siete segmentos. Es más tarde, en 2007, cuando Delp *et al.* [5] presenta propiamente OpenSim en otro artículo. OpenSim es una plataforma de software para modelar humanos, animales, robots y el ambiente, y simular su interacción y movimiento. OpenSim dispone de una interfaz gráfica de usuario (GUI por sus siglas en inglés) para visualizar modelos y generar y analizar simulaciones. Una captura del aspecto de esta GUI se puede observar en la Figura 1.1. De esta manera, OpenSim constituye una herramienta de fácil manejo para realizar análisis dinámicos inversos de modelos del cuerpo humano. Simbody es el código detrás de OpenSim y surge de la necesidad de la investigación biomédica de tener un software de dinámica multicuerpo orientado específicamente a la simulación de sistemas biomecánicos como el cuerpo humano. Se trata de un motor dinámico pensado para programadores o desarrolladores de aplicaciones que no necesariamente tienen que ser expertos en dinámica [19]. El código es de acceso libre y se puede encontrar en <https://simtk.org/home/simbody>. Con este último trabajo se llega a la aparición de este software, que supone el punto de partida del presente trabajo.

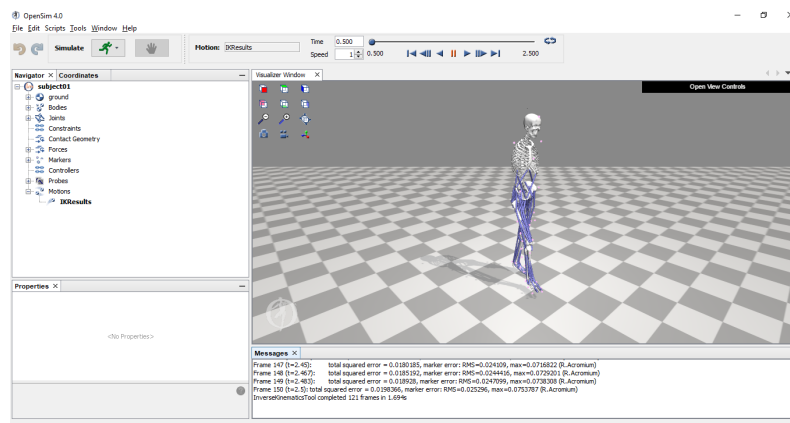


Figura 1.1 Aspecto de la GUI de OpenSim.

Frente a estos trabajos, el presente utiliza un modelo tridimensional del aparato locomotor humano basado en el uso de un conjunto mínimo de coordenadas relativas. Esto permite que estas coordenadas no estén sujetas a ecuaciones de restricción, lo que provoca un ahorro computacional ya que no hay que resolverlas. Por otra parte, al plantear el problema de forma general, es decir,

aplicando las ecuaciones de equilibrio sobre el conjunto de sólidos que conforman el sistema no es necesario conocer el valor de las fuerzas de reacción con el suelo.

Los dos códigos de Matlab programados resuelven el problema dinámico inverso emulando a OpenSim pero mediante un código mas sencillo y que permite una mayor personalización por parte del usuario. Aunque el código de OpenSim está publicado y se puede acceder a el para modificarlo, está escrito usando lenguaje C++, que esta muy optimizado para mejorar el rendimiento pero puede resultar difícil para un ingeniero que no este especializado en el. Por ello el presente trabajo se plantea el objetivo de emular a OpenSim pero escribiendo el código en Matlab, un lenguaje de programación mas orientado a la ingeniería y que se imparte y utiliza ampliamente en la ESI. Uno de los retos que se plantean es hacer comparable la velocidad de ejecución del código en Matlab con el código en C++ que utiliza OpenSim y que a priori es mas rápido Esto se debe a que C++ es un lenguaje de programación compilado frente a M (el lenguaje de programación de Matlab), que es un lenguaje interpretado. La diferencia entre ambos es que en el lenguaje compilado el código se traduce a lenguaje maquina en un paso previo, mientras que en el interpretado este paso se va haciendo a medida que se ejecuta el código. La ventaja del lenguaje compilado es que cuando se ejecuta el código ya esta en lenguaje maquina, por lo que la ejecución es mucho mas rápida. La desventaja del uso de lenguaje compilado es que hay que compilar el código cada vez que este cambia.

1.2 Objetivos

El objetivo principal de este trabajo es resolver un problema dinámico inverso de la marcha humana. Para ello se generan dos códigos de Matlab que resuelven la cinemática y la dinámica inversa. En el primero de ellos se obtiene el valor de los ángulos en las articulaciones; los resultados obtenidos se usan en el segundo para obtener el valor de los momentos en ellas. El esquema general del trabajo se muestra en la Figura 1.2. Desde el punto de vista computacional, la resolución del problema cinemático inverso usa calculo numérico para resolver un problema de optimización. Por su parte, el problema dinámico inverso consiste en última instancia en la resolución de un sistema de ecuaciones lineal (mas sencillo que una optimización) pero con la dificultad añadida del uso de calculo simbólico para la obtención de las matrices dinámicas. El modelo biomecánico utilizado es el conocido como gait2392 y se presentará en el tercer capítulo. Este modelo se encuentra dentro del paquete de descargas de OpenSim. Como datos de partida se tienen todos los parámetros del modelo y un fichero también presente en el paquete de descargas de OpenSim con los resultados de un experimento de análisis de la marcha humana.

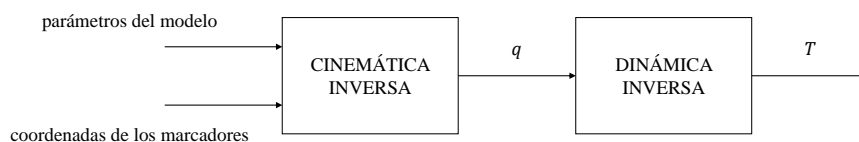


Figura 1.2 Esquema global.

2 Conceptos Teóricos

En este capítulo se explican algunos conceptos importantes de los que se hará uso a lo largo de este trabajo y que conviene tener claros para una buena comprensión del mismo.

2.1 Ejes y planos anatómicos

Antes de nada, al hacer referencia continua al movimiento del cuerpo humano y puntos anatómicos de este, es conveniente introducir los ejes y planos anatómicos que establecen una base para poder hablar de la disposición de los miembros del cuerpo. En la Figura 2.1 se observan los ejes y planos anatómicos que definen la geometría del cuerpo, así como los ejes de coordenadas globales (fijos en el espacio) que se van a utilizar a lo largo de todo el trabajo:

- Eje X: en la dirección de la marcha.
- Eje Y: eje vertical hacia arriba.
- Eje Z: producto vectorial de los anteriores.

Los tres planos anatómicos principales son:

- Plano coronal o frontal: divide al cuerpo en las mitades posterior y anterior.
- Plano sagital: divide al cuerpo en las mitades derecha e izquierda. En este plano se dan los movimientos principales relacionados con la marcha.
- Plano transversal o axial: divide al cuerpo en las mitades craneal y caudal.

Estos planos se cortan entre sí dando lugar a los ejes principales. Las direcciones de estos ejes servirán para describir las rotaciones que se producen en las diferentes articulaciones del cuerpo.

- Eje anteroposterior: es el corte de los planos sagital y transversal.
- Eje craneocaudal: es el corte de los planos sagital y coronal.
- Eje latero-lateral: es el corte de los planos coronal y transversal.

En la Figura 2.2 se presentan algunos términos que definen las posiciones de los puntos anatómicos respecto a los planos anatómicos. Así, a lo largo del eje craneocaudal un punto es proximal frente a uno distal por estar más cerca del centro del cuerpo, es decir, del plano transversal tal y como está representado en la figura Figura 2.1. Así, la tibia es un hueso más distal que el fémur, por ejemplo. Si nos referimos ahora al eje latero-lateral, un punto es medial frente a uno lateral si está más cerca del plano sagital. Así, el dedo gordo del pie está en una posición medial frente al pequeño, que está en una posición lateral. Por último, para la pareja anterior-posterior nos tenemos que fijar en si el

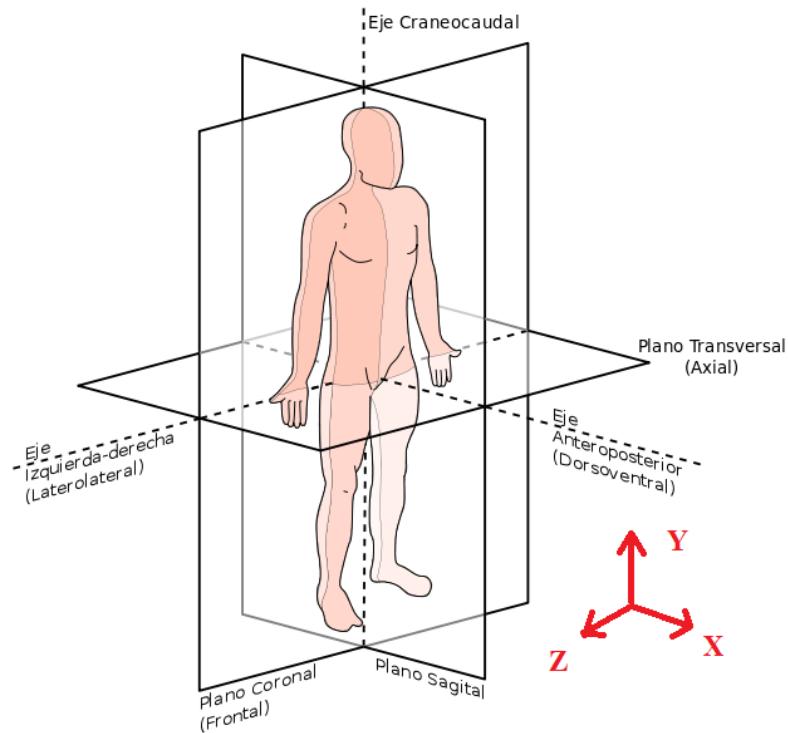


Figura 2.1 Ejes anatómicos.

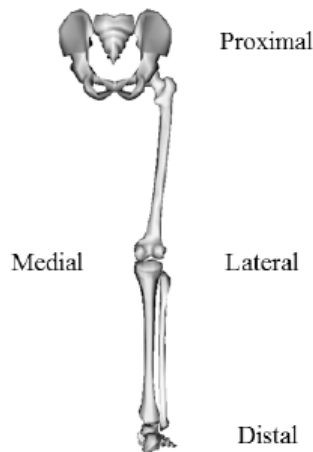


Figura 2.2 Términos anatómicos.

punto anatómico esta delante o detrás del plano coronal. Si está delante, como el pecho, la posición es anterior. Por el contrario, si se encuentra detrás, como la espalda, la posición es posterior. Otra pareja de términos que se va a utilizar a lo largo del trabajo es la de dorsal/ plantar. Dorsal hace referencia al empuñe del pie frente a plantar que hace referencia a la planta del pie.

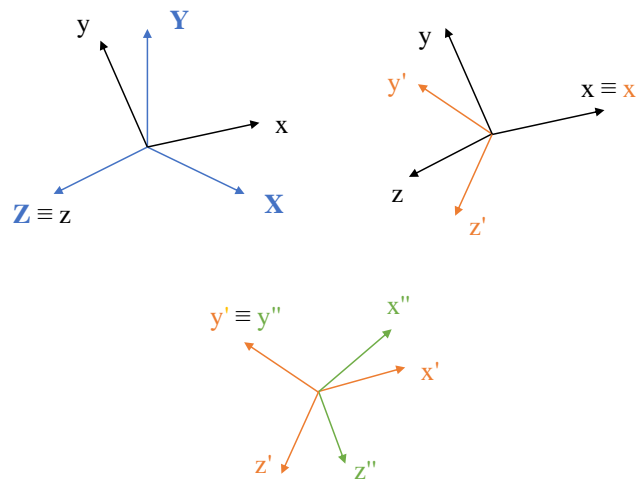


Figura 2.3 Secuencia de rotaciones.

2.2 Ecuaciones cinemáticas

2.2.1 Movimiento de un sólido

En cinemática se define un sistema de coordenadas global, fijo en el espacio, que sirve como referencia. A cada sólido se le acopla un sistema de coordenadas que se denomina local. La posición y orientación de dicho sistema local frente al global es lo que permite describir el movimiento del sólido. Todo movimiento resulta de una combinación de un movimiento de traslación con uno de rotación. El movimiento de traslación hace referencia a como cambia la posición del origen de coordenadas local respecto al global. El de rotación, por su parte, hace referencia a como varía la orientación de los ejes locales respecto a los globales.

Si se piensa en dos dimensiones, describir el movimiento resulta sencillo: la traslación tiene lugar en un plano y la rotación consiste en un giro alrededor de un eje perpendicular al plano. Si bien el movimiento de traslación en tres dimensiones es homólogo al caso tridimensional, la definición de la orientación es algo más complicado. Aunque existen diferentes métodos para esta definición, a lo largo de este trabajo se han utilizado dos de ellos: ángulos de Tait-Bryan y parámetros de Euler.

Ángulos de Tait-Bryan

Los ángulos de Tait-Bryan describen tres giros sucesivos que permiten pasar de la orientación del sistema global a la del sistema local.

El primer giro se realiza a lo largo del eje Z global, como se aprecia en la Figura 2.3

Realizando este giro se obtiene un sistema de coordenadas intermedio $x' - y' - z'$. El siguiente giro se produce alrededor del eje x' , como se aprecia en la Figura 2.3

El último giro se produce alrededor del eje y'' , como se aprecia en la Figura 2.3 Este tipo de rotaciones se denominan intrínsecas ya que se producen de forma sucesiva alrededor de los ejes que ya se han rotado previamente. En el modelo utilizado todas las rotaciones descritas con ángulos de Tait-Bryan se realizan de la manera descrita, es decir, alrededor de los ejes $z - x' - y''$. En ningún caso las rotaciones se producen alrededor de los ejes sin girar del sistema de coordenadas original.

Parámetros de Euler

Otro de los métodos utilizados para hallar la orientación de un sólido en este trabajo está basado en los parámetros de Euler. En este caso el paso de un sistema de coordenadas a otro se lleva a cabo mediante la rotación alrededor de un único eje. Para describir esta rotación es necesario conocer la

dirección de dicho eje: v_1, v_2, v_3 son las coordenadas de un vector unitario en la dirección de dicho eje; por otra parte, el ángulo de rotación se denota como α . Con estos valores se construyen los cuatro parámetros de Euler:

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} \cos(\frac{\alpha}{2}) \\ v_1 \sin(\frac{\alpha}{2}) \\ v_2 \sin(\frac{\alpha}{2}) \\ v_3 \sin(\frac{\alpha}{2}) \end{bmatrix} \quad (2.1)$$

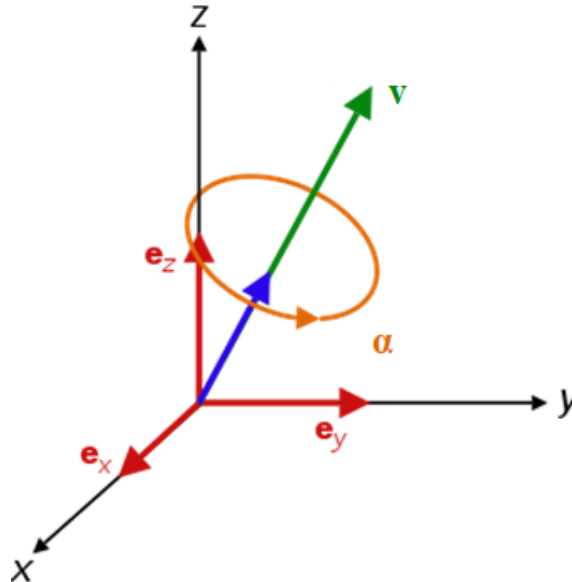


Figura 2.4 Parámetros de Euler.

Ecuación cinemática fundamental

A partir de los ángulos de Tait-Bryan o de los parámetros de Euler se construye la matriz de orientación del sólido. Para obtener el vector con las coordenadas de un punto cualquiera del sólido expresadas en el sistema global hace falta conocer esta matriz de orientación y la posición del origen de coordenadas locales expresado en el sistema de coordenadas global:

$$\mathbf{r} = \mathbf{R} + \mathbf{A}\mathbf{u} \quad (2.2)$$

Siendo \mathbf{u} el vector de posición del punto expresado en el sistema de coordenadas locales, \mathbf{A} la matriz de orientación del sólido, \mathbf{R} la posición del origen de coordenadas locales expresado en coordenadas globales y \mathbf{r} el vector de posición de dicho punto expresado en el sistema de referencia inercial (coordenadas globales).

2.2.2 Movimiento de un conjunto de sólidos

Hasta ahora se ha estado centrado en conocer la posición y orientación de un sólido en particular, sin embargo, el aparato locomotor humano se trata como un conjunto de sólidos articulados entre sí. Para poder obtener la posición de cada uno de estos sólidos se utilizará un conjunto de coordenadas relativas. Este tipo de coordenadas describen la posición y orientación de cada cuerpo respecto al cuerpo que le precede en la cadena cinemática. La ventaja del uso de estas coordenadas radica en que no se imponen restricciones sobre ellas, lo que permite no tener que resolver ningún tipo de ecuaciones de restricción.

En el modelo utilizado en este trabajo el origen de la cadena cinemática es la pelvis. La cinemática de este cuerpo se define a partir de la posición del origen del sistema de coordenadas asociado a él y de los tres ángulos de Tait-Bryan que definen su matriz de rotación asociada. A partir de estos datos se puede obtener la posición de cualquier punto que se encuentre en la pelvis:

$$\mathbf{r}_{\text{pelvis}} = \mathbf{R}_{\text{pelvis}} + \mathbf{A}_{\text{pelvis}} \mathbf{u}_{\text{pelvis}} \quad (2.3)$$

Como se ha dicho, la expresión de $\mathbf{R}_{\text{pelvis}}$ y $\mathbf{A}_{\text{pelvis}}$ se obtienen a partir de la posición del origen del sistema de coordenadas asociado a él y de los tres ángulos de Tait-Bryan, que constituyen las coordenadas generalizadas asociadas a la pelvis. Por tanto, la ecuación (2.3) se escribe como:

$$\mathbf{r}_{\text{pelvis}}(\mathbf{q}) = \mathbf{R}_{\text{pelvis}}(\mathbf{q}) + \mathbf{A}_{\text{pelvis}}(\mathbf{q}) \mathbf{u}_{\text{pelvis}} \quad (2.4)$$

La matriz de rotación del fémur, el siguiente en la cadena cinemática, se define a partir de la orientación relativa de este sólido respecto a la pelvis. Esta orientación se define a partir de la matriz de giro de la cadera, que es la articulación entre ambos sólidos. Por tanto la matriz de rotación del fémur se calcula de la siguiente manera:

$$\mathbf{A}_{\text{femur}} = \mathbf{A}_{\text{pelvis}} \mathbf{A}_{\text{cadera}} \quad (2.5)$$

El origen del sistema de coordenadas asociado al fémur también se define de manera relativa al de la pelvis, ya que se encuentra en la cadera, la articulación que une ambos sólidos. Con la posición y la matriz de rotación asociada se puede calcular la posición de cualquier punto perteneciente al fémur:

$$\mathbf{r}_{\text{femur}} = \mathbf{R}_{\text{femur}} + \mathbf{A}_{\text{femur}} \mathbf{u}_{\text{femur}} \quad (2.6)$$

Esto se va haciendo sucesivamente para cada sólido del modelo, desde el más proximal al más distal. Este resulta el principal inconveniente del uso de coordenadas relativas, ya que la posición de los segmentos más distales se define en función de la posición de todos los segmentos que le anteceden en la cadena cinemática. De esta manera se complican las expresiones de las matrices de rotación de estos segmentos más distales, ya que dependen de un gran número de coordenadas generalizadas.

2.3 Ecuaciones dinámicas

Si en el apartado anterior se explicó la cinemática del modelo, es decir, como se describen los movimientos del mismo, en el presente se explica como se hallan las fuerzas que producen dicho movimiento. La dinámica del modelo se resuelve aplicando las ecuaciones de Newton-Euler al conjunto de sólidos que componen el modelo, como se muestra en la ecuación (2.7)

$$\underbrace{\begin{bmatrix} m^i \mathbf{1} & & & & \\ & \ddots & & & \\ & & m^i \mathbf{1} & & \\ & & & \bar{\mathbf{I}}^i & \\ & & & & \ddots \\ & & & & & \bar{\mathbf{I}}^i \end{bmatrix}}_{\hat{\mathbf{M}}} \underbrace{\begin{bmatrix} \mathbf{a}_G^i \\ \vdots \\ \mathbf{a}_G^i \\ \bar{\alpha}^i \\ \vdots \\ \bar{\alpha}^i \end{bmatrix}}_{\hat{\mathbf{Q}}} = \underbrace{\begin{bmatrix} \mathbf{F}^i \\ \vdots \\ \mathbf{F}^i \\ \bar{\mathbf{M}}^i \\ \vdots \\ \bar{\mathbf{M}}^i \end{bmatrix}}_{\hat{\mathbf{Q}}} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -\bar{\omega}^i \wedge \bar{\mathbf{I}}^i \bar{\omega}^i \\ \vdots \\ -\bar{\omega}^i \wedge \bar{\mathbf{I}}^i \bar{\omega}^i \end{bmatrix}}_{\hat{\mathbf{Q}}_v} \quad (2.7)$$

A continuación se explica el desarrollo matemático que permite escribir la ecuación (2.7) en función de las coordenadas generalizadas del modelo tal y como explica en el documento del profesor Escalona [7]. El primer paso consiste en expresar el vector de aceleraciones lineales y angulares en función de las coordenadas generalizadas, como muestra la ecuación (2.8).

$$\begin{bmatrix} \mathbf{a}_G^i \\ \vdots \\ \mathbf{a}_G^i \\ \alpha^i \\ \vdots \\ \bar{\alpha}^i \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{H}^i \\ \vdots \\ \mathbf{H}^i \\ \bar{\mathbf{G}}^i \\ \vdots \\ \bar{\mathbf{G}}^i \end{bmatrix}}_{\mathbf{L}} \ddot{\mathbf{q}} + \underbrace{\begin{bmatrix} \mathbf{h}^i \\ \vdots \\ \mathbf{h}^i \\ \bar{\mathbf{g}}^i \\ \vdots \\ \bar{\mathbf{g}}^i \end{bmatrix}}_{\mathbf{I}} \dot{\mathbf{q}} \quad (2.8)$$

Para poder escribir esta ecuación los valores de las coordenadas generalizadas obtenidos en la cinemática inversa se derivan de forma numérica para obtener $\dot{\mathbf{q}}$ y $\ddot{\mathbf{q}}$. La construcción de las matrices \mathbf{L} y \mathbf{I} requiere de los valores de \mathbf{R}_G^i y de ω_i . La posición del centro de gravedad de los segmentos del modelo se extrae del modelo de cinemática inversa. Para las velocidades angulares, en un principio se siguió el documento del profesor Escalona [7], que presenta una forma general para realizar este cálculo a partir de las matrices de rotación:

$$\tilde{\omega}^i = \mathbf{A}^{iT} \dot{\mathbf{A}}^i \quad (2.9)$$

Sin embargo, al usar coordenadas relativas, la expresión de las matrices de rotación de los segmentos distales puede complicarse bastante. Por ejemplo:

$$\mathbf{A}_{\text{tibia}} = \mathbf{A}_{\text{pelvis}} \mathbf{A}_{\text{cadera}} \mathbf{A}_{\text{rodilla}} \quad (2.10)$$

Es fácil entender la dificultad de tener que derivar una matriz de este tipo. La solución pasa por utilizar esta formulación tan solo para la pelvis y la combinación de velocidades angulares para el resto de segmentos. De esta manera se calcula la velocidad angular de cada segmento a partir de la velocidad angular del segmento anterior:

$$\omega_{i-1} = \omega_{i-1} + \omega_{i-1} \quad (2.11)$$

Donde ω_{i-1} es la velocidad angular absoluta del sólido i , ω_{i-1} la velocidad angular absoluta del sólido $i-1$ y ω_{i-1} es la velocidad angular relativa del sólido i respecto al sólido $i-1$. Esta velocidad angular es la asociada a la articulación que une los dos segmentos consecutivos. Su expresión es fácil de conseguir debido al uso de coordenadas relativas en el modelo. De esta manera, una vez obtenida la velocidad angular de la pelvis, la expresión de las velocidades angulares asociadas a espalda y cadera es inmediata ya que tienen la misma expresión pero con el conjunto de coordenadas generalizadas asociado a la articulación.

A partir de \mathbf{R}_G^i y ω_i se obtienen las siguientes matrices:

$$\mathbf{H}^i = \frac{\partial \mathbf{R}_G^i}{\partial \mathbf{q}}, \bar{\mathbf{G}}^i = \frac{\partial \tilde{\omega}^i}{\partial \dot{\mathbf{q}}} \quad (2.12)$$

$$\mathbf{h}^i = \frac{\partial \mathbf{H}^i \dot{\mathbf{q}}}{\partial \mathbf{q}}, \bar{\mathbf{g}}^i = \frac{\partial \tilde{\omega}^i}{\partial \mathbf{q}} \quad (2.13)$$

Con estas ecuaciones ya se pueden construir las matrices \mathbf{L} y \mathbf{I} de la ecuación (2.8) Sustituyendo la ecuación 2.8 en la (2.7) queda

$$\hat{\mathbf{M}}(\mathbf{L}\ddot{\mathbf{q}} + \mathbf{I}\dot{\mathbf{q}}) = \hat{\mathbf{Q}} + \hat{\mathbf{Q}}_v \quad (2.14)$$

Premultiplicando por L^T

$$M\ddot{q} = Q + Q_v \quad (2.15)$$

Donde

$$M = L^T \hat{M} L, Q = L^T \hat{Q}, Q_v = L^T (\hat{Q}_v - \hat{M} l \dot{q}) \quad (2.16)$$

De la forma en que esta definido el problema no existen fuerzas de reacción como tal ya que no se definen ecuaciones de restricción sobre las coordenadas generalizadas. Por tanto, las únicas fuerzas existentes son las asociadas a las coordenadas generalizadas y las gravitatorias:

$$Q = Q_{gdl} + Q_{grav} \quad (2.17)$$

Por tanto, la ecuación (2.15) se puede escribir de la siguiente manera:

$$M\ddot{q} = Q_{gdl} + Q_{grav} + Q_v \quad (2.18)$$

Despejando Q_{gdl} en la anterior ecuación se llega a la ecuación (2.19)

$$Q_{gdl} = M(q)\ddot{q} - Q_v(q, \dot{q}) - Q_{grav}(q) \quad (2.19)$$

Donde:

- Q_{gdl} es el vector de fuerzas generalizadas asociadas a las coordenadas generalizadas
- $M(q)$ es la matriz de masas del sistema
- \ddot{q} es la derivada segunda de las coordenadas generalizadas
- $Q_v(q, \dot{q})$ es el vector de fuerzas cuadráticas en velocidad
- $Q_{grav}(q)$ es el vector de fuerzas gravitatorias

En la ecuación (2.19) todos los términos del segundo miembro son conocidos ya que el movimiento es conocido y se trata de obtener Q_{gdl} . Al existir una fuerza generalizadas por cada coordenada generalizada (\ddot{q} y Q_{gdl} tienen en mismo tamaño) se tienen el mismo numero de ecuaciones que de incógnitas. Una vez obtenido Q_{gdl} no queda mas que calcular la matriz con los vectores de posición de los puntos de aplicación de las fuerzas S para obtener el vector de momentos en las articulaciones T :

$$T = S \backslash Q_{gdl} \quad (2.20)$$

2.4 El ciclo de marcha

La marcha humana se divide en dos fases principales:

- Fase de apoyo, en la que el pie de referencia está en contacto con el suelo. Ocupa aproximadamente el 60% del tiempo de un ciclo.

- Fase de balanceo, en la que el pie de referencia está suspendido en el aire. Le corresponde el 40% del tiempo de un ciclo.

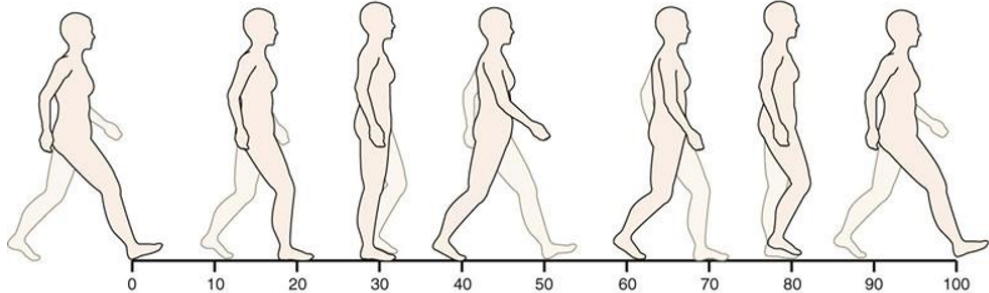
Por tanto, un ciclo de marcha va desde el apoyo en el suelo de un pie hasta el siguiente apoyo del mismo pie. Durante este ciclo el cuerpo da dos pasos. La longitud que se recorre entre que apoya el pie de referencia (inicio del ciclo) y apoya el contrario hace un paso y hasta que vuelve a apoyar el pie de referencia hace otro, completando un ciclo. Durante este ciclo hay momentos en los que todo el peso del cuerpo recae en un solo pie, lo que se denomina apoyo simple. Por contra, cuando el peso del cuerpo se distribuye sobre ambos pies se está en doble apoyo.

Dentro de la fase de apoyo se distinguen cuatro fases. El fin de cada fase marca el inicio de la siguiente:

- Contacto del talón: empieza con el primer contacto con el suelo, el cual se realiza en el talón y termina cuando deja de apoyar el pie que no es de referencia, en este caso el izquierdo. En esta fase, por tanto, se está en doble apoyo. En esta fase los dedos del pie de referencia aun no han apoyado, toda la fuerza del impacto debe ser amortiguada en el talón.
- Apoyo medio: en esta fase el pie se apoya en su borde más lateral. Durante esta fase el cuerpo está en apoyo simple. Esta fase termina cuando deja de contactar el talón del pie de referencia, hecho que se produce aproximadamente a la mitad del tiempo que dura la fase de apoyo.
- Propulsión activa: Todo el peso del cuerpo recae en la parte delantera del pie de referencia, que pasa de funcionar como un adaptador del movimiento a servir como palanca rígida que impulsa el cuerpo hacia delante; es en esta transición donde suelen fallar los mecanismos del pie, produciendo una anomalía al caminar. Esta fase termina cuando empieza a apoyar el pie que no es de referencia, comenzando de nuevo el doble apoyo.
- Propulsión pasiva: esta fase coincide con la fase de contacto del talón del pie contrario al de referencia. Termina cuando deja de apoyar el pie de referencia, que entra así en la fase de balanceo.

Dentro de la fase de balanceo se distinguen tres fases:

- Balanceo Inicial: en esta fase el ritmo de balanceo acelera.
- Balanceo Intermedio: esta fase constituye un periodo de transición
- Balanceo Terminal: en esta fase el ritmo de balanceo decelera.



Contacto del talón	Apoyo Medio	Propulsión Activa	Propulsión Pasiva	Balaneo Inicial	Balaneo Intermedio	Balaneo Terminal
Doble Apoyo	Apoyo Simple		Doble Apoyo	Apoyo Simple		
Fase de Apoyo				Fase de Balanceo		

Figura 2.5 Ciclo de marcha.

3 Modelo Utilizado

El modelo utilizado en este trabajo es el denominado gait2392, uno de los incluidos al descargar el software de OpenSim. Este modelo se utiliza para analizar la marcha humana y fue desarrollado por Delp *et al.* [6]. Su nombre se debe a que consta de 23 grados de libertad y 92 actuadores que representan 76 músculos del torso y las extremidades inferiores. La Figura 3.1 muestra este modelo. Sobre este modelo original se han realizado dos simplificaciones con el fin de reducir los tiempos de ejecución de los programas que resuelven la cinemática y la dinámica inversa. A continuación se analizan todas las características del modelo original en primer lugar y las simplificaciones llevadas a cabo en segundo para finalmente explicar como se han implementado en Matlab.

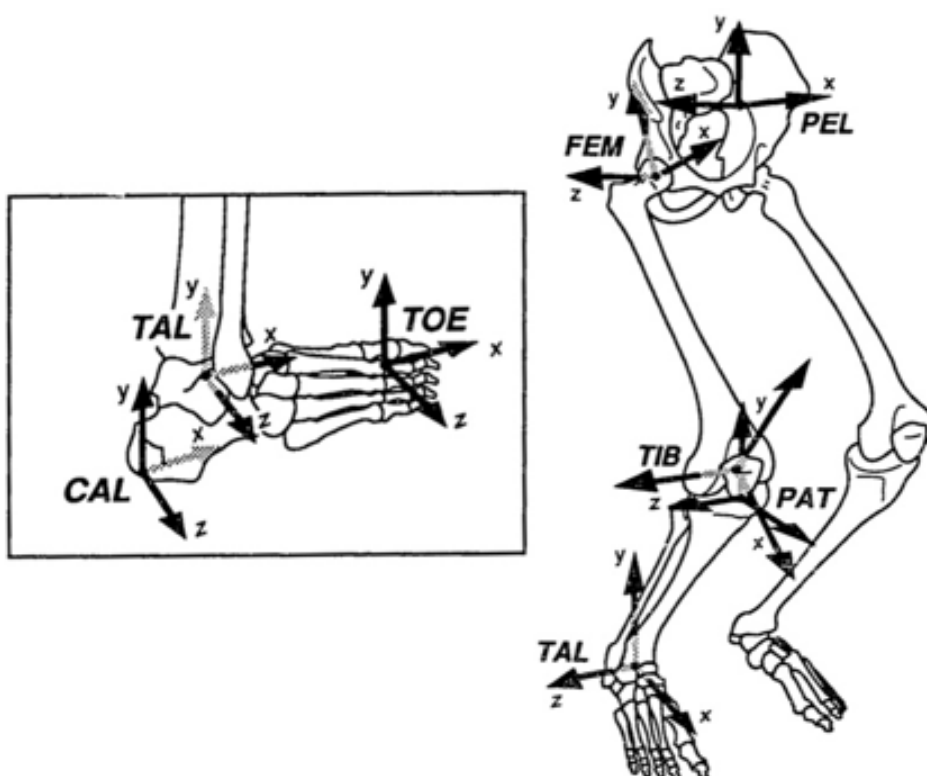


Figura 3.1 Modelo Gait2392.

3.1 Modelo Original

3.1.1 Cuerpos

El modelo consta de 12 cuerpos con características de sólido rígido que representan torso, pelvis y ambas extremidades inferiores. A cada uno de ellos se le asocia un sistema de coordenadas locales solidario con el.

- Torso: este cuerpo modela toda la parte superior del cuerpo, es decir, torso, cabeza y ambos brazos.
- Pelvis: el origen del sistema de coordenadas asociado se encuentra en el punto medio de la línea que une los puntos antero-superior de la cresta iliaca. Este cuerpo es el primero de la cadena cinemática.
- Fémur: el origen del sistema de coordenadas asociado se encuentra en la cabeza del fémur
- Tibia: el origen del sistema de coordenadas asociado en el punto medio de la línea que une los epicóndilos femorales
- Talón: el origen del sistema de coordenadas asociado punto medio de los ápices de los maleolos medial y lateral
- Calcáneo: el origen del sistema de coordenadas asociado se encuentra en el punto mas interior y lateral de la superficie posterior del hueso del calcáneo
- Dedos: el origen del sistema de coordenadas asociado se encuentra en la base del segundo metatarso. En principio nos encontramos con un modelo de pie multisegmento, formado por 3 sólidos articulados entre si: talón, calcáneo y dedos.

3.1.2 Articulaciones

El modelo consta de 11 articulaciones entre los sólidos definidos anteriormente.

- Espalda: entre el torso y la pelvis. En la Figura 3.2 se aprecian los movimientos que se producen en esta articulación:
 - Flexión/extensión: se trata de una rotación alrededor del eje latero-lateral.
 - Flexión lateral: se trata de una rotación alrededor del eje anteroposterior.
 - Rotación: se trata de una rotación alrededor del eje craneocaudal.
- Cadera: entre los sólidos pelvis y fémur. En la Figura 3.3 se aprecian los movimientos que se producen en esta articulación:
 - Flexión/extensión: se trata de una rotación alrededor del eje latero-lateral.
 - Abducción/adducción: se trata de una rotación alrededor del eje anteroposterior.
 - Rotación interna/externa: se trata de una rotación alrededor del eje craneocaudal.
- Rodilla: entre los sólidos fémur y tibia. En la Figura 3.4 se aprecia el movimiento de flexión/extensión que se produce entre ambos sólidos. Este movimiento se produce alrededor del eje latero-lateral.
- Tobillo: entre tibia y talón. En la Figura 3.5 se aprecia el movimiento de flexión plantar (plantar flexion)/flexión dorsal (dorsiflexion) que se produce entre ambos sólidos. Este movimiento se produce alrededor del eje latero-lateral.
- Articulación subtalar: entre talón y calcáneo.
- Articulación metatarso-falángica: entre calcáneo y dedos.

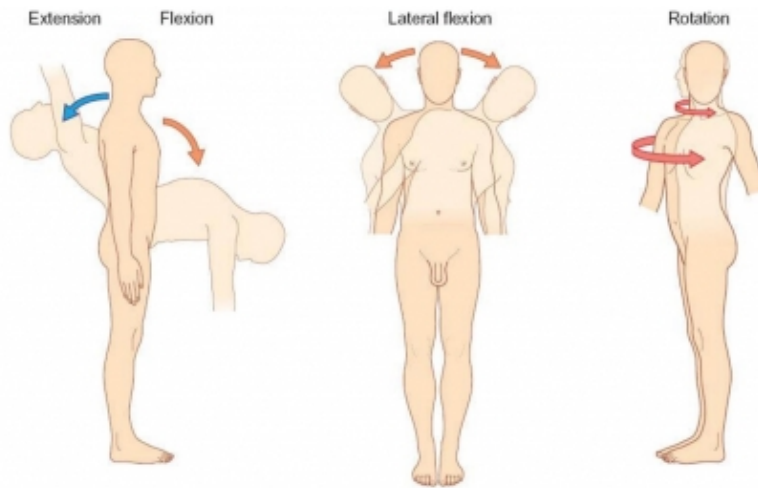


Figura 3.2 Movimiento de la espalda.

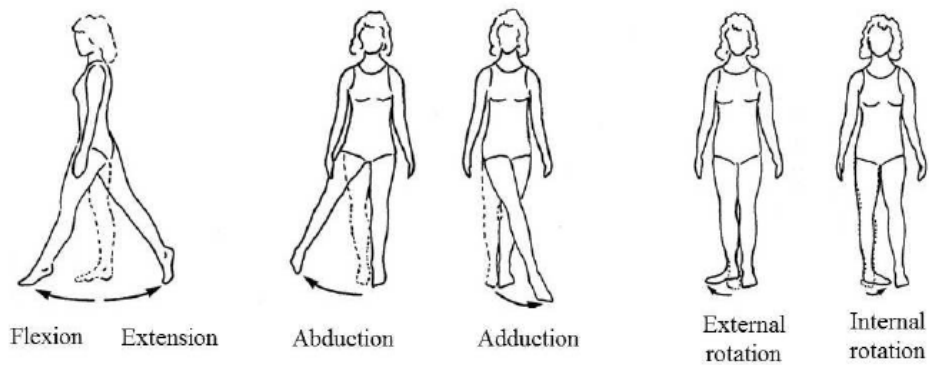


Figura 3.3 Movimiento de la cadera.

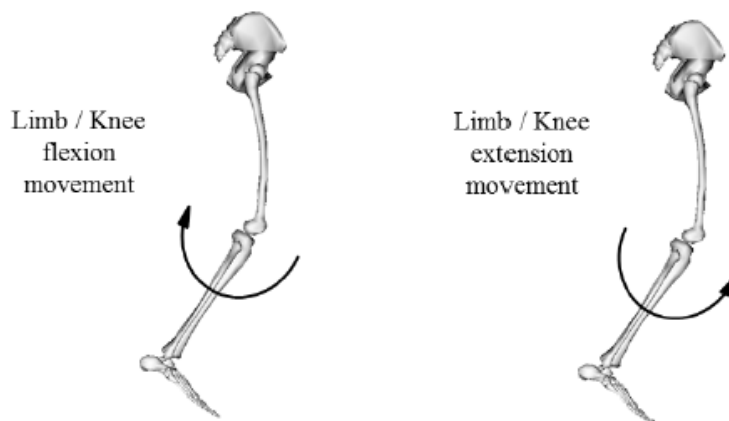


Figura 3.4 Movimiento de la rodilla.

3.1.3 Coordenadas

El modelo utiliza coordenadas generalizadas relativas. Asociada a la pelvis se tienen 6 coordenadas generalizadas que definen su posición y orientación en el espacio tridimensional. Posteriormente,

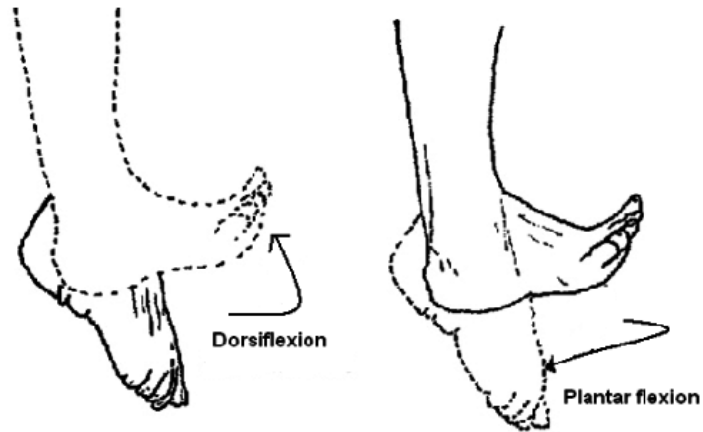


Figura 3.5 Movimiento del tobillo.

para cada articulación se establece un número de coordenadas generalizadas igual al número de grados de libertad que permite la articulación. Estas coordenadas, por tanto, describen el movimiento relativo entre los sólidos. Las coordenadas constituyen una forma matemática de caracterizar los movimientos de las articulaciones descritos en el apartado anterior. En total se tiene un conjunto de 23 coordenadas generalizadas se denota por \mathbf{q}

$$\mathbf{q} = \begin{pmatrix} pelvis_tx \\ pelvis_ty \\ pelvis_tz \\ pelvis_tilt \\ pelvis_list \\ pelvis_rotation \\ hip_flexion_r \\ hip_adduction_r \\ hip_rotation_r \\ knee_angle_r \\ ankle_angle_r \\ subtalar_angle_r \\ mtp_angle_r \\ hip_flexion_l \\ hip_adduction_l \\ hip_rotation_l \\ knee_angle_l \\ ankle_angle_l \\ subtalar_angle_l \\ mtp_angle_l \\ lumbar_extension \\ lumbar_bending \\ lumbar_rotation \end{pmatrix} \quad (3.1)$$

- Pelvis: las coordenadas pelvis tx, pelvis ty y pelvis tz describen la traslación en los ejes X, Y y Z globales, respectivamente. Se miden en metros. Las coordenadas pelvis tilt, pelvis list y pelvis rotation son los tres ángulos de Tait-Bryan que permiten pasar del sistema de coordenadas global al sistema de coordenadas local de la pelvis.

- Espalda: esta articulación se modela como un par esférico, por lo que permite tres rotaciones y las tres coordenadas generalizadas son lumbar extensión, lumbar bending y lumbar rotation, que constituyen los tres ángulos de Tait-Bryan con los que se puede construir la matriz de rotación de esta articulación. Esta matriz de rotación permite pasar del sistema de coordenadas de la pelvis al del torso.
- Cadera: se establece un par esférico, que permite tres rotaciones y al que se le asocian las tres coordenadas generalizadas hip flexion, hip adduction y hip rotation, que constituyen los tres ángulos de Tait-Bryan con los que se puede construir la matriz de rotación de esta articulación. Esta matriz de rotación permite pasar del sistema de coordenadas de la pelvis al del fémur.
- Rodilla: en ella se produce una rotación entre fémur y tibia, pero el centro de rotación instantáneo no se mantiene fijo durante el movimiento. Se trata de un movimiento plano (en el plano que forman los ejes x y locales de la tibia) del centro de rotación instantáneo que depende del valor de la coordenada knee angle, como se muestra en la Figura 3.6. Más concretamente se utilizan splines que en función del valor de la coordenada knee angle dan los valores del movimiento en el plano. La rotación se produce alrededor del eje z del fémur. Esta articulación permite un grado de libertad, ya que conociendo el valor de la coordenada generalizada asociada (knee angle) se conoce tanto el movimiento del eje instantáneo de rotación como la rotación en sí.

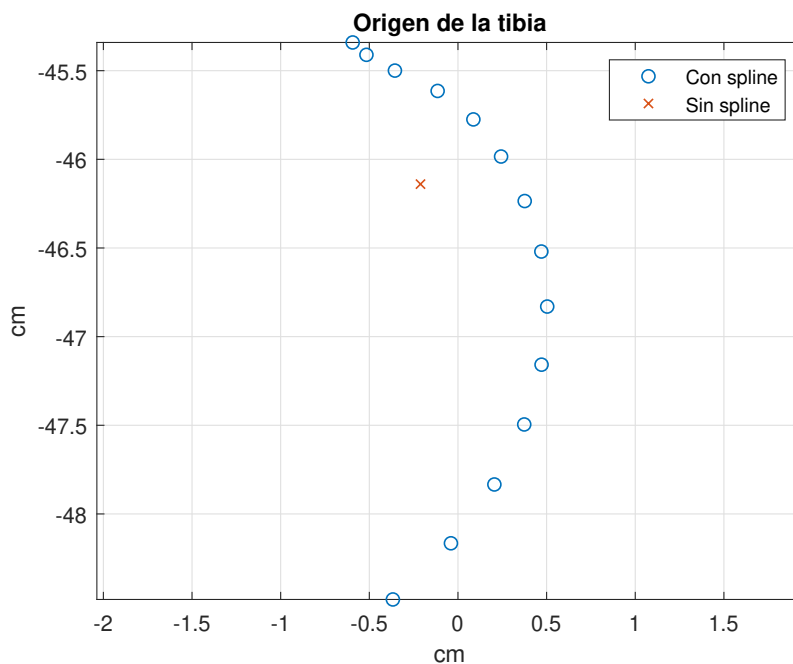


Figura 3.6 Origen de la tibia en el plano sagital.

- Tobillo: esta articulación se modela como un par de revolución entre tibia y talón cuya coordenada generalizada asociada es ankle angle. La matriz de rotación de esta articulación se construye a partir de parámetros de Euler donde la dirección del eje de revolución es un parámetro tomado del modelo y el ángulo de rotación es la coordenada generalizada ankle angle.
- Articulación subtalar: similar a la articulación del tobillo pero entre talón y calcáneo. La coordenada generalizada asociada es subtalar angle.

- Articulación metatarso-falángica: similar a la articulación del tobillo pero entre calcáneo y dedos. La coordenada generalizada asociada es mtp angle. Las rotaciones asociadas a las coordenadas generalizadas ankle angle, subtalar angle y mtp angle (metatarsophalangeal angle) son alrededor de unos ejes que especifica el modelo y que se aprecian en la figura 3.7

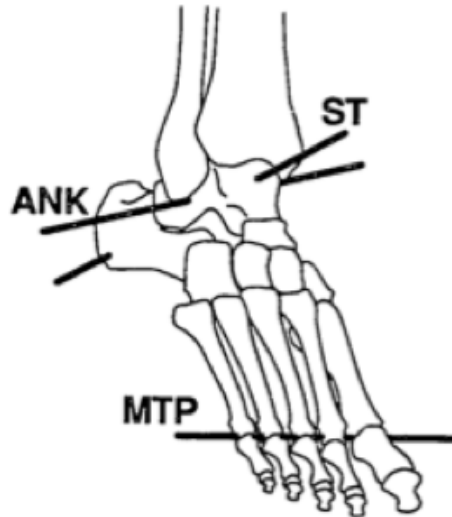


Figura 3.7 Ejes articulaciones del pie.

Conociendo el valor de las 23 coordenadas generalizadas queda definido la posición de cualquier punto del modelo. Además estas 23 coordenadas son independientes entre sí, no están sujetas a ninguna restricción. Por tanto se tiene un modelo de 23 grados de libertad. Como se ha visto en el apartado de orientación de sólidos en el espacio, la principal desventaja del uso de estas coordenadas es la expresión difícil de los últimos segmentos de la cadena cinemática; sin embargo, estas coordenadas permiten describir de manera unívoca la posición de un sistema multicuerpo y son independientes entre sí. Como se verá en el capítulo dedicado a la cinemática inversa, esto permite resolver el problema sin tener que imponer ninguna restricción sobre ellas.

3.1.4 Datos de entrada del modelo

Marcadores

Se tiene un total de 31 marcadores que se utilizarán en la resolución de la cinemática y la dinámica inversa.

- Torso
 - Sternum
 - R.Acromion
 - L.Acromion
 - TopHead
- Pelvis
 - R.ASIS
 - L.ASIS

- V.Sacral
- Fémur
 - Thigh.Upper
 - Thigh.Front
 - Thigh.Rear
- Tibia
 - Shank.Upper
 - Shank.Front
 - Shank.Rear
- Calcáneo
 - Heel
 - Midfoot.Sup
 - Midfoot.Lat
 - Toe.Lat
 - Toe.Med
 - Toe.Tip

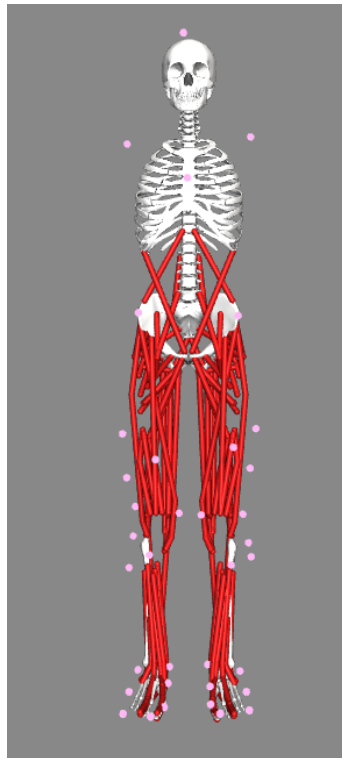


Figura 3.8 Posición de los marcadores: vista anterior.

Conviene aclarar que estos son los marcadores cuya posición se utiliza para resolver la cinemática inversa. La posición de estos marcadores aparece recogida en un fichero que constituye un dato de partida para el modelo. Sin embargo, para la construcción del modelo se entiende que se utilizó un menor número de ellos, como se explica en el siguiente apartado.

Parámetros cinemáticos

Los datos experimentales incluidos en el modelo se tomaron siguiendo las pautas dispuestas en el trabajo de Chand [9] pero usando un sujeto diferente al que se describe en dicho trabajo. El modelo general de OpenSim corresponde a un sujeto de unos 1.8 metros de altura y 75.16 kilogramos de peso. En el trabajo de Chand se realizaron mediciones sobre 8 sujetos sanos caminando a 4 velocidades diferentes, siguiendo el trabajo de Liu [10]. Las articulaciones subtalar y metatarso-falángica se bloquearon en ángulos anatómicos neutros. El protocolo de toma de datos fue el mismo definido en el trabajo de Schwartz [16]. En el se sigue un protocolo Plug-in-Gait usando cámaras Vicon. Este protocolo fue desarrollado en la Universidad de Oxford y se basa en el modelo Newington de marcha [11]. En el se usan de un número mínimo de marcadores para reconstruir el movimiento del sujeto. Se trata de un modelo predictivo, lo que quiere decir que se utilizan ecuaciones de regresión para obtener las coordenadas de los centros de articulaciones en el sistema de coordenadas local como función de parámetros antropométricos. La principal ventaja de este protocolo es que es mínimamente invasivo [11]. Se asume que mediante el uso de este protocolo se obtuvieron los orígenes de los sistemas de coordenadas asociados a cada segmento y las coordenadas de los centros de las articulaciones, que constituyen los parámetros cinemáticos del modelo.

Parámetros dinámicos

Los parámetros dinámicos del modelo son necesarios para la resolución de la dinámica inversa. Los datos necesarios son las masas y momentos de inercia de cada segmento del modelo. Con estos datos se construye la matriz de masa que aparece en la ecuación dinámica. En este trabajo estos parámetros se han tomado del código de OpenSim. Estos parámetros están basados en datos medios antropométricos obtenidos de cinco sujetos para todos los segmentos excepto para calcáneo y dedos. Para estos últimos los datos se obtienen de representar su volumen a partir de la medida de una zapatilla de tenis [2].

3.2 Modelos simplificados

La primera simplificación realizada consiste en considerar el pie como monosegmento. Aunque, como se ha visto, en el modelo original el pie esta formado por tres cuerpos (talón, calcáneo y toes) unidos por dos articulaciones (subtalar y metatarso-falángica) en las opciones por defecto de OpenSim estas articulaciones se fuerzan a cero, por lo que en realidad el pie se modela como un segmento único. Esta simplificación se entenderá mucho mejor tras haber leído el capítulo dedicado a la cinemática inversa.

La segunda simplificación se tomo durante el desarrollo del módulo de dinámica inversa debido a la dificultad del tratamiento del spline de la rodilla. De esta manera, se elimino el desplazamiento relativo entre tibia y fémur convirtiéndose la articulación de la rodilla en un par de rotación sencillo. La posición de este centro de rotación se hizo fija y se tomo como la media de las posiciones para cada uno de los valores del spline.

En la Figura 3.6 se representa el movimiento que realiza el origen de la tibia en el plano sagital considerando el spline para todos los valores posibles de ángulo de flexión de la rodilla comparado con el eje fijo que se toma al eliminar el spline.

4 Cinemática Inversa

En el presente capítulo se explica cómo se ha resuelto el problema cinemático inverso. El objetivo de este problema es calcular el valor de las coordenadas generalizadas a partir de las coordenadas de los marcadores del modelo. La obtención del valor de estas coordenadas se ha explicado en el apartado de medidas experimentales y como se utilizan para obtener los parámetros cinemáticos del modelo se ha explicado en el apartado correspondiente del segundo capítulo. Con estos datos de partida ya se puede plantear el problema cinemático inverso. En este capítulo se explica en primer lugar como se formula, para posteriormente describir como se ha implementado en Matlab y finalmente presentar los resultados obtenidos y su comparación con los de OpenSim.

4.1 Formulación

De la manera en que está planteado el problema se tiene un mayor número de ecuaciones que incógnitas. Las ecuaciones son la posición de 31 marcadores (de valor conocido) en función de las coordenadas generalizadas. Por tanto se tienen 93 ecuaciones (3 por cada marcador) frente a 23 incógnitas (el valor de las coordenadas generalizadas). Se trata por tanto frente a un sistema de ecuaciones sobredeterminado. La solución pasa por tanto por encontrar el valor de las coordenadas generalizadas no que cumplan las 93 ecuaciones si no que produzcan el menor error posible. Tenemos por tanto una optimización numérica en la que la función a minimizar es el error cometido y se define de la siguiente manera [5].

$$\min \left[\sum_{i \in \text{marcadores}} \omega_i \|\mathbf{x}_i^{\text{exp}} - \mathbf{x}_i(\mathbf{q})\|^2 + \sum_{j \in \text{coordenadas prescritas}} \omega_j (q_j^{\text{exp}} - q_j)^2 \right] \quad (4.1)$$

Donde:

- ω_i es un peso asociado al marcador i
- ω_j es un peso asociado a la coordenada generalizada j
- $\mathbf{x}_i^{\text{exp}}$ es la posición en globales del marcador i registrada en el laboratorio
- $\mathbf{x}_i(\mathbf{q})$ es la posición en globales del marcador i calculada analíticamente a partir de los valores de las coordenadas generalizadas, como se ha explicado en el capítulo dedicado a los conceptos teóricos.
- q_j^{exp} es el valor de la coordenada generalizada j que el usuario puede forzar a que tenga un valor determinado (coordenada de valor prescrito)
- q_j es el valor de la coordenada generalizada j obtenida de manera analítica

Se aprecia que este error tiene una componente asociada a la posición de los marcadores y otra asociada a las coordenadas generalizadas. La componente asociada a la posición de los marcadores es la obvia, ya que se pretende que la posición calculada de los marcadores se parezca lo máximo posible a la obtenida de manera experimental. La contribución del error de cada marcador se controla mediante el peso asociado. En este trabajo se han utilizado los vectores de pesos por defecto de OpenSim. En cuanto al vector de pesos ω_i este contiene pesos mas altos para los marcadores mas cercanos al hueso y mas bajos para aquellos que pueden moverse debido al movimiento del tejido blando. Esto se debe a que este movimiento de tejido blando desvirtúa la hipótesis tomada de que cada cuerpo es un solido rígido e introduce errores en la reconstrucción del movimiento muy importantes. De esta manera, para los marcadores R.ASIS y L.ASIS, colocados en el punto mas antero-superior de la cresta iliaca (un punto anatómico fácilmente reconocible y muy cercano al hueso de la pelvis), se tienen unos pesos de 10 frente a los marcadores colocados en el muslo, que tienen un peso asociado unidad ya que estarán sometidos al movimiento de toda la musculatura del muslo.

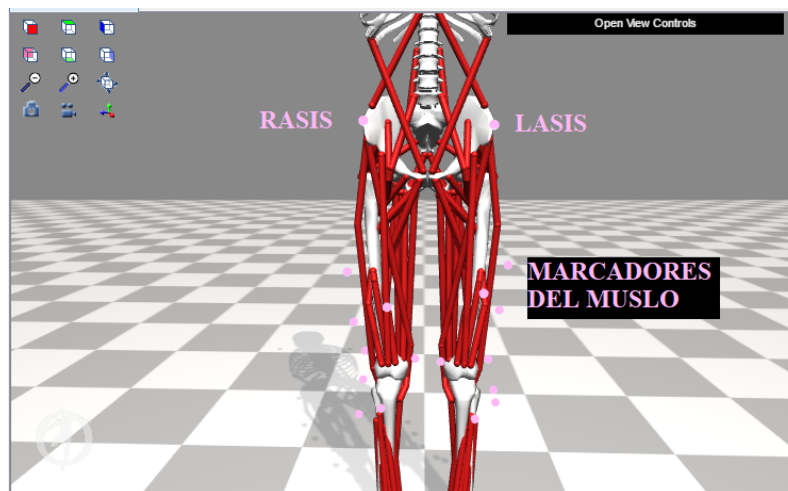


Figura 4.1 Marcadores de la pelvis y del muslo.

En la Tabla 4.1 se aprecia el peso ω_i asociado a cada marcador.

Por su parte, la segunda componente del error tan solo sirve si hay alguna coordenada de valor prescrito. En las opciones por defecto de OpenSim las únicas coordenadas de valor prescrito han sido las asociadas al movimiento del pie, es decir, *subtalar_angle* y *mtp_angle*. Ambas se fuerzan a cero con un peso asociado de 1000. De esta manera, aunque originalmente el modelo considera el pie formado por tres segmentos, en la practica se tiene un pie monosegmento.

En todo momento se sigue el protocolo de reconstrucción del movimiento conocido como LOM (Local Optimization Method). Este método se centra en minimizar el error cometido al obtener la posición de los marcadores en función de las coordenadas generalizadas sin imponer ninguna restricción cinemática sobre ellas. Este método se contrapone al método AM (Adaptation Method), el cual desprecia el error cometido limitándose a imponer sobre las posiciones de los marcadores las restricciones cinemáticas del modelo. [11]

En un principio se utilizo la formula presentada en la ecuación (4.1) para calcular el error. Los resultados obtenidos usando esta formulación se presentan como “modelo original Delp” en la sección de resultados. Posteriormente, se elimino la parte del error asociada al valor de las coordenadas generalizadas y se utilizo la restricción que permite $fmincon A_{eq}x = b_{eq}$ para forzar las

Tabla 4.1 Peso asociado a cada marcador.

Marcador	ω_i
Sternum	1
R.Acromium	0.5
L.Acromium	0.5
TopHead	0.1
R.ASIS	10
L.ASIS	10
V.Sacral	10
R.Thigh.Upper	1
R.Thigh.Front	1
R.Thigh.Rear	1
R.Shank.Upper	1
R.Shank.Front	1
R.Shank.Rear	1
R.Heel	10
R.Midfoot.Sup	1
R.Midfoot.Lat	1
R.Toe.Lat	1
R.Toe.Med	1
R.Toe.Tip	10
L.Thigh.Upper	1
L.Thigh.Front	1
L.Thigh.Rear	1
L.Shank.Upper	1
L.Shank.Front	1
L.Shank.Rear	1
L.Heel	10
L.Midfoot.Sup	1
L.Midfoot.Lat	1
L.Toe.Lat	1
L.Toe.Med	1
L.Toe.Tip	10

coordenadas a cero. La función objetivo queda como sigue:

$$\min \left[\sum_{i \in \text{marcadores}} \omega_i \|\mathbf{x}_i^{\text{exp}} - \mathbf{x}_i(\mathbf{q})\|^2 \right] \quad (4.2)$$

Donde:

- ω_i es un peso asociado al marcador i
- $\mathbf{x}_i^{\text{exp}}$ es la posición en globales del marcador i registrada en el laboratorio
- $\mathbf{x}_i(\mathbf{q})$ es la posición en globales del marcador i calculada analíticamente

Los resultados obtenidos usando esta formulación se presentan como “modelo original con bloqueo” en la sección de resultados. Tanto en estos resultados como los siguientes que se presentan se ha seguido esta formulación, ya que se consigue un menor tiempo de ejecución.

Una segunda mejora en la ejecución del código se desarrollo cuando se descubre que la parte mas costosa computacionalmente es la que corresponde a la construcción de los splines para hallar la posición del origen de la tibia. Así, se construyó un modelo similar al original pero no considerando el desplazamiento relativo entre fémur y tibia, por lo que la rodilla se modela como un par de rotación. Este modelo produjo una reducción del tiempo de ejecución bastante importante y se presenta en la sección de resultados como “modelo sin splines”.

La ultima mejora consistió en eliminar del modelo sin splines las coordenadas generalizadas asociadas a los grados de libertad del pie. De esta manera, los segmentos talón, calcáneo y dedos pasan a combinarse en uno solo denominado pie. Es asumible realizar esta simplificación ya que aunque el modelo original del pie es multisegmento, en la práctica los grados de libertad asociados a dichos segmentos se restringen. Con esta simplificación, al eliminarse los dos últimos segmentos de la cadena cinemática, que generan expresiones mas complicadas, la resolución del problema se vuelve más sencilla. Los resultados cinemáticos de este modelo se presentan en la sección de resultados como “modelo monosegmento sin splines”.

4.2 Modelado en Matlab

La optimización numérica se ha llevado a cabo usando la función de Matlab `fmincon`, un algoritmo que busca el mínimo de una función de varias variables no lineal y restringida [1]. Este algoritmo plantea el problema de minimización de la siguiente forma.

$$\min_x \mathbf{f}(\mathbf{x}) \text{ tal que } \begin{cases} \mathbf{c}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{c}_{\text{eq}} = \mathbf{0} \\ \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{b}_{\text{eq}} \\ \mathbf{l}_b \leq \mathbf{x} \leq \mathbf{u}_b \end{cases} \quad (4.3)$$

En un primer momento, los únicos parámetros que se utilizaron fueron los limites superior e inferior \mathbf{l}_b y \mathbf{u}_b con los mismos valores que toma OpenSim por defecto. Posteriormente se uso la restricción $\mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{b}_{\text{eq}}$.

En un primer momento se utilizo lenguaje simbólico para definir la formula del error a minimizar en el modulo. El objetivo era calcular el gradiente del error de forma simbólica para pasarlo a la función `fmincon`, mejorando de esta manera su rendimiento. Sin embargo, aunque se consiguió hallar de forma simbólica la formula del error, la extensión del gradiente hacia muy poco practico su uso. Los resultados actuales surgen de haber usado calculo numérico en MATLAB, después de haber realizado una optimización del código para mejorar el rendimiento, ya que en un primer momento el modulo empleaba un tiempo de unos 30 minutos, un tiempo prohibitivo. La optimización del código se llevo a cabo usando cuando era posible vectorización, al ser el modo en el que esta pensado Matlab para funcionar y minimizando el numero de operaciones a realizar. Por otra parte se ha utilizado una función recurrente siempre que ha sido posible. Esto consiste en aprovechar los cálculos ya realizados y supone una ventaja cuando se hace uso de coordenadas relativas.

En cuanto al algoritmo utilizado, el que mejor resultados ha dado ha sido el `sqp` (sequential quadratic programming) frente al interior point. Los criterios de convergencia han sido `TolFun` $1e-4$ (como OpenSim) y máximo numero de iteraciones 400 (frente a 1000 de OpenSim).

A continuación se describe de manera abreviada la estructura del código:

- `rot_mat`, `rot_pie`, `rot_rodilla`: estas funciones sirven para calcular las matrices de rotación empleadas en el modelo. La función `rot_mat` devuelve la matriz de rotación asociada a pelvis, cadera y espalda dados los tres ángulos de Euler en radianes; la matriz `rot_pie` devuelve la matriz de rotación asociada a tobillo, articulación subtalar o metatarso-falángica en función de la coordenada generalizada asociada; la función `rot_rodilla` devuelve la

matriz de rotación asociada a la rodilla dado el valor de la coordenada generalizada knee angle.

- `calcula_globales`: esta función devuelve la posición en coordenadas globales de un punto dados su posición en el sistema de coordenadas local así como el origen y la matriz de orientación asociados a este.
- `cinematica2392`: esta función devuelve las posiciones de los marcadores del modelo dado el vector de coordenadas generalizadas \mathbf{q} . En esta función se llama tanto a las funciones `rot_mat`, `rot_pie` y `rot_rodilla` como a la función `calcula_globales`.
- `sq_err`: esta función calcula el valor de la función objetivo para un instante de tiempo dados el vector de coordenadas generalizadas y las posiciones de los marcadores medidas de forma experimental. Esta función llama a la función `cinematica2392`.
- `ik_main`: es el código principal en el que se resuelve el problema cinemático inverso realizando la optimización descrita al principio del capítulo. En esta función se realiza un bucle en el que se llama a `fmincon` para minimizar la función `sq_err` para cada instante de tiempo. Una vez ejecutado este código se obtiene el valor de las coordenadas generalizadas para cada instante de tiempo, que se analizan en el siguiente apartado.

4.3 Presentación de resultados

4.3.1 Modelo original Delp

De la Figura 4.2 a la Figura 4.5 se representan los valores obtenidos de las coordenadas generalizadas frente al porcentaje del ciclo de marcha usando esta formulación comparados con los de OpenSim. Se observa que los resultados obtenidos son casi idénticos a los de OpenSim pero con un tiempo de ejecución mucho mayor, 73 segundos frente a tan solo 2. En la Figura 4.3 y la Figura 4.4 se observa que se obtienen valores cercanos a cero para las coordenadas subtalar angle y mtp angle, pero con mucho ruido.

El error se ha calculado usando la fórmula propuesta por [12] para hallar el error residual asociado a la posición de cada marcador:

$$e_i = \frac{1}{n_t} \sum_j \|\mathbf{x}_{ij}^{\text{exp}} - \mathbf{x}_{ij}(\mathbf{q})\| \quad (4.4)$$

Donde $\mathbf{x}_{ij}^{\text{exp}}$ es la posición en el instante j del marcador i registrada en el laboratorio, $\mathbf{x}_i(\mathbf{q})$ es la posición en el instante j del marcador i calculada analíticamente y n_t el número de instantes de tiempo considerados.

Posteriormente se calcula una media ponderada de los errores de cada marcador:

$$error = \frac{\sum_i \omega_i e_i}{\sum_i \omega_i} \quad (4.5)$$

Donde ω_i es un peso asociado al marcador i

Tabla 4.2 Comparación de métodos IK.

Modelo	Error (cm)	Tiempo de ejecución (s)
OpenSim	0.7797	2
Modelo original Delp	0.7795	73

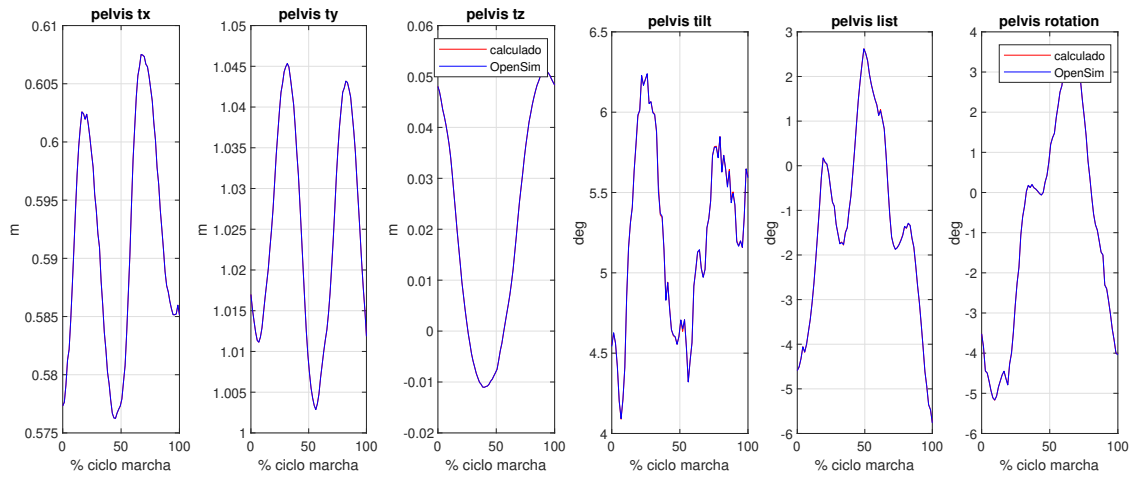


Figura 4.2 Resultados del modelo original Delp para la pelvis.

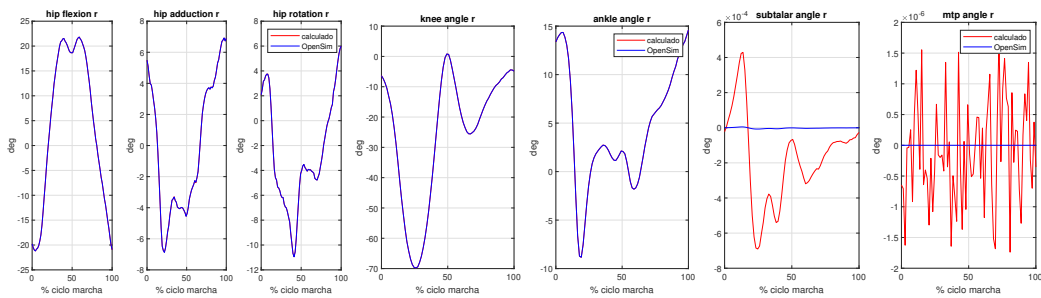


Figura 4.3 Resultados del modelo original Delp para la pierna derecha.

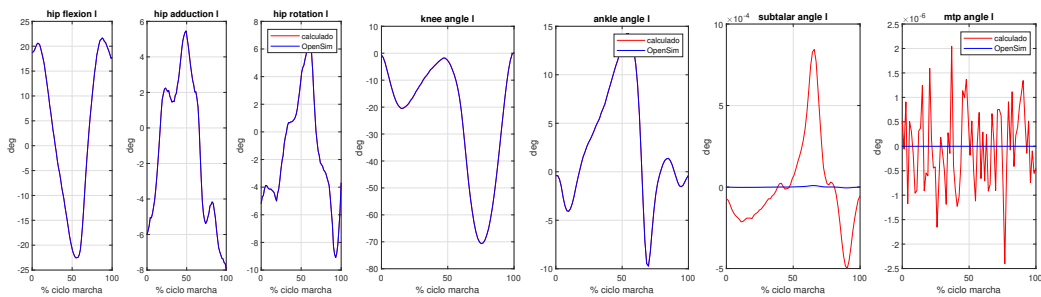


Figura 4.4 Resultados del modelo original Delp para la pierna izquierda.

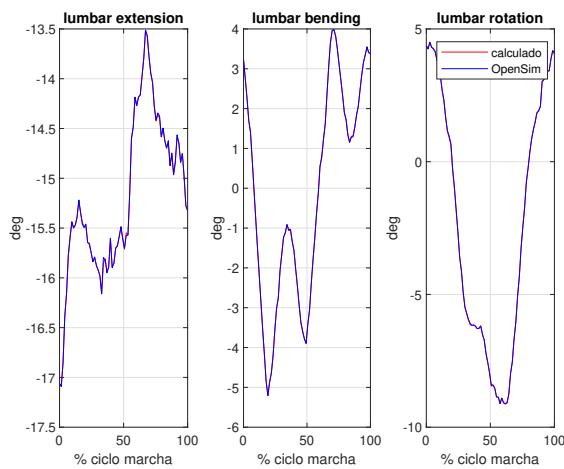


Figura 4.5 Resultados modelo original Delp para la espalda.

4.3.2 Modelo original con bloqueo

Los resultados al utilizar este modelo son muy parecidos a los obtenidos con el anterior modelo con la única diferencia presente en las coordenadas asociadas al movimiento del pie como se observa de la Figura 4.6 a la Figura 4.9. En la Figura 4.7 y la Figura 4.8 se observa que el valor de las coordenadas subtalar y metatarso-falángica son exactamente cero ya que en la optimización se ha hecho esa restricción. En la Tabla 4.3 se observa que se consigue una disminución del tiempo de ejecución de 73 a 59 segundos respecto al modelo anterior, lo cual supone una disminución del 19 %

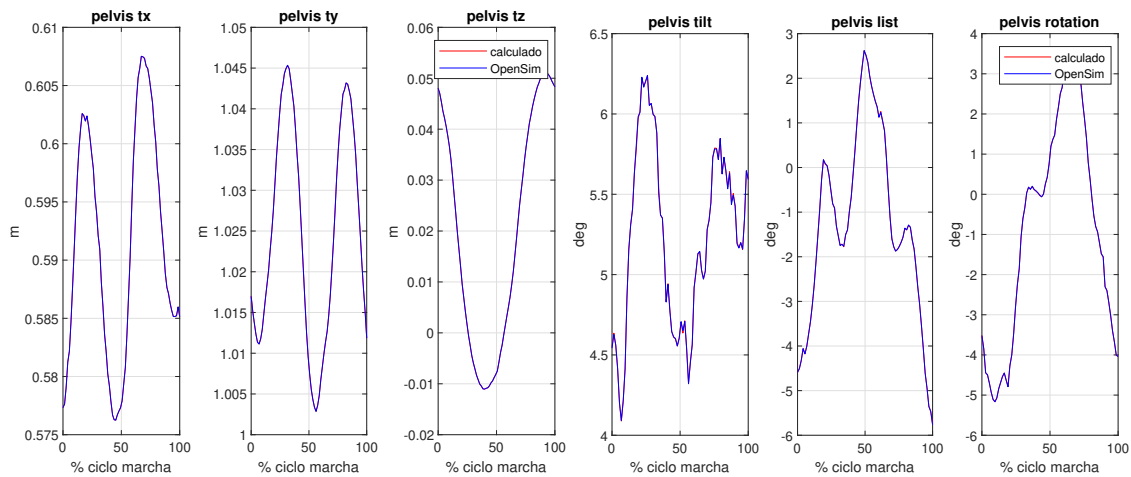


Figura 4.6 Resultados del modelo original con bloqueo para la pelvis.

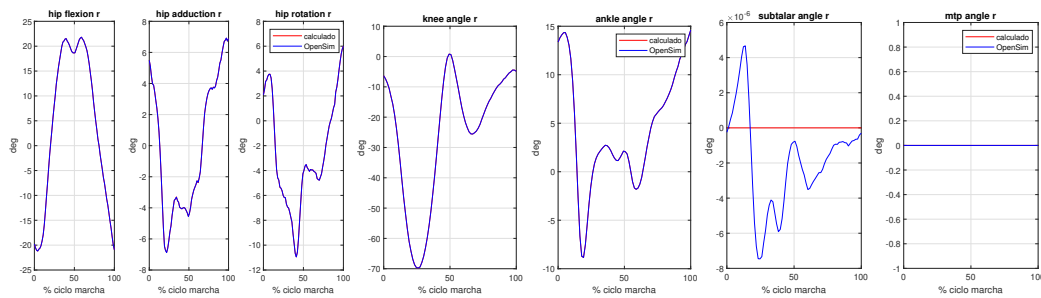


Figura 4.7 Resultados del modelo original con bloqueo para la pierna derecha.

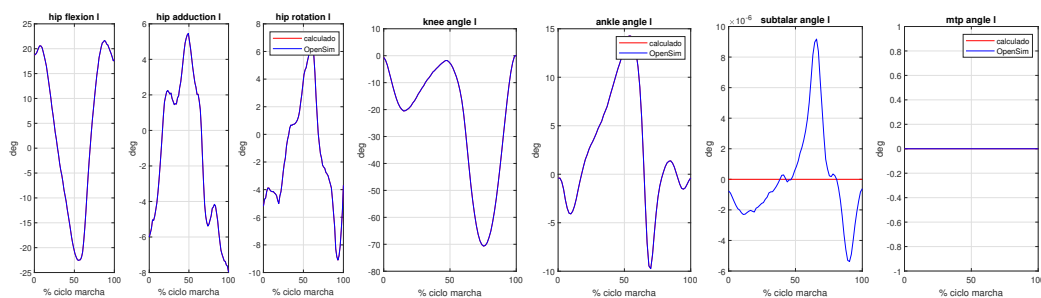


Figura 4.8 Resultados del modelo original con bloqueo para la pierna izquierda.

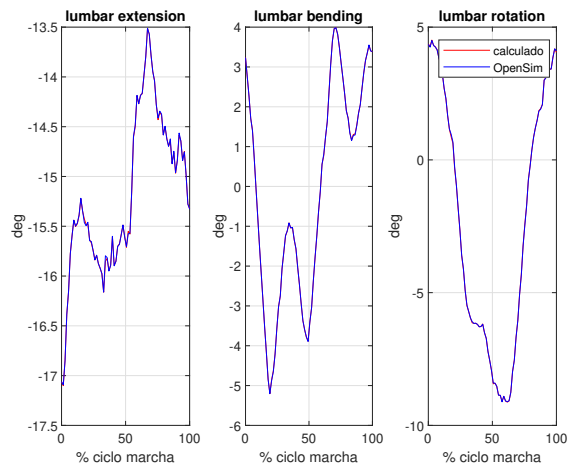


Figura 4.9 Resultados del modelo original con bloqueo para la espalda.

Tabla 4.3 Comparación de métodos IK.

Modelo	Error (cm)	Tiempo de ejecución (s)
OpenSim	0.7797	2
Modelo original Delp	0.7795	73
Modelo original con bloqueo	0.7795	59

4.3.3 Modelo sin splines

Usando este modelo se obtienen resultados que difieren mas de los de OpenSim como se aprecia de la Figura 4.10 a la Figura 4.13. Sin embargo, se tiene la ventaja de que el tiempo de ejecución pasa de 59 a 19 segundos (Tabla 4.4), lo que supone una reducción del 56%.

Tabla 4.4 Comparación de métodos IK.

Modelo	Error (cm)	Tiempo de ejecución (s)
OpenSim	0.7797	2
Modelo original Delp	0.7795	73
Modelo original con bloqueo	0.7795	59
Modelo sin splines	0.7847	19

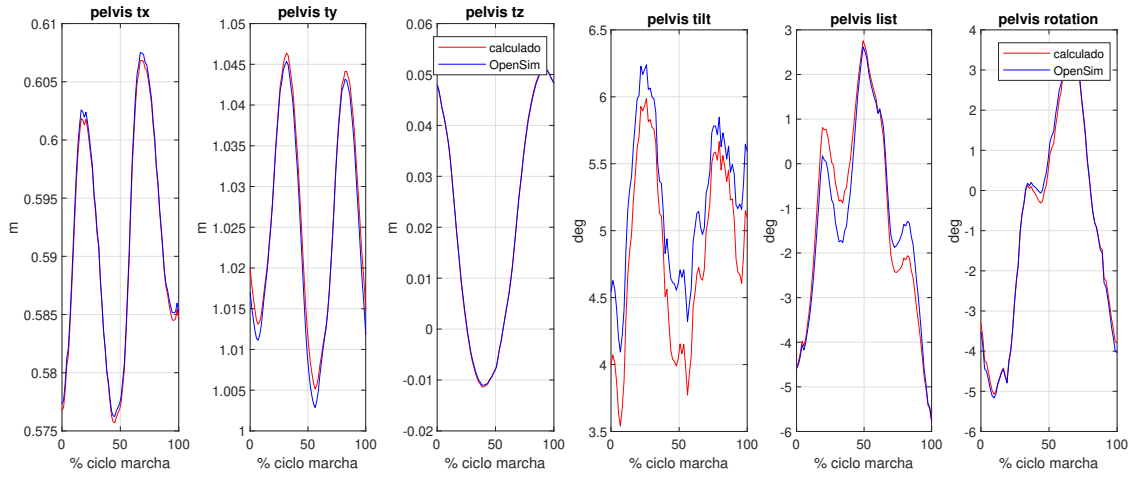


Figura 4.10 Resultados del modelo sin splines para la pelvis.

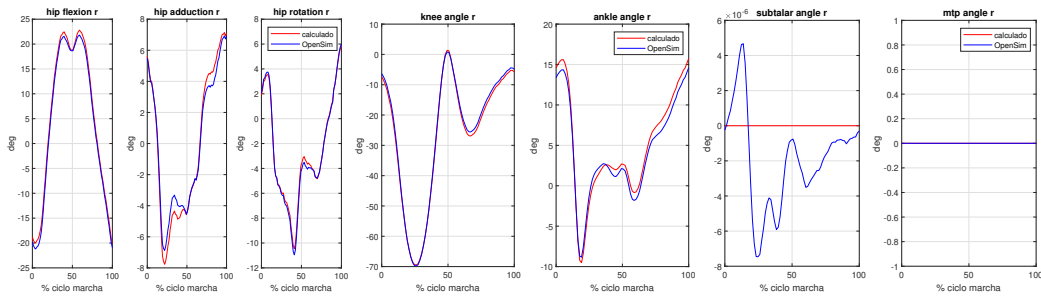


Figura 4.11 Resultados del modelo sin splines para la pierna derecha.

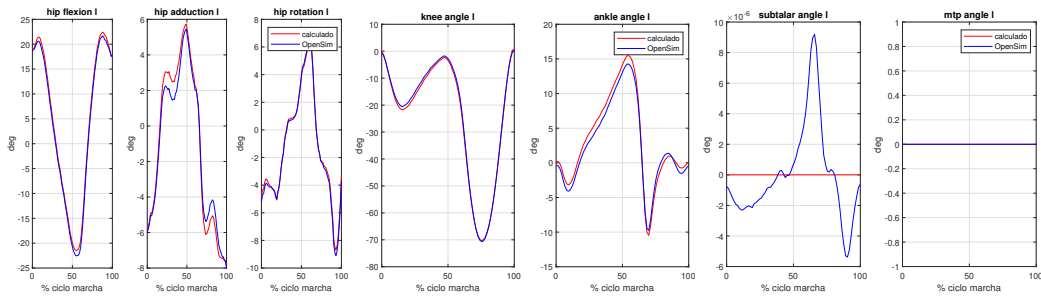


Figura 4.12 Resultados del modelo sin splines para la pierna izquierda.

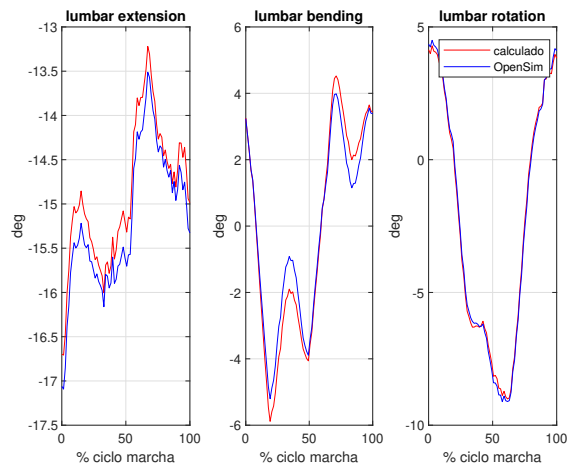


Figura 4.13 Resultados modelo sin splines para la espalda.

4.3.4 Modelo monosegmento sin splines

Los resultados cinemáticos de este último modelo son muy parecidos al modelo anterior, como se aprecia de la Figura 4.14 a la Figura 4.17. El tiempo de ejecución es el mismo que para el modelo anterior. El interés de este modelo reside en que permite resolver el problema dinámico inverso. En la Tabla 4.6 se aprecian los errores para cada marcador obtenidos usando cada modelo. Dicho error se ha calculado usando la ecuación (4.4).

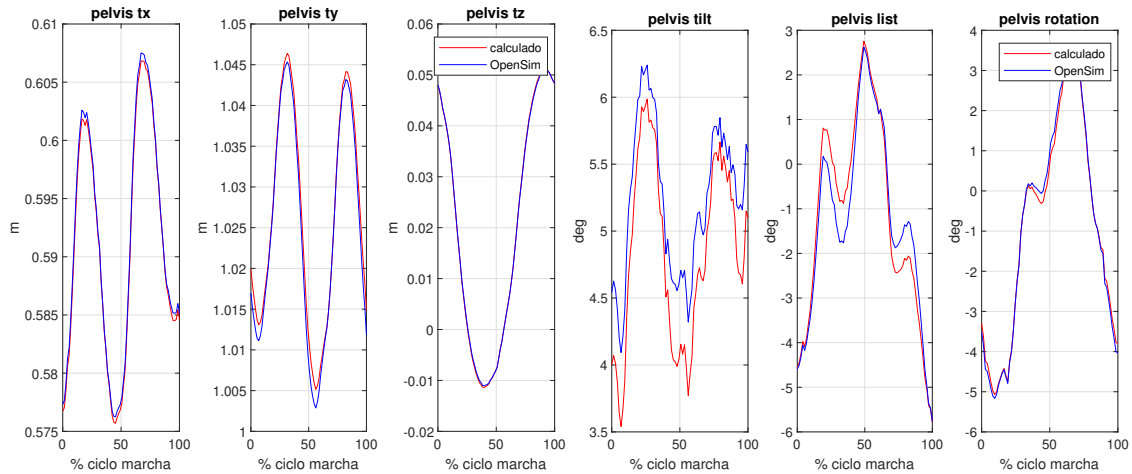


Figura 4.14 Resultados del modelo monosegmento sin splines para la pelvis.

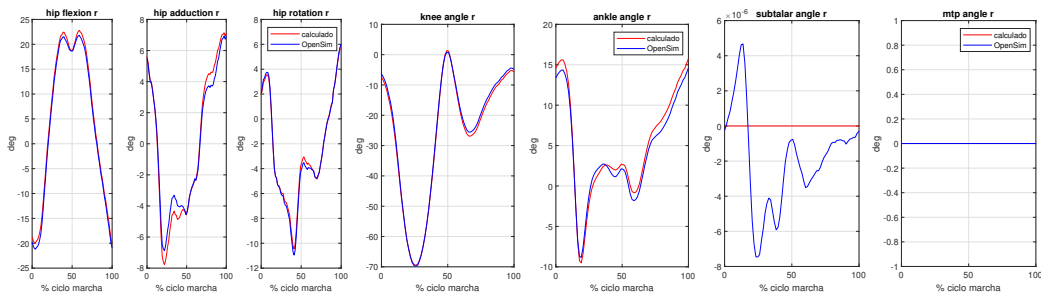


Figura 4.15 Resultados del modelo monosegmento sin splines para la pierna derecha.

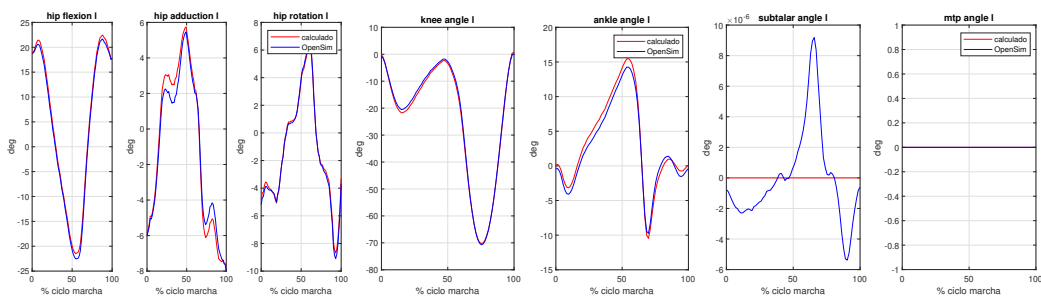


Figura 4.16 Resultados del modelo monosegmento sin splines para la pierna izquierda.

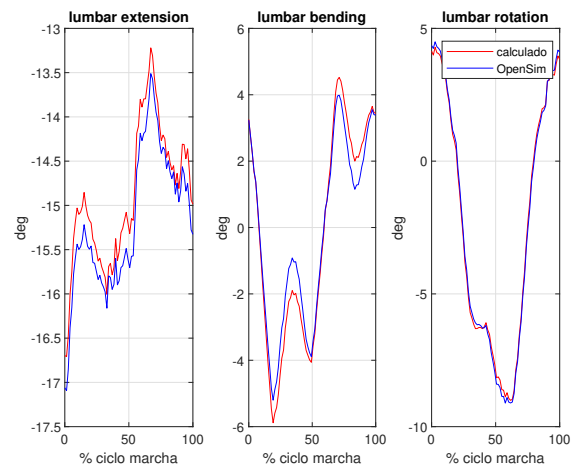


Figura 4.17 Resultados modelo monosegmento sin splines.

Tabla 4.5 Comparación de métodos IK.

Modelo	Error (cm)	Tiempo de ejecución (s)
OpenSim	0.7797	2
Modelo original Delp	0.7795	73
Modelo original con bloqueo	0.7795	59
Modelo sin splines	0.7847	19
Modelo monosegmento sin splines	0.7847	19

Tabla 4.6 Error asociado a cada marcador.

Marcador	Modelo original Delp	Modelo original con bloqueo	Modelo sin splines	Modelo sin splines monosegmento
Sternum	2.04	2.04	2.19	2.19
R.Acromium	7.75	7.75	7.95	7.95
L.Acromium	6.19	6.19	6.36	6.36
TopHead	3.26	3.26	3.23	3.23
R.ASIS	0.58	0.58	0.65	0.65
L.ASIS	0.72	0.72	0.77	0.77
V.Sacral	0.93	0.93	0.69	0.69
R.Thigh.Upper	1.49	1.49	1.47	1.47
R.Thigh.Front	1.80	1.80	1.85	1.85
R.Thigh.Rear	1.31	1.31	1.36	1.36
R.Shank.Upper	0.80	0.80	0.89	0.89
R.Shank.Front	0.95	0.95	1.14	1.14
R.Shank.Rear	0.63	0.63	0.64	0.64
R.Heel	0.46	0.46	0.56	0.56
R.Midfoot.Sup	0.71	0.71	0.58	0.58
R.Midfoot.Lat	0.67	0.67	0.75	0.75
R.Toe.Lat	0.93	0.93	0.95	0.95
R.Toe.Med	0.80	0.80	0.73	0.73
R.Toe.Tip	0.36	0.36	0.34	0.34
L.Thigh.Upper	1.74	1.75	1.73	1.73
L.Thigh.Front	2.18	2.18	2.16	2.16
L.Thigh.Rear	1.69	1.69	1.67	1.67
L.Shank.Upper	0.93	0.93	0.98	0.98
L.Shank.Front	0.94	0.94	0.98	0.98
L.Shank.Rear	0.78	0.78	0.79	0.79
L.Heel	0.58	0.58	0.62	0.62
L.Midfoot.Sup	1.09	1.09	0.99	0.99
L.Midfoot.Lat	0.62	0.62	0.69	0.69
L.Toe.Lat	0.72	0.72	0.78	0.78
L.Toe.Med	0.74	0.74	0.65	0.65
L.Toe.Tip	0.46	0.46	0.45	0.45

5 Dinámica Inversa

En el presente capítulo se explica cómo se ha resuelto el problema dinámico inverso. El objetivo de este problema es calcular el valor de las fuerzas y momentos en las articulaciones a partir del valor de las coordenadas generalizadas obtenidas en la resolución del problema cinemático inverso. En este capítulo se explica en primer lugar como se formula, para posteriormente describir como se ha implementado en Matlab y finalmente presentar los resultados obtenidos y su comparación con los de OpenSim.

5.1 Formulación

Para este módulo la formulación se reduce a resolver el conjunto de ecuaciones de Newton-Euler que se han descrito en el capítulo dedicado a los conceptos teóricos.

$$\mathbf{Q}_{gdl} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{Q}_v(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{Q}_{grav}(\mathbf{q}) \quad (5.1)$$

5.2 Modelado en Matlab

A continuación se presenta un extracto del código en el que se aprecia que la velocidad angular de la pelvis se calcula siguiendo la formulación del profesor Escalona, mientras que las velocidades angulares de espalda y cadera se definen directamente ya que su expresión es análoga a la de la pelvis.

Código 5.1 Código de generación de matrices simbólicas.

```
%% Derivadas de las matrices de rotacion
dA_pelvis = deriva_matriz_optimizado(A_pelvis,dq);
w_pelvis_sk = simplify(A_pelvis' * dA_pelvis);
w_pelvis = anti_skew (w_pelvis_sk);

% aqui empieza lo interesante
w_back = [dq18*cos(q19) - dq17*cos(q18)*sin(q19);
          dq19 + dq17*sin(q18);
          dq18*sin(q19) + dq17*cos(q18)*cos(q19)]; % misma expresion que
          w_pelvis pero con q17,q18,q19
```

```
w_hip_r = [dq8*cos(q9) - dq7*cos(q8)*sin(q9);
           dq9 + dq7*sin(q8);
           dq8*sin(q9) + dq7*cos(q8)*cos(q9)]; % misma expresion que w_pelvis
           pero con q7,q8,q9

w_hip_l = [dq13*cos(q14) - dq12*cos(q13)*sin(q14);
           dq14 + dq12*sin(q13);
           dq13*sin(q14) + dq12*cos(q13)*cos(q14)]; % misma expresion que
           w_pelvis pero con q12,q13,q14
```

El proceso de creación de las matrices dinámicas es el siguiente: en primer lugar uso de calculo simbólico para obtener las expresiones de todas las matrices necesarias. En ese mismo código, uso de la función `matlabFunction` para convertir todas las expresiones en funciones de Matlab. Posteriormente y en otro código, llamada a dichas funciones para resolver el problema de forma numérica.

Código 5.2

Codigo de generacion de matrices simbolicas.

```
q = sym('q',[19 1],'real') ;
dq = sym('dq',[19 1],'real');
syms q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11 q12 q13 q14 q15 q16 q17 q18 q19
    real
syms dq1 dq2 dq3 dq4 dq5 dq6 dq7 dq8 dq9 dq10 dq11 dq12 dq13 dq14 dq15
    dq16 dq17 dq18 dq19 real

G_pelvis = jacobian(w_pelvis,dq);

matlabFunction(G_pelvis,'File','G_pelvis_maza','Vars',{q,dq})
```

Código 5.3

Uso de matrices simbolicas.

```
clear all
close all
clc
load('qopt_sin_spline')

for i= 1:119
    G_pelvis_modelo = G_pelvis_maza (q(:,i),dq(:,i));
    T(:,i) = S\Q_accion;
end
```

Conviene aclarar que las coordenadas generalizadas se han filtrado utilizando un filtro pasobajo IIR tomado de OpenSim y traducido a Matlab. La frecuencia de corte escogida han sido 6 Hz, como hace OpenSim de manera predeterminada. Sin este filtro las fuerzas generalizadas tienen mucho ruido ya que son proporcionales a la derivada segunda de las coordenadas generalizadas.

El tiempo de ejecución del código que escribe las funciones es bastante elevado, debido a que se hace uso de calculo simbólico. Una de las tareas mas costosas computacionalmente es cuando se ejecuta `matlabFunction`. En los casos de matrices con una expresión mas compleja es necesario

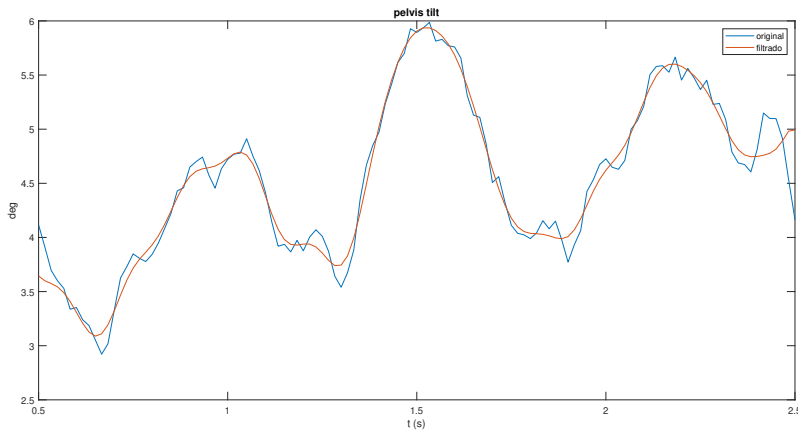


Figura 5.1 Filtrado.

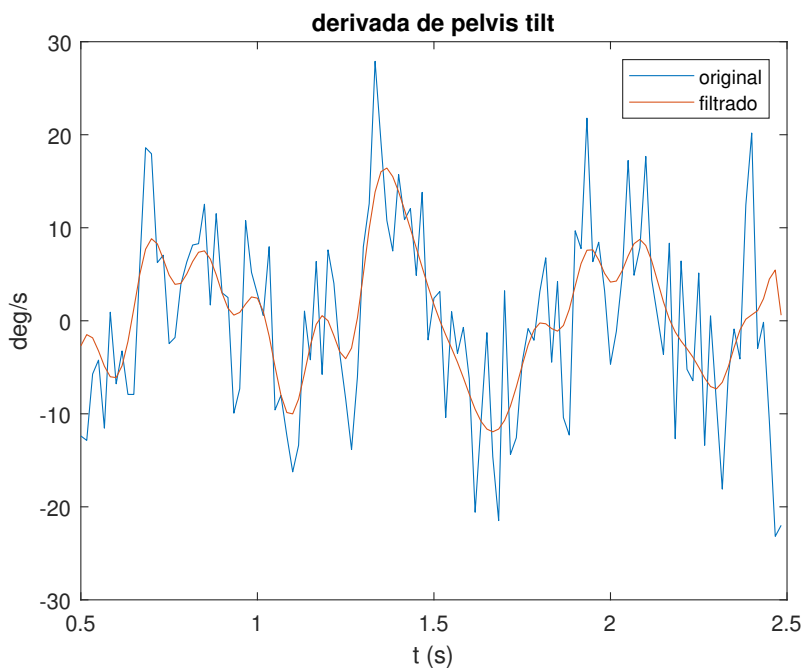


Figura 5.2 Filtrado.

desactivar la optimización que Matlab hace por defecto al crear la función. Esto se lleva a cabo utilizando la opción 'Optimize', false. Por su parte, el tiempo de ejecución del código que evalúa numéricamente las matrices y obtiene el valor de los momentos para cada instante de tiempo es de unos 10 segundos.

5.3 Presentación de resultados

El valor de las coordenadas generalizadas utilizado como dato de partida ha sido el obtenido con el modelo monosegmento sin splines. De la Figura 5.3 a la Figura 5.6 se presentan los resultados obtenidos y su comparación con los de OpenSim. Se aprecia que se obtienen resultados aproximados a los de OpenSim excepto para los momentos en la pelvis y el momento de flexión del tobillo. En ambos casos se desconoce el motivo por el que no se obtienen unos buenos resultados.

En la Tabla 5.1 se presenta el error cuadrático medio cometido para cada fuerza generalizada.

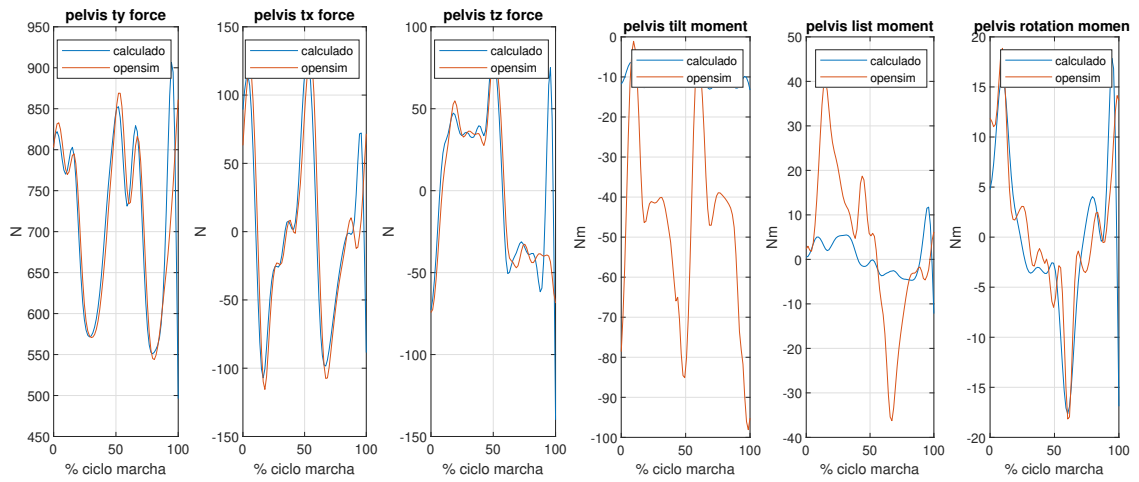


Figura 5.3 Resultados dinámicos para la pelvis.

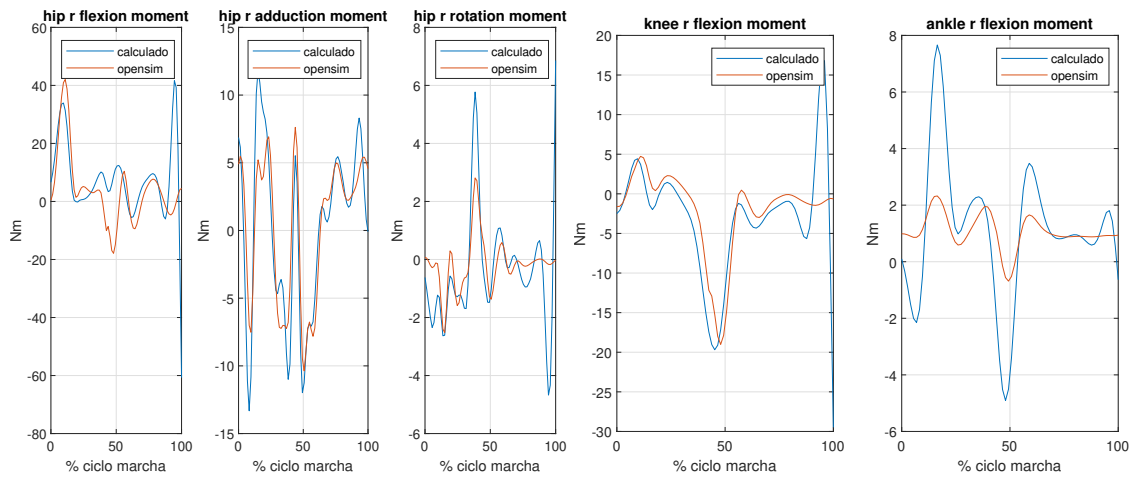


Figura 5.4 Resultados dinámicos para la pierna derecha.

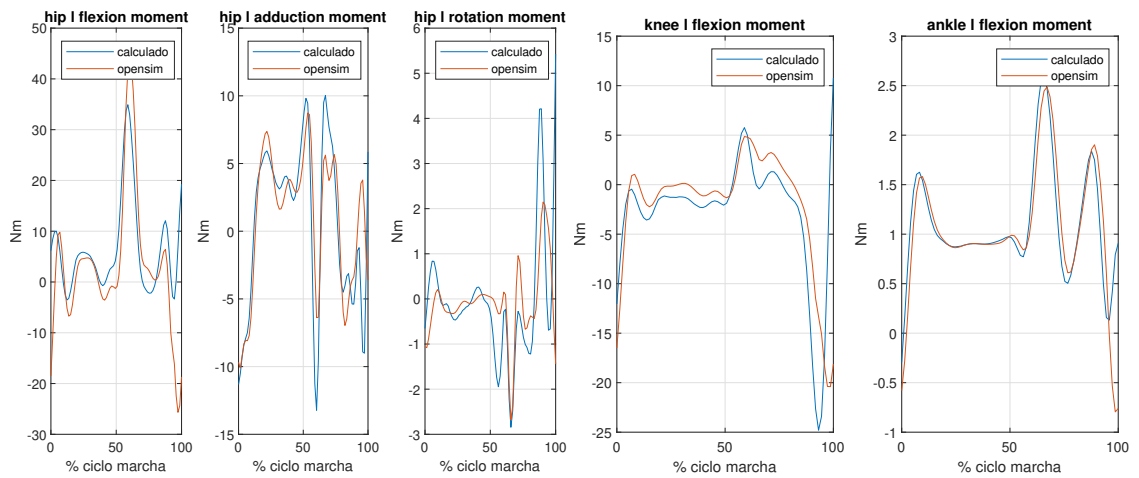


Figura 5.5 Resultados dinámicos para la pierna izquierda.

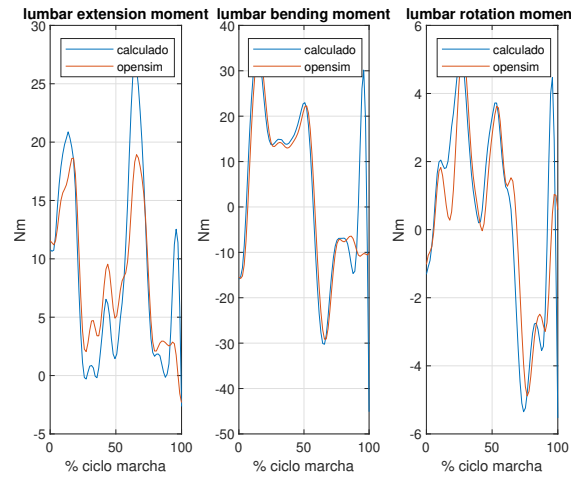


Figura 5.6 Resultados dinámicos para la espalda.

Tabla 5.1 Error asociado a cada fuerza generalizada.

Fuerza generalizada	Error (N o Nm)
pelvis_ty_force	63.38
pelvis_tx_force	31.33
pelvis_tz_force	26.82
pelvis_tilt_moment	42.11
pelvis_list_moment	15.97
pelvis_rotation_moment	5.55
hip_r_flexion_moment	14.47
hip_r_adduction_moment	3.08
hip_r_rotation_moment	1.52
knee_r_flexion_moment	5.59
ankle_r_flexion_moment	1.96
hip_l_flexion_moment	10.29
hip_l_adduction_moment	3.46
hip_l_rotation_moment	1.28
knee_l_flexion_moment	5.55
ankle_l_flexion_moment	0.36
lumbar_extension_moment	4.06
lumbar_bending_moment	9.44
lumbar_rotation_moment	1.55

6 Conclusiones

Las conclusiones generales que se pueden extraer de la realización de este trabajo es que se obtienen unos resultados muy similares a los obtenidos por OpenSim pero con un tiempo de ejecución mucho mayor. Esto se debe a que OpenSim esta programado en C++, un lenguaje de programación compilado. Matlab mas que un lenguaje de programación es una herramienta de calculo, muy potente, pero que no se puede enfrentar a otros lenguajes de programación. También hay que tener en cuenta que el programa de OpenSim ha sido desarrollado por un equipo con ingenieros de software, lo que da una idea de la importancia que tiene un código bien escrito y optimizado para las simulaciones cinemáticas y dinámicas.

En cuanto a los diferentes modelos planteados, se considera que el modelo monosegmento sin splines es adecuado para el objetivo propuesto de emular a OpenSim ya que con el se obtienen valores muy similares a los del programa y además permite llevar a cabo el módulo de dinámica inversa en Matlab. En cuanto a los resultados obtenidos en este último módulo para los momentos en la pelvis y el momento de flexión del tobillo no se obtienen valores similares a los de OpenSim por alguna razón desconocida.

Una posible mejora de este trabajo pasaría por obtener los residuales dinámicos para evaluar el error cometido en los resultados dinámicos en lugar del error cuadrático medio. Otra pasaría por abarcar no solo el modelo gait2392, si no muchos más y que al comienzo del programa se le preguntara al usuario que modelo quiere usar y con qué opciones. Esto permitiría que usuarios sin muchos conocimientos de programación pudiesen usar el programa y a la vez, al estar programado en Matlab, un lenguaje mas sencillo que C++, que usuarios con mayores conocimientos de programación pudiesen modificar el código.

Otra posible mejora de este trabajo pasaría por hacer uso de Python, un lenguaje de programación orientado a objetos que dispone de librerías muy interesantes de álgebra matricial.

Los principales retos a los que se ha tenido que hacer frente en el desarrollo de este trabajo han sido en primer lugar la necesaria comprensión del modelo gait2392 para programar su cinemática. Posteriormente, en el modulo de dinámica inversa Matlab se quedaba sin memoria al hallar las matrices dinámicas utilizando los procedimientos descritos en el documento del profesor JL Escalona. Este problema quedo solventado al calcular las velocidades angulares de manera directa como se explica en el capitulo dedicado a la dinámica inversa.

La continuación natural de este trabajo pasa en primer lugar por la programación en Matlab del modulo de Optimización estática. Este modulo consiste en, a partir de los valores de los momentos

en las articulaciones obtener el valor de las fuerzas producidas por cada musculo en cada instante de tiempo. Estas fuerzas se obtienen de una minimización de la suma de las activaciones de los músculos al cuadrado. El hecho de que sea necesaria una optimización se debe a que por cada momento existe mas de un musculo que contribuye a realizarlo, por lo tanto es necesario un criterio para de alguna manera repartir que musculo hace mas o menos fuerza. En este caso el criterio es la minimización de las activaciones musculares, lo cual es una manera de minimizar la potencia muscular necesaria para realizar el movimiento.

El otro problema de este modulo se produce al derivar numéricamente el valor de las coordenadas generalizadas para obtener velocidades y aceleraciones, ya que, como es conocido, la derivación hace se produzca ruido en las señales de velocidad y aceleración.

Bibliografía

- [1] *fmincon doc page*, <https://es.mathworks.com/help/optim/ug/fmincon.html>.
- [2] *Gait 2392 and 2354 models*, <https://simtk-confluence.stanford.edu/display/OpenSim/Gait+2392+y+2354+Models>.
- [3] B. Bresler and J.P. Frankel, *The forces and moments in the leg during level walking*, Transactions of the American Society of Mechanical Engineers **72** (1950), 27–36.
- [4] M. Damsgaard, J. Rasmussen, S.T. Christensen, E. Surma, and M. de Zee, *Analysis of musculoskeletal systems in the AnyBody Modeling System*, Simulation Modelling Practice and Theory **14** (2006), no. 8, 1100–1111.
- [5] S.L. Delp, F.C. Anderson, A.S. Arnold, P. Loan, A. Habib, C.T. John, E. Guendelman, and D.G. Thelen, *Opensim: Open-source software to create and analyze dynamic simulations of movement*, IEEE Transactions on Biomedical Engineering **54** (2007), no. 11, 1940–1950.
- [6] S.L. Delp, J.P. Loan, M.G. Hoy, F.E. Zajac, E.L. Topp, and J.M. Rosen, *An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures*, IEEE Transactions on Biomedical Engineering **37** (1990), no. 8, 757–767.
- [7] J.L. Escalona, *Dinámica inversa de una bicicleta a partir de datos experimentales*, Apuntes de clase.
- [8] J.R. Gage, M.H. Schwartz, and S.E. Koop, *Identification and treatment of gait problems in cerebral palsy*, ch. 2.6, 2009.
- [9] C.T. John, A. Seth, M.H. Schwartz, and S.L. Delp, *Contributions of muscles to mediolateral ground reaction force over a range of walking speeds*, Journal of Biomechanics **45** (2012), no. 14, 2438–2443.
- [10] M.Q. Liu, F.C. Anderson, M.H. Schwartz, and S.L. Delp, *Muscle contributions to support and progression over a range of walking speeds*, Journal of Biomechanics **41** (2008), no. 15, 3243–3252.
- [11] J. Ojeda, *Application of multibody system techniques to human locomotor system*, Ph.D. thesis, Escuela Superior de Ingenieros Industriales of the University of Seville, 2012.
- [12] J. Ojeda, J. Martínez-Reina, and J. Mayo, *A method to evaluate human skeletal models using marker residuals and global optimization*, Mechanism and Machine Theory **73** (2014), 259–272, cited By 10.

-
- [13] M.G. Pandy and N. Berme, *A numerical method for simulating the dynamics of human walking*, Journal of Biomechanics **21** (1988), no. 12, 1043–1051.
- [14] M.R. Pitkin, *Biomechanical investigations of sound and prosthetic gait*, ch. 2, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [15] R. Riemer, E.T. Hsiao-Weckler, and X. Zhang, *Uncertainties in inverse dynamics solutions: A comprehensive analysis and an application to gait*, Gait & Posture **27** (2008), no. 4, 578–588.
- [16] M.H. Schwartz, A. Rozumalski, and J.P. Trost, *The effect of walking speed on the gait of typically developing children*, Journal of Biomechanics **41** (2008), no. 8, 1639–1650.
- [17] A. Seireg and R.J. Arvikar, *A mathematical model for evaluation of forces in lower extremities of the musculo-skeletal system*, Journal of Biomechanics **6** (1973), no. 3, 313–322, IN19–IN20, 323–326.
- [18] A. Seth, M. Sherman, P. Eastman, and S. Delp, *Minimal formulation of joint motion for biomechanisms*, Nonlinear Dynamics **62** (2010), no. 1-2, 291–303.
- [19] M.A. Sherman, A. Seth, and S.L. Delp, *Simbody: Multibody dynamics for biomedical research*, vol. 2, 2011, pp. 241–261.
- [20] D.A. Winter, *Overall principle of lower limb support during stance phase of gait*, Journal of Biomechanics **13** (1980), no. 11, 923–927.