



UNIVERSIDAD
DE PIURA

REPOSITORIO INSTITUCIONAL
PIRHUA

VALORACIÓN ENERGÉTICA EN LA MARCHA HUMANA UTILIZANDO UNA ÓRTESIS ACTIVA, BAJO EL ANÁLISIS DINÁMICO DE SISTEMAS MULTICUERPO

Arnold Quispe-García

Lima, diciembre de 2015

FACULTAD DE INGENIERÍA

Área Departamental de Ingeniería Industrial y de Sistemas

Quispe, A. (2015). *Valoración energética en la marcha humana utilizando una órtesis activa, bajo el análisis dinámico de sistema multicuerpo*. Tesis de pregrado no publicado en Ingeniería Industrial y de Sistemas. Universidad de Piura. Facultad de Ingeniería. Programa Académico de Ingeniería Industrial y de Sistemas. Lima, Perú.

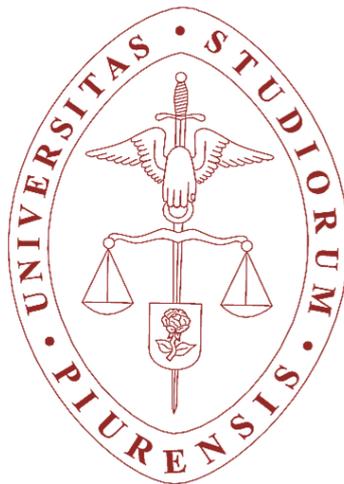


Esta obra está bajo [una licencia Creative Commons Atribución-NoComercial-SinDerivadas 2.5 Perú](#)

Repositorio institucional PIRHUA – Universidad de Piura

UNIVERSIDAD DE PIURA

FACULTAD DE INGENIERÍA



**“VALORACIÓN ENERGÉTICA EN LA MARCHA HUMANA UTILIZANDO
UNA ÓRTESIS ACTIVA, BAJO EL ANÁLISIS DINÁMICO DE SISTEMAS
MULTICUERPO”**

Tesis para optar el Título de
Ingeniero Industrial y de Sistemas

Arnold Eloy Quispe García

Asesor: Ing. Eliodoro Carrera Chinga, PhD.

Lima, diciembre de 2015

A mis abuelos.

PRÓLOGO

Cuando se quiere mejorar algún proceso, es necesario conocer el problema, generar alternativas y escoger una de ellas con el mejor criterio establecido. En este sentido, el estudio de la Biomecánica permite determinar la mejor solución a través de criterios mecánicos aplicados al cuerpo humano. Y para esto es importante conocer las características de la marcha humana en estudio para definir tipos de rehabilitación. Cuando se detecta una patología en la marcha de un individuo, se plantean soluciones adecuadas, por ejemplo, intervención quirúrgica, tipos y horas de rehabilitación o diseñar algún aparato ortopédico para disminuir las características de dicha patología.

Las simulaciones del movimiento, el cual deseamos estudiar, son una herramienta valiosa para el análisis y diseño de sistemas multicuerpo de dicho movimiento. Esto se puede representar gráficamente de forma real para tener una ayuda visual del sistema en estudio.

El Laboratorio de Ingeniería Mecánica de la Universidade A Coruña, viene desarrollando investigaciones sobre diversos temas de simulación y realidad virtual, y uno de ellos es en temas biomecánicos. Esta Universidad junto con la Universidad Politécnica de Cataluña y Universidad de Extremadura, desarrollan un prototipo de órtesis activa para pacientes con lesiones medulares incompletas.

El fin de la presente tesis es determinar la energía consumida cuando el paciente utiliza este tipo de órtesis, y realizar un análisis cualitativo del movimiento, usando la simulación de la marcha del paciente. La herramienta principal en este estudio será el software Matlab que permitirá el desarrollo de problemas cinemáticos y dinámicos, así como la simulación mencionada.

No quiero terminar sin agradecer especialmente a mi asesor, Ing. Eliodoro Carrera, PhD. por su orientación en el presente trabajo. Al director del LIM, Prof. Ing. Javier Cuadrado, PhD. por su confianza, paciencia y enseñanzas impartidas en mi estadía en UDC, campus de Esteiro, y a todos los amigos e investigadores del LIM: Urbano, Miguel Ángel, Alberto, Emilio, Amelia, David, por compartir sus experiencias, consejos y amistad.

RESUMEN

La presente tesis muestra el modelamiento y simulación de la marcha humana, en 2D; utilizando una órtesis activa en el paciente de estudio, se realiza el cálculo energético en la marcha del sujeto, mostrando la ventaja del uso de este mecanismo para realizar una marcha normal.

En la primera parte de este trabajo se presenta el estudio de la marcha humana bajo conceptos biomecánicos. Además se introduce el modelamiento de sistemas mecánicos, coordenadas naturales y se desarrolla el estudio de mecanismos comunes: mecanismo de cuatro barras, con sus variaciones, y el manipulador plano. Con el desarrollo de estos modelamientos, se muestran los resultados en Matlab.

En la segunda parte se desarrolla la cinemática y dinámicas de los segmentos corporales, modelizados como sólidos rígidos. Se aplica cinemática inversa para determinar los ángulos entre los segmentos, luego continúa con el cálculo de velocidades y aceleraciones por diferenciación numérica. Posteriormente, se realiza la dinámica inversa, para el cálculo de momentos en las articulaciones.

En la parte final, se introduce el concepto de órtesis activa. Para el nuevo modelo de estudio, se introduce la órtesis y las muletas al modelo, posteriormente se realiza los cálculos necesarios. Y finalmente, calcular el gasto energético, el cual permite el análisis cualitativo de la marcha de una persona con lesión medular.

ÍNDICE GENERAL

INTRODUCCIÓN.....	1
Capítulo 1.....	2
PLANTEAMIENTO DEL PROBLEMA.....	2
1.1 DESCRIPCIÓN DE LA PROBLEMÁTICA	2
1.2 FORMULACIÓN DEL PROBLEMA	2
1.3 OBJETIVOS DE LA INVESTIGACIÓN.....	3
1.4 JUSTIFICACIÓN.....	3
Capítulo 2.....	4
MARCO TEÓRICO	4
2.1 CONCEPTOS BÁSICOS.....	4
2.1.1 Elementos matemáticos	4
2.1.2 Conceptos Físicos	6
2.1.3 Biomecánica	7
2.2 ESTUDIO DE LA MARCHA HUMANA.....	9
2.2.1 Historia	9
2.2.2 Definiciones	10
2.2.3 Estudio y Características de la marcha	13
2.3 MODELIZACION DE SISTEMAS MECÁNICOS	17
2.3.1 Coordenadas naturales	17
2.3.2 Estudio de mecanismos comunes	21
2.3.3 Four bar linkage	21
2.3.4 Manipulador plano.....	28
Capítulo 3.....	34
CINEMÁTICA Y DINÁMICA ORIENTADA A SEGMENTOS CORPORALES	34
3.1 MODELIZACIÓN	34
3.2 CÁLCULO DE LA ORIENTACIÓN DE CADA SEGMENTO.....	38
3.3 CÁLCULO DE LOS ÁNGULOS: CINEMÁTICA INVERSA	48
3.4 VELOCIDADES Y ACELERACIONES: DIFERENCIACIÓN NUMÉRICA ...	50
3.5 CÁLCULO DE REACCIONES Y ESFUERZOS MOTORES: DINÁMICA INVERSA	54
3.6 PROTOTIPO ÓRTESIS ACTIVA.....	57

3.6.1 Características.....	58
3.6.2 Funcionamiento	59
3.6.3 Introducción de las órtesis y muletas en el modelo	60
3.7 EFICIENCIA ENERGÉTICA EN LA MARCHA	62
Capítulo 4.....	64
EXPERIMENTACIÓN Y RESULTADOS	64
4.1 CAPTURA DE DATOS	64
4.1.1 Pre proceso de datos	64
4.1.2 Proceso de capturas	66
4.2 ANÁLISIS DE RESULTADOS	70
CONCLUSIONES Y RECOMENDACIONES.....	80
BIBLIOGRAFÍA.....	81
ANEXOS.....	82
A.1 Características de la sala de captura y el equipo	82
A.2 Presupuesto para equipar una sala de capturas	86
A.3 Presupuesto para la elaboración de un prototipo de órtesis activa.....	86
A.4 Técnicas de Captura y Medición	87
A.5 Programación de problemas cinemáticos y dinámicos en MATLAB	91

INTRODUCCIÓN

Panorama actual de la Biomecánica en el Perú

A medida que la tecnología se desarrolla, los métodos de rehabilitación de personas inválidas son cada vez mejores y sofisticados. Así como el tiempo de vida de las personas aumenta, el número de personas de mayor edad crece y un mayor enfoque en los asuntos de salud incrementará la demanda de mejores sistemas y equipos médicos diseñados para el servicio a la población.

Por lo tanto, se ha pronosticado que en el ámbito de la Biomedicina y Biomecánica se necesitarán ingenieros especializados para solucionar diferentes problemáticas en estos aspectos.

En Estados Unidos el programa de Ingeniería Biomecánica es la carrera con más perspectivas de crecimiento: 72%. 2008-2018 (Bureau of Labor Statistics, 2008). Con este desarrollo, tendrá un impacto positivo local, se necesitarán especialistas en el manejo y desarrollo de equipo médico, recursos tecnológicos en hospitales y capaces de trabajar en forma multidisciplinaria con médicos.

El Instituto Nacional de Rehabilitación del Perú cuenta con un departamento de Biomecánica, el cual brinda la atención en la elaboración de prótesis, órtesis y calzados ortopédicos; sin embargo, esto no va más allá en la innovación de los aparatos antes mencionados.

En la Clínica San Juan de Dios, hay un Laboratorio de la Marcha, en la cual se brinda un servicio de análisis de la marcha para niños y adultos con diferentes deficiencias. Al finalizar la captura y la valoración de datos en este laboratorio, se entrega los resultados al médico quien indicará la mejor forma de afrontar la lesión o rehabilitación del paciente.

Capítulo 1

PLANTEAMIENTO DEL PROBLEMA

La tesis está enfocada en el campo de la biomecánica, específicamente en la rama de la medicina humana, y trata sobre la problemática de asistir a lesionados medulares, tipo incompleta, con características particulares: cierta sensibilidad en segmentos corporales que no puede mover, mover un miembro más que el otro o tener más funciones en un lado del cuerpo. Un ejemplo de este tipo de lesión, es una persona que no tenga actividad motora independiente en el tobillo y rodilla, pero sí el de la cadera.

1.1 DESCRIPCIÓN DE LA PROBLEMÁTICA

Estos pacientes al carecer de la sincronización y fuerza para poder realizar una marcha normal, pueden exagerar en sus movimientos, excediendo el consumo de energía al caminar, incluso utilizando aparatos y/o ayuda ortopédica. De esta forma, el paciente enfrenta limitaciones para poder realizar actividades rutinarias.

La solución más común y económica es una órtesis pasiva que permite el bloqueo mecánico de la rodilla, es decir, no permita la flexión de la rodilla, consiguiendo estabilidad en el paciente cuando este se apoya en sus muletas. Con este tipo de órtesis se puede lograr que el lesionado consiga un movimiento parecido a la marcha humana pero con un consumo de energía excesivo, puesto que tendría que exagerar sus movimientos para el desplazamiento.

La innovación de este tipo de órtesis, es una órtesis de tipo activa, la cual permite la flexión de la rodilla al detectar el contacto del talón en la pierna contraria, en consecuencia, mejora la locomoción del paciente disminuyendo la energía en la marcha.

La mejora introducida por una órtesis activa es que este aparato posee un control que está configurado para la sincronización de la misma, es decir, maneja el bloqueo de la órtesis y el ángulo con el que la rodilla realiza la flexión en la marcha.

1.2 FORMULACIÓN DEL PROBLEMA

De acuerdo con lo expuesto anteriormente, una pregunta fundamental es la siguiente:

“¿El uso de una órtesis activa beneficia en el proceso de la marcha de un paciente con lesión medular de tipo incompleta?, ¿La evaluación y medida de la eficiencia energética en la marcha es una herramienta eficiente para determinarlo?”

Es importante el cálculo energético con la finalidad de demostrar que la reducción en el consumo de energía en el proceso de la marcha del paciente se consigue con una órtesis activa. Además de comprobar que la aplicación de una estrategia de control, ya establecida por los desarrolladores de este tipo de órtesis, ayuda para la marcha del paciente en estudio.

1.3 OBJETIVOS DE LA INVESTIGACIÓN

Objetivo principal de este proyecto:

- Cuantificar la energía mecánica consumida en la marcha de un paciente con lesión medular cuando utiliza un prototipo de órtesis activa.

El uso de diferentes tipos de órtesis para tratar de corregir movimientos erráticos, en este tipo de pacientes, se está convirtiendo en un reto para investigadores y empresas que apuestan por brindar un mejor producto en la rehabilitación y poder disminuir el consumo energético en la marcha de dichos pacientes.

Objetivos secundarios:

- Comprobar que la adición de un control de la órtesis mejora la marcha que se analiza.
- Realizar el modelamiento y una simulación de la marcha la cual evidencia las características de una marcha deficiente, desde el punto de vista energético.
- Simular en MATLAB un modelo 2D, proyección en el plano sagital buscando simplificar variables y carga computacional, de la marcha de una persona sana y un paciente utilizando una órtesis activa, con diferentes formulaciones dinámicas y analizar los datos que provienen de las capturas de movimientos.
- Motivar el planteamiento de mejores estrategias de control, para posteriores investigaciones, que optimice la energía en la marcha.

1.4 JUSTIFICACIÓN

“La Aplicación de técnicas de dinámica multicuerpo al diseño de órtesis activas para ayuda a la marcha” es un proyecto que se enfoca en el beneficio del uso de un prototipo diseñado y elaborado por los departamentos de Ingeniería Mecánica de la Universidad de La Coruña, Universidad de Extremadura y la Universidad Politécnica de Cataluña. Se realiza una simulación en 3D del control de la órtesis para realizar un estudio de la marcha sin participación constante del paciente, ya que esto permite que el paciente no realice las pruebas de órtesis, una y otra vez, generando molestia en él. Para lo cual se diseña una órtesis activa que con ayuda de actuadores ubicados en la rodilla y tobillo ayude a mejorar la marcha de lesionados medulares, utilizando como herramienta la simulación de la dinámica del movimiento humano. Esto se puede comprobar cuando el paciente realiza un movimiento más natural y sin un coste de energía excesivo.

Sin embargo, no se verifica si esta estrategia de control es capaz de minimizar la energía, de acuerdo a las características de la lesión del paciente y su marcha. A consecuencia de esto, en este proyecto de tesis se realizará un análisis energético en 2D que comprueba los beneficios de la adición de control en una órtesis activa. No está dentro del alcance de este trabajo seleccionar la estrategia de control que optimiza la marcha.

Capítulo 2

MARCO TEÓRICO

2.1 CONCEPTOS BÁSICOS

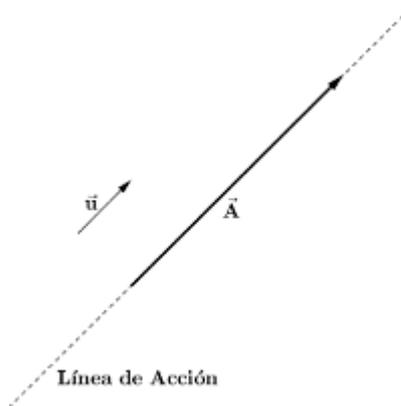
2.1.1 Elementos matemáticos

2.1.1.1 Escalar

Cantidad física que puede ser describe con un número, decimos que es una **cantidad escalar** (Sears, F. W., Zemansky, M. W., y Young, H. D., 2009). Ejemplos: tiempo, temperatura, densidad, masa, volumen.

2.1.1.2 Vector

Una cantidad vectorial tiene tanto una magnitud (el “qué tanto”) como una dirección en el espacio (Sears, F. W., Zemansky, M. W., y Young, H. D., 2009). Ejemplos: posición, velocidad, aceleración, peso, aceleración, etc. Un vector posee módulo y dirección.



Se tiene que:

$$\vec{A} = n\vec{u}; \text{ donde } n \in \mathbb{R}$$

\vec{u} : vector unitario

$$\vec{A} \times \vec{u} = 0$$

Nota: un vector “físico” es una magnitud física cuya característica, con respecto al origen, es el punto de aplicación de esa magnitud física, pero que también se puede definir por las componentes independientes de acuerdo al sistema de coordenadas que el observador puede tomar arbitrariamente.

2.1.1.3 Función Vector y Jacobiano

Para definir el término Jacobiano, es necesario definir qué es un Vector de Funciones, y éste a su vez tendrá diferencias con una Función de un vector. La importancia de la definición del Jacobiano se debe a que es usado como una herramienta para realizar aproximaciones lineales de una función a un determinado punto.

2.1.1.4 Función de un vector

Es aquella función matemática que se expresa en términos de al menos uno de los componentes de dicho vector (Callacná, 2005).

$$f(\vec{x}) = f(x_1, x_2, \dots, x_n)$$

Ejemplos:

$$f_1(\vec{x}) = x_1^2 + x_2^2$$

$$f_2(\vec{x}) = x_3x_1 + x_2x_1$$

$$f_3(\vec{x}) = \cos(x_1^2x_3)$$

2.1.1.5 Vector de Funciones

Es un arreglo matemático que tiene como componentes a funciones de vectores.

Dado el vector (Callacná, 2005, pág. 4):

$$f(\vec{x}) = f(x_1, x_2, \dots, x_n)$$

Tenemos el siguiente vector de funciones:

$$\vec{f}(\vec{x}) = \begin{bmatrix} f_1(\vec{x}) \\ f_2(\vec{x}) \\ \vdots \\ f_m(\vec{x}) \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_m(x_1, x_2, \dots, x_n) \end{bmatrix}$$

Ejemplo:

$$\vec{f}(\vec{x}) = \begin{bmatrix} f_1(\vec{x}) \\ f_2(\vec{x}) \\ f_3(\vec{x}) \end{bmatrix} = \begin{bmatrix} x_1^2 + x_2^2 \\ x_3x_1 + x_2x_1 \\ \cos(x_1^2x_3) \end{bmatrix}$$

2.1.1.6 Derivada de un vector de funciones (Callacná, 2005, pág. 6)

$$\frac{\partial \vec{f}}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial \vec{x}} \\ \frac{\partial f_2}{\partial \vec{x}} \\ \vdots \\ \frac{\partial f_m}{\partial \vec{x}} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}}_{\text{Jacobiano } \mathbf{J}}$$

Ejemplo:

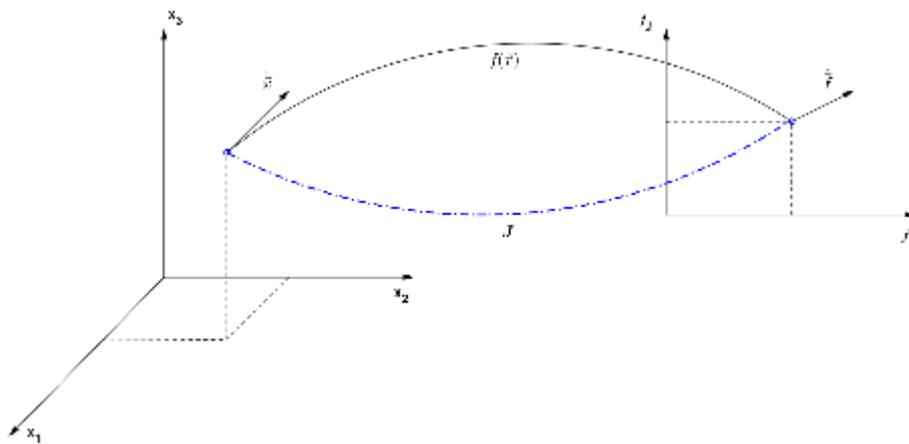
$$\vec{f}(\vec{x}) = \begin{bmatrix} x_1^2 + x_2^2 \\ x_3x_1 + x_2x_1 \end{bmatrix}$$

$$J = \frac{\partial \vec{f}}{\partial \vec{x}} = \begin{bmatrix} 2x_1 & 2x_2 & 0 \\ x_3 + x_2 & x_1 & x_1 \end{bmatrix}$$

Interpretación geométrica:

Relación de velocidad

$$\dot{\vec{f}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} \dot{x}_1 + \frac{\partial f_1}{\partial x_2} \dot{x}_2 + \frac{\partial f_1}{\partial x_3} \dot{x}_3 \\ \frac{\partial f_2}{\partial x_1} \dot{x}_1 + \frac{\partial f_2}{\partial x_2} \dot{x}_2 + \frac{\partial f_2}{\partial x_3} \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} \rightarrow \dot{\vec{f}} = J \dot{\vec{x}}$$



2.1.2 Conceptos Físicos

2.1.2.1 Energía

La energía es la capacidad que tienen los cuerpos para realizar un trabajo. La unidad de la Energía es el Julio o Joule (J). La energía mecánica se divide en energía cinética (E_k) y energía potencial (U).

2.1.2.2 Energía cinética lineal (E_{kl})

Es la energía que tienen los cuerpos a causa del movimiento de éste. Se expresa de la siguiente manera:

$$E_{kl} = \frac{1}{2} m v^2$$

Donde:

m : Masa del cuerpo

v : Velocidad del cuerpo

2.1.2.3 Energía cinética rotacional (E_{kr})

La energía cinética de un cuerpo que gira, es análoga a la energía cinética lineal. Y es expresada en términos de momento de inercia y velocidad angular.

Donde:

I : Momento de inercia

ω : Velocidad angular

$$E_{Kr} = \frac{1}{2} I \omega^2$$

2.1.2.4 *Energía potencial (U)*

Energía que está en función de la posición de un determinado cuerpo. Este tipo de energía, a su vez, se divide en dos tipos:

Energía potencial gravitatoria (U_g): Energía relativa al campo gravitacional, posición del cuerpo y nivel de referencia establecido.

$$U_g = mgh \quad \text{Donde:}$$

m : Masa del cuerpo.

h : Altura del cuerpo con respecto a un nivel de referencia.

g : Aceleración de la gravedad.

Energía potencial elástica (U_e): Energía a consecuencia de la deformación de un objeto elástico.

$$U_e = \frac{1}{2} kx^2 \quad \text{Donde:}$$

k : Constante de elasticidad lineal del objeto elástico.

x : Deformación del objeto elástico.

2.1.2.5 *Torque*

También llamado momento de una fuerza. Es una cantidad vectorial que mide el efecto de rotación o tendencia a la rotación debido a una fuerza que actúa sobre un cuerpo, respecto a un centro de giro.

$$\tau = \vec{r} \times \vec{F} \quad \text{Donde:}$$

\vec{r} : Vector de posición de la fuerza con respecto al centro de giro.

\vec{F} : Fuerza aplicada en un punto.

2.1.3 *Biomecánica*

La Biomecánica es la ciencia que estudia las fuerzas internas y externas que afectan al movimiento y su incidencia sobre sistemas biológicos, para nuestro objetivo, el cuerpo humano. Y por métodos mecánicos se puede obtener información sobre el sistema

músculo-esquelético, permite realizar comparaciones cuantitativas y cualitativas de diferentes estilos de locomoción.

Esta ciencia tiene aplicaciones en los siguientes campos:

- Medicina y fisioterapia: prótesis de cadera o rodilla, órtesis, dispositivo de rehabilitación o de asistencia a un discapacitado.
- Deportes: mejora de técnicas deportivas para alcanzar un mejor rendimiento, diseño de calzado deportivo con propiedades de absorción de impactos.
- Aplicaciones ergonómicas: diseño de un adecuado puesto de trabajo, herramientas y manuales.

La marcha es un proceso mecánico controlado por un sistema biológico, es por eso que es adecuado el estudio del movimiento con términos mecánicos. Ya que la mecánica es una materia de gran amplitud, se mencionan términos importantes para el análisis de la marcha, especialmente el tiempo, masa, fuerza, centro de gravedad, momento de fuerza y movimiento lineal y angular (Whittle, 2007, pág. 31).

Tiempo

La unidad básica del tiempo en el Sistema Internacional (SI) es el *segundo (s)*. Esto servirá para realizar medidas entre eventos de la marcha humana y posiciones de segmentos. Además el tiempo está relacionado con sucesos periódicos, repeticiones, esto es la frecuencia, en Hertz (Hz). Siendo 1 Hz igual a un ciclo por segundo.

Masa

La Inercia es la propiedad inherente de cada cuerpo. La masa es la expresión cuantitativa de esta inercia. La unidad básica de medida de la masa es el *kilogramo (kg)*. Cada día experimentamos la presencia de la masa de diferentes cuerpos, incluso del nuestro, sin poder renunciar a ello. La masa y el peso son completamente distintos, pero con una relación muy fuerte. La masa, que se representa como un escalar, no depende sobre el efecto de la fuerza gravitatoria, mientras que el peso es una fuerza, por lo tanto se puede representar como un vector, relacionada con la gravedad.

Fuerza

Ningún cuerpo es capaz por sí solo de modificar su estado de reposo o de variar su velocidad, en definitiva es incapaz de producir una aceleración, para que esto suceda es necesario que otro cuerpo actúe sobre él. La unidad básica de la fuerza en el SI es el *newton (N)*. Para describir una fuerza es necesario establecer una magnitud, línea de acción y sentido.

Toda la ciencia detrás de la mecánica está basada en las tres leyes de la fuerza propuesta por Sir Isaac Newton (Whittle, 2007, pág. 32):

Primera Ley de Newton: Un cuerpo continúa en estado de reposo, o en movimiento uniforme en una línea recta, a menos que sea afectado por una fuerza externa.

Segunda Ley de Newton: Una fuerza externa causará que un cuerpo acelere en dirección de la fuerza. La aceleración (a) es igual a la magnitud de la fuerza (F) dividida por la masa (m) del cuerpo, como en la ecuación:
$$a = \frac{F}{m}$$

Tercera Ley de Newton: Para una acción existe una reacción, la cual es de igual magnitud y de sentido contrario.

Obviando el extraño comportamiento de las partículas atómicas y subatómicas, todo sistema físico obedece simultáneamente a las tres leyes de Newton.

Centro de Gravedad

La masa de cualquier objeto está distribuida por toda su dimensión, pero frecuentemente es conveniente imaginar que toda la masa está concentrada en un solo punto, el cual es llamado centro de masa o centro de gravedad. Para cuerpos regulares, por ejemplo, un cubo, es fácil determinar su centro de gravedad, porque es el centro geométrico; para cuerpos irregulares, como el cuerpo humano, es necesario realizar mediciones directas para determinar el CG. Para hallar el CG de un objeto compuesto por otros objetos menores, es posible calcular el CG de éstos últimos para encontrar el CG del objeto principal. Se establece que el CG del cuerpo humano se ubica en frente de la unión lumbosacra, esta posición es aproximada en una postura anatómica pero el CG se moverá de acuerdo al movimiento del cuerpo (Whittle, 2007, pág. 36).

Momento de fuerza

La medición e interpretación del momento de fuerza es esencial para un entendimiento completo de una marcha normal y una patológica. Un *par* es un momento el cual es producido por dos fuerzas de igual magnitud y paralelas pero actuando en sentido contrario. En el Sistema Internacional de Unidades, la unidad del Momento de fuerza es el *newton metro* o *newton-metro*, N m o N·m. Las fuerzas se cancelan a sí mismas y producen una rotación alrededor del punto intermedio entre estas fuerzas. Un ejemplo de esto es el par al producirse la flexión plantar o la extensión de la rodilla.

2.2 ESTUDIO DE LA MARCHA HUMANA

2.2.1 Historia

Desde la época de los griegos, hubo interés en analizar la locomoción de los seres vivos. Sin embargo este análisis era con tan solo la observación. Una de las primeras obras de este análisis fue “De Motu Animalium” de Aristóteles, en donde se describe la acción de los músculos cuando un animal realiza un movimiento de desplazamiento o al mover una

carga. El análisis del movimiento del ser humano también fue plasmado en este estudio. Es importante señalar que este tipo de análisis mecánico era necesario para Aristóteles porque en el estudio completo de esta obra, él pretendía dar una conclusión sobre la observación del movimiento de la parte física cuyo origen es algo que no se mueve, pero esto lo dejaremos fuera por ser ajeno al propósito de este estudio.

Tiempo después, Borelli, estudiante de Galileo, hizo un estudio sobre la relación del sistema muscular y esquelético, introduciendo el principio de la palanca, observando la variación de la longitud y volumen de los músculos en el traslación de una persona, así como también en animales. Este análisis fue realizado en su libro “De Motu Animalium”, el cual no es el mismo al trabajo realizado por Aristóteles.

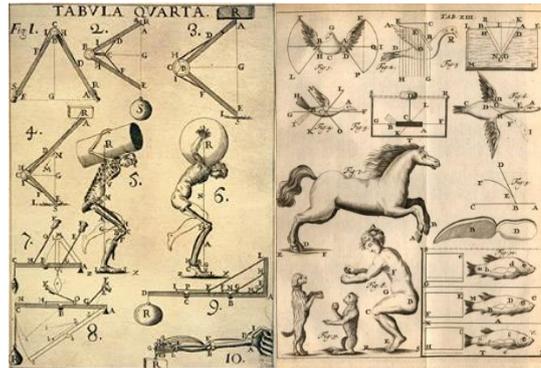


Figura 1 Dibujos “De Motu Animalium” por Borelli (1680)

Borelli se dio cuenta que es necesario determinadas fuerzas en las uniones de los segmentos del cuerpo humano para cumplir el equilibrio. También determinó la posición del centro de gravedad del cuerpo y como el balanceo es continuo en la marcha, mantenido por el constante movimiento delantero y soportado por el área de los pies.

2.2.2 Definiciones

La marcha humana se caracteriza por el la acción alternada de los miembros inferiores y particularmente por la sucesión de doble apoyo y apoyo unipodal. Entonces, cuando se realiza este movimiento, siempre habrá contacto con la superficie terrestre.

El inicio de todo el movimiento parte en el cerebro, específicamente en los centros supraespinales, con un impulso del sistema nervioso central que termina en la generación de fuerzas de reacción de la superficie de contacto en los pies. Y de acuerdo con la Tercera Ley de Newton, la acción producida por ese impulso nervioso, genera una reacción de la superficie con la que se mantiene el contacto.

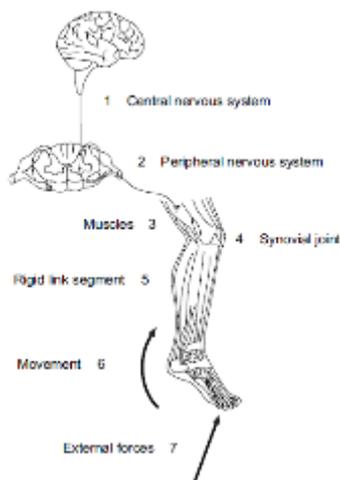


Figura 2 Los siete componentes que conforman la función básica de la manera en cómo caminamos. (Fuente: *Dynamics of Human Gait, Vaughan*)

Se establecen 7 secuencias que ocurren para la acción de la marcha (Vaughan, Davis, & O'Connor, 1999):

1. Registro y activación del comando marcha en el sistema nervioso central.
2. Transmisión de las señales de la marcha al sistema periférico central.
3. Contracción de los músculos que desarrollan la tensión
4. Generación de fuerzas y momentos en las uniones.
5. Regulación de las fuerzas y momentos en las uniones por los segmentos rígidos del esqueleto basado en su antropometría.
6. Desplazamiento de los segmentos: movimiento y desarrollo de la marcha.
7. Generación de las reacciones de la superficie a causa del apoyo del pie.

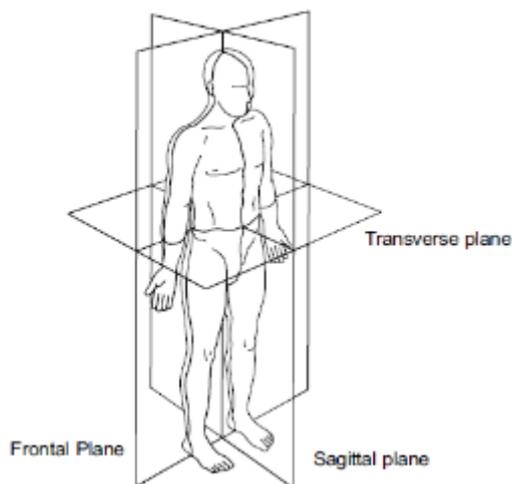


Figura 3 Representación de los diferentes planos del cuerpo humano. (Fuente: *Dynamics of Human Gait, Vaughan*)

El análisis de la marcha se puede realizar entorno a tres planos:

1. *Sagital*, este plano divide al cuerpo en parte izquierda y derecha.
2. *Frontal*, plano que divide en parte anterior y posterior a todo el cuerpo.
3. *Transversal*, divide al cuerpo en cuerpo en proporciones superiores e inferiores.

Dependiendo desde qué plano se efectúa este análisis se puede obtener diferentes datos, como por ejemplo la inclinación del ángulo de los segmentos. El sagital nos puede brindar mayor información ya que es aquí en donde se realiza el mayor de los movimientos y el análisis de este estudio es sobre este plano. Sin embargo, algunas patologías en la marcha se muestran, incluso, en los otros planos, por lo cual se sugiere tener cuidado en tomar poca importancia en ellos.

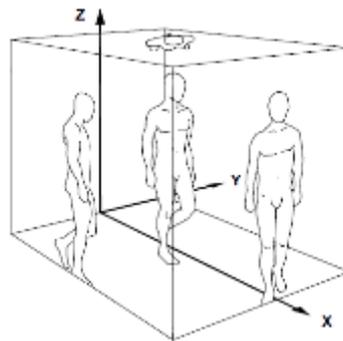


Figura 4 Proyección del movimiento en los tres planos (Fuente: *Dynamics of Human Gait, Vaughan*)

Luego de haber descrito los planos para el estudio de la marcha, se menciona las maneras en que los segmentos corporales, y uniones entre éstos, pueden desarrollar su movimiento a través de éstos planos.

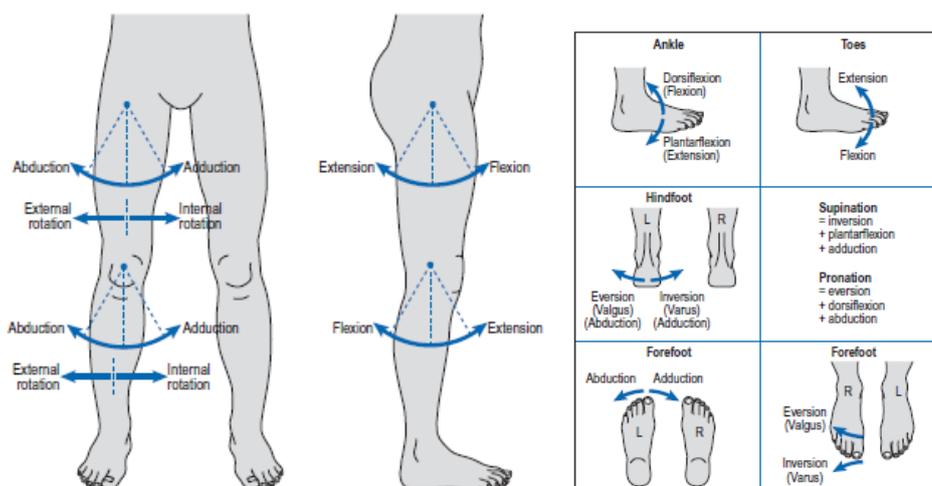


Figura 5 Movimiento alrededor de la cintura, rodilla y pie. (Fuente: *Gait Analysis, Whittle M.W.*)

La mayoría de las articulaciones solo se restringen al movimiento en uno o dos de esos tres planos señalados en la *Figura 3*. Los posibles movimientos que se muestra en la *Figura 5* son:

1. *Flexión y extensión* se realiza en el plano sagital; en el tobillo, estos movimientos son llamados *flexión dorsal* y *flexión plantar*, respectivamente.
2. *Abducción y Aducción* toman parte en el plano frontal.
3. *Rotación Interna y Rotación Externa* se realiza en el plano transversal

2.2.3 Estudio y Características de la marcha

Para efectos prácticos, la descripción de la marcha se divide en fases, de esta manera se puede analizar y realizar una descripción más ordenadas de los segmentos corporales con las características de cada fase.

Durante un ciclo de marcha completo, se considera dos fases:

1. Fase de apoyo (Stance phase): cuando el pie de referencia está en contacto con el suelo. Representa el 60% de un ciclo completo de marcha.
2. Fase de oscilación (Swing phase): pie de referencia realiza la oscilación en el aire. Representa el 40% de un ciclo completo de marcha.

Las fases de apoyo y oscilación son análogas de acuerdo a qué piernas se realiza el análisis. De acuerdo a la *Figura 6*, se analiza las fases relativas a la pierna derecha (segmento gris).

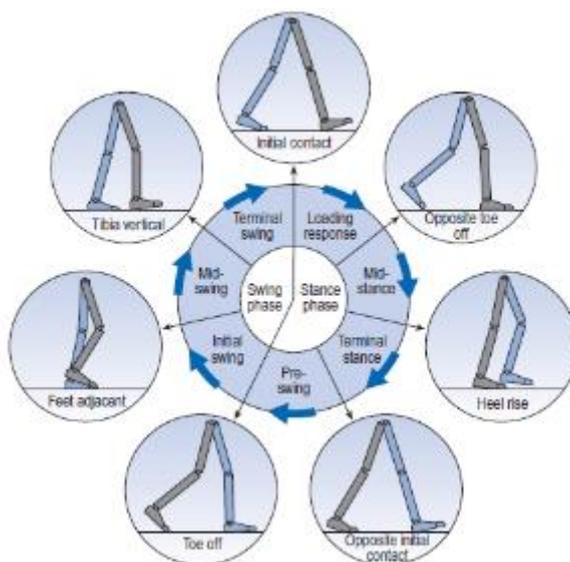


Figura 6 Posiciones de las piernas en una marcha simple (Fuente: *Gait Analysis, Whittle M.W.*)

En la fase de apoyo ocurren cuatro periodos (Whittle, 2007, pág. 53):

- i. Carga de respuesta (Loading response)
- ii. Postura Media (Mid-Stance)
- iii. Postura terminal (Terminal stance)
- iv. Pre-Oscilación (Pre-Swing)

Y en la fase de oscilación, tres periodos (Whittle, 2007, pág. 53):

- i. Oscilación Inicial (Initial Swing)
- ii. Oscilación Media (Mid-Swing)
- iii. Oscilación Terminal (Terminal Swing)

Y dentro de estas dos fases, se tienen eventos en donde se determina la posición de la pierna derecha, la cual se ha tomado como referencia.

Estos eventos son (Whittle, 2007, pág. 53):

1. Contacto Inicial (Initial contact), se caracteriza porque el centro de gravedad del cuerpo se encuentra en su posición más baja.
 2. Despegue de la punta del pie opuesto (Opposite toe off), la planta del pie de referencia se encuentra completamente en la superficie de contacto.
 3. Elevación de talón (Heel rise), el pie que realiza la oscilación pasa a la fase de apoyo. Centro de gravedad del cuerpo se ubica en la posición más alta.
 4. Contacto inicial opuesto (Opposite initial contact), el talón del pie de referencia comienza a perder contacto con la superficie.
 5. Despegue de la punta del pie (Toe off), comienzo de la fase de balanceo por parte de la pierna de referencia.
 6. Pies adyacentes (Feet adjacent), a medida que el pie de referencia acelera, hay un movimiento de cadera para ayudar al balanceo de la pierna.
 7. Tibia en posición vertical (Tibia vertical), reducción de la aceleración de segmento de referencia.
- (1. Contacto Inicial)

Esta nomenclatura puede variar de acuerdo a las diferentes publicaciones sobre el ciclo de la marcha.

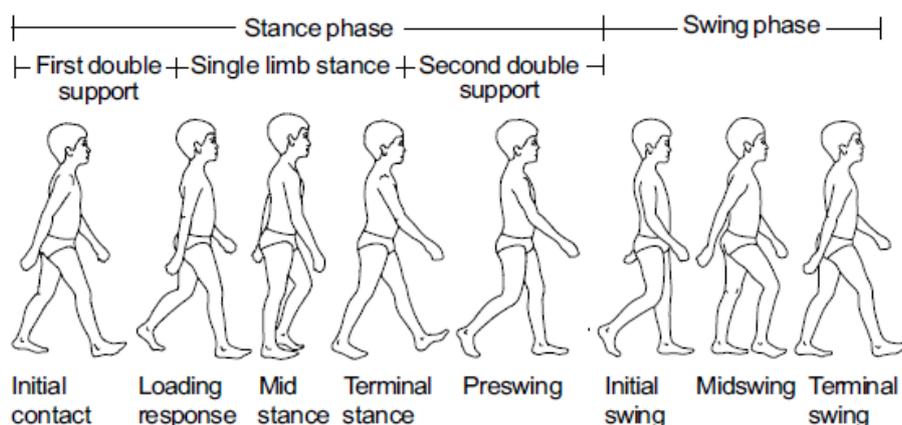


Figura 7 Ciclo normal de la marcha, niño de 8 años. (Fuente: *Dynamics of Human Gait*, Vaughan)

Durante el ciclo de la marcha se deja al descubierto el tiempo utilizado para la fase de apoyo y balanceo de los segmentos inferiores del cuerpo humano, de acuerdo a las características del propio movimiento.

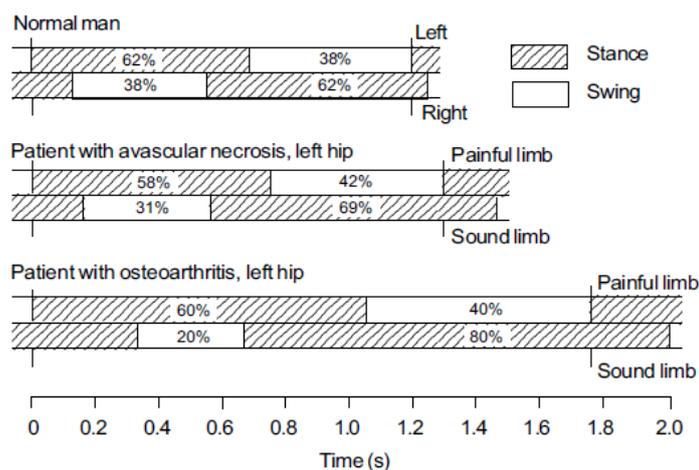


Figura 8 Tiempo que está apoyada una extremidad inferior en un ciclo de marcha de una persona normal y dos pacientes. (Fuente: *Dynamics of Human Gait*, Vaughan)

En la parte superior de la figura se muestra la marcha normal de una persona. Existe una simetría en el porcentaje del tiempo usado por la pierna izquierda y derecha del individuo en la fase de apoyo. Por otro lado, si el sujeto padece de alguna deficiencia en la marcha, se puede ver que utilizará más tiempo de apoyo la pierna contraria al lado deficiente, además depende de qué tipo de lesión se trata, de este modo se contará con un porcentaje de tiempo de apoyo menor o mayor.

Existen otros parámetros que puedan explicar las características únicas de una marcha describiendo la longitud, ancho de paso, ángulo del paso, la frecuencia del paso y la velocidad de la marcha.

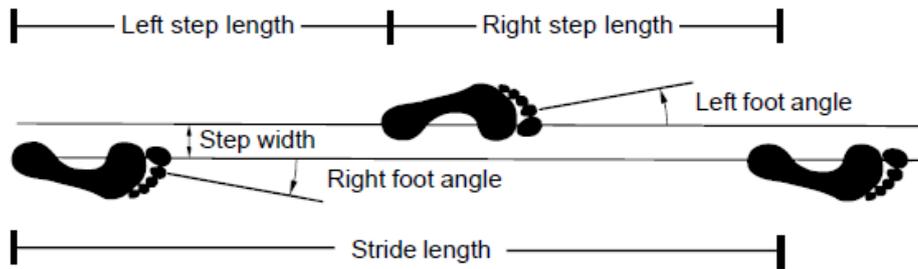


Figura 9 Las huellas de una persona normalmente proveen parámetros de distancia útiles. (Fuente: *Dynamics of Human Gait, Vaughan*)

La longitud de la zancada (Stride length) es medida desde el inicio hasta el final de un ciclo completo de la marcha, y se mide desde la primera posición del talón hasta la segunda posición, final, del talón del mismo pie de inicio.

Para una marcha normal, se tendría, en promedio, que la longitud de dos pasos (Left step length + Right step length) sea igual a la longitud de una zancada. Y para marchas anormales, esta proporción no se mantendría simétrica. Con respecto al ancho de paso (Step width), una marcha normal tiene un ancho de paso de pocos centímetros a comparación de una marcha de una persona con problemas de balance, como por ejemplo personas con ataxia cerebelosa (pérdida de la coordinación al caminar) o parálisis cerebral parcial, cuyo ancho de paso aumenta en 10 ó 20 cm. Algo similar podríamos mencionar sobre el ángulo de inclinación de la huella del pie (Right/Left foot angle) con respecto a una línea horizontal referencial.

2.3 MODELIZACION DE SISTEMAS MECÁNICOS

Ya que el modelo en 2D del cuerpo humano, para el análisis Biomecánico, es un conjunto de sólidos rígidos articulados entre ellos, estos sólidos rígidos representan una porción del cuerpo humano y que los llamaremos segmentos corporales, cuyas características másicas e inerciales se obtienen mediante tablas antropométricas. Es necesario estudiar mecanismos simples que nos permitan identificar y conceptualizar la metodología del análisis respectivo. Para que, finalmente, podamos llevar esa metodología de estudio al modelo final de todo este trabajo.

Para empezar, es primordial establecer las coordenadas en las que definirá el mecanismo a estudiar. Estas coordenadas determinan la posición del sistema. Y cuando existe una variación de estas coordenadas con respecto al tiempo, se puede definir el movimiento del sistema.

2.3.1 Coordenadas naturales (Cuadrado, 1999)

Las coordenadas naturales son un tipo de coordenadas dependientes, ya que el número de parámetros es mayor al número de grados de libertad del mecanismo. Pero este tipo de coordenadas, a diferencia de las coordenadas relativas y de punto de referencia, permiten definir a cada segmento con independencia de los demás. Las coordenadas de cada elemento se ubicarán en los pares, de esta manera se definen dos elementos consecutivos. Las variables de tipo angular no son necesarias para la definición de la orientación del elemento, ya que las ecuaciones que se establecerían podrían resultar tediosas.

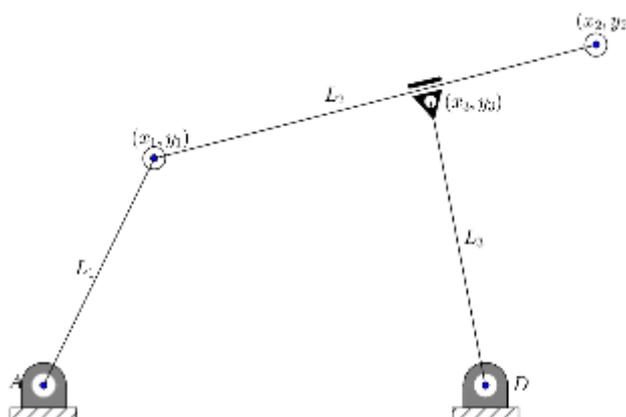


Figura 10 Coordenadas naturales

Entonces, como tenemos más parámetros que grados de libertad, existirán unas relaciones que los ligen. Estas relaciones se denominan ecuaciones de restricción. El número de restricciones r , lo determinamos por la siguiente ecuación:

$$r = n - g \dots (1)$$

Donde n es el número de coordenadas dependientes utilizadas y g es los grados de libertad del mecanismo.

Para nuestro ejemplo, *Figura 10*, tenemos que $n = 6$ y $g = 1$, por lo tanto, es necesario 5 ecuaciones de restricción.

A continuación se definen las ecuaciones de restricción para el mecanismo de la *Figura 10*.

$$(x_1 - x_A)^2 + (y_1 - y_A)^2 - L_1^2 = 0 \dots (2)$$

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 - L_2^2 = 0 \dots (3)$$

$$(x_3 - x_D)^2 + (y_3 - y_D)^2 - L_3^2 = 0 \dots (4)$$

Estas tres ecuaciones aseguran que la longitud de los segmentos que unen los puntos no varíe, imponiendo el carácter rígido a los segmentos.

Luego, tenemos la ecuación de restricción que determina que los segmentos L_2 y L_3 sean perpendiculares. Sean los vectores \vec{L}_2 y \vec{L}_3 :

$$\vec{L}_2: \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix}$$

$$\vec{L}_3: \begin{bmatrix} x_3 - x_D \\ y_3 - y_D \end{bmatrix}$$

Entonces, se tiene que cumplir:

$$\vec{L}_2 \cdot \vec{L}_3 = 0 \text{ (Condición de perpendicularidad)}$$

$$(x_2 - x_1)(x_3 - x_D) + (y_2 - y_1)(y_3 - y_D) = 0 \dots (5)$$

Por último, se define la condición para que el punto 3 pertenezca en cualquier instante al segmento L_2 .

$$\vec{L}_2 \times \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} = 0$$

$$(x_3 - x_1)(y_2 - y_1) - (y_3 - y_1)(x_2 - x_1) = 0 \dots (6)$$

Podemos concluir que las ventajas del uso de coordenadas naturales son de uso simple y que las ecuaciones de restricción son fáciles de definir.

Ya que el modelo que se desarrolla más adelante es en la perspectiva plana, se desarrollará un análisis detallado sobre las coordenadas naturales en el caso plano.

2.3.1.1 Coordenadas naturales en caso plano

Las coordenadas naturales se definen como puntos básicos en el plano cartesiano. Además para poder realizar un correcto modelamiento de un mecanismo (Cuadrado, 1999):

- Para poder definir la posición de un sólido rígido, es necesario definir, por lo menos, dos puntos básicos.
- En cada articulación debe ubicarse un punto básico, de esta manera los dos sólidos contiguos comparten dicho punto, imponiendo la condición de par de revolución.
- En caso de pares prismáticos, se define el eje por medio de dos puntos básicos.
- Pueden utilizarse más puntos básicos en caso se quieran definir ángulos, distancias, puntos concretos de interés, etc.

Veamos la aplicación de estas reglas en el mecanismo de biela-manivela que se presenta a continuación, e introduciremos el ángulo que forma la manivela con la horizontal como variable:

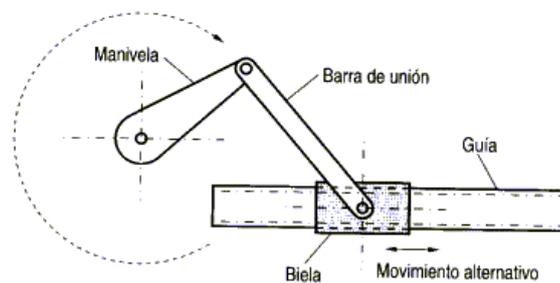


Figura 11 Mecanismo biela-manivela

De acuerdo con lo anterior, el modelo con coordenadas naturales sería:

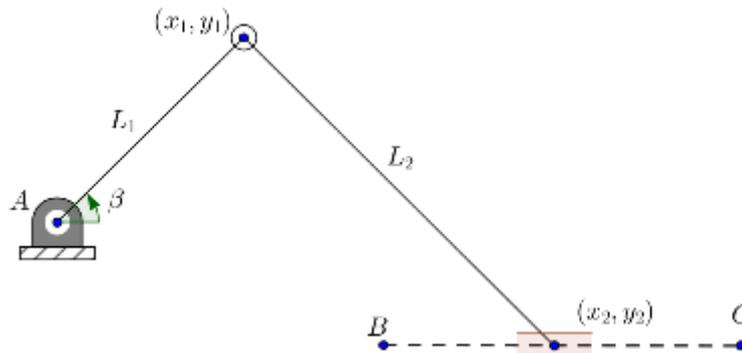


Figura 12 Modelización del mecanismo biela-manivela en coordenadas naturales

Podemos definir los sólidos rígidos L_1 y L_2 mediante los puntos (A_x, A_y) , (x_1, y_1) y (x_2, y_2) . Además el par prismático está determinado por los puntos B y C. Además, los puntos A, B y C ya están determinados por condiciones iniciales al modelamiento. Entonces tenemos que las variables en nuestro modelo son 5, las coordenadas del punto 1, del punto 2 y el ángulo de la manivela.

A continuación se calcula el número de restricciones que se necesitaría para poder terminar la posición del mecanismo en un determinado instante. Para el mecanismo de la Figura 12, $n = 5$ y $g = 1$, por lo tanto, se necesita 4 ecuaciones de restricción:

$$(x_1 - x_A)^2 + (y_1 - y_A)^2 - L_1^2 = 0 \dots (7)$$

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 - L_2^2 = 0 \dots (8)$$

$$(x_2 - x_C)(y_C - y_B) - (y_2 - y_C)(x_C - x_B) = 0 \dots (9)$$

$$(x_1 - x_A) - L_1 \cos \beta = 0 \dots (10)$$

Las ecuaciones de restricción 7 y 8, claramente, determinan la condición de sólido rígido de la manivela y la biela, respectivamente. También se vio anteriormente la ecuación de 9, la cual expresa el par de la deslizadera (guía). Por último, la ecuación 10 permite relacionar el ángulo que forma la manivela con las demás variables. Esta ecuación se obtiene del producto escalar del vector \vec{L}_1 y el unitario (1,0).

Cuando se realiza el modelamiento con coordenadas naturales, permite ser flexible. En la *Figura 3*, se pudo prescindir de la variable y_2 , ya que los puntos B, 2 y C pertenecen a una misma recta ($y_2 - y_C = 0$). En consecuencia, la ecuación de restricción 9 no sería necesaria. Por lo tanto, ya no tendríamos 5 restricciones, sino 4.

2.3.1.2 Ecuación del ángulo (Cuadrado, 1999)

Dado el siguiente elemento articulado, modelado como se muestra, *Figura 13*.

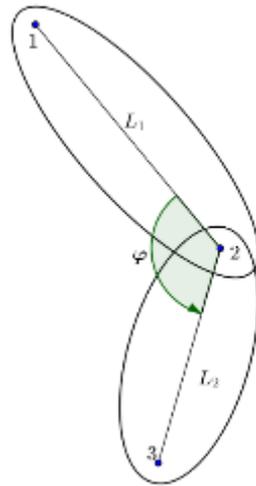


Figura 13 Ángulo entre dos segmentos.

Para definir el ángulo adecuado entre estos dos sólidos, se tiene que relacionar con los puntos básicos que definen los segmentos. Una de las maneras de relacionarlos es con el producto escalar de los vectores

$$\vec{L}_1 \cdot \vec{L}_2 - |\vec{L}_1| |\vec{L}_2| \cos \varphi = 0$$

$$(x_1 - x_2)(x_3 - x_2) + (y_1 - y_2)(y_3 - y_2) - |\vec{L}_1| |\vec{L}_2| \cos \varphi = 0 \dots (11)$$

Pero también es válido relacionarlos por medio del producto vectorial:

$$\vec{L}_1 \times \vec{L}_2 - |\vec{L}_1| |\vec{L}_2| \sin \varphi = 0$$

$$(x_1 - x_2)(y_3 - y_2) - (y_1 - y_2)(x_3 - x_2) - |\vec{L}_1| |\vec{L}_2| \sin \varphi = 0 \dots (12)$$

Ahora la cuestión es saber cuál de estas dos restricciones del ángulo se debería utilizar. La mejor manera de determinar cuál de ellas, es de acuerdo a la variación del ángulo según lo siguiente:

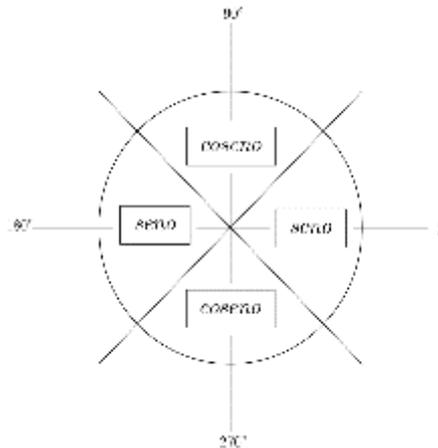


Figura 14 Ángulo según posición.

Puede suceder que el ángulo entre dos sólidos rígidos varíe en las regiones que corresponda utilizar las ecuaciones 11 y 12, con lo cual se podría plantear las dos ecuaciones en simultáneo, teniendo un sistema con ecuaciones redundantes.

2.3.2 Estudio de mecanismos comunes

Por medio de ejemplos de mecanismos y con ayuda computacional de Matlab, realizaremos el modelamiento de dichos mecanismos, aplicando coordenadas naturales, y que finalmente se creará una herramienta (algoritmo) que permita la simulación del movimiento. Cabe resaltar que para solucionar el problema cinemático (consistencia cinemática) de los mecanismos, es simplemente resolver el problema geométrico el cual restringe el movimiento, siendo el movimiento independiente de las fuerzas que actúan en el mecanismo. Los valores de los datos de la posición inicial, velocidades y aceleraciones son conocidos por el analista.

2.3.3 Mecanismo de cuatro barras

El cuadrilátero articulado es un mecanismo muy básico y fácil de modelar. Consta con un mínimo de tres barras y dos puntos fijos a una superficie, lo cual la distancia entre ellos no varía; y también se podría modelar estos dos puntos fijos como los extremos de un sólido rígido.

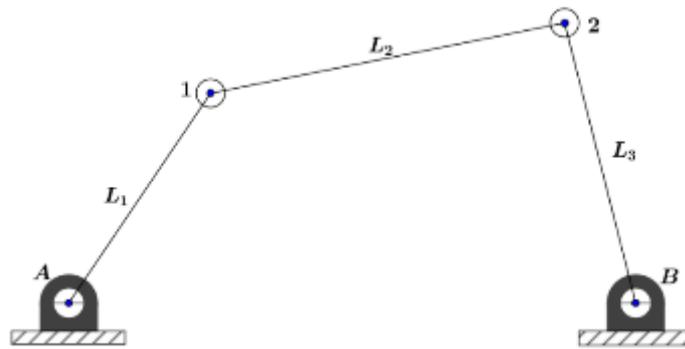


Figura 15 Mecanismo de cuatro barras

Dependiendo de las longitudes de las barras, determinadas por el analista, el mecanismo tiene diferentes comportamientos en su movimiento.

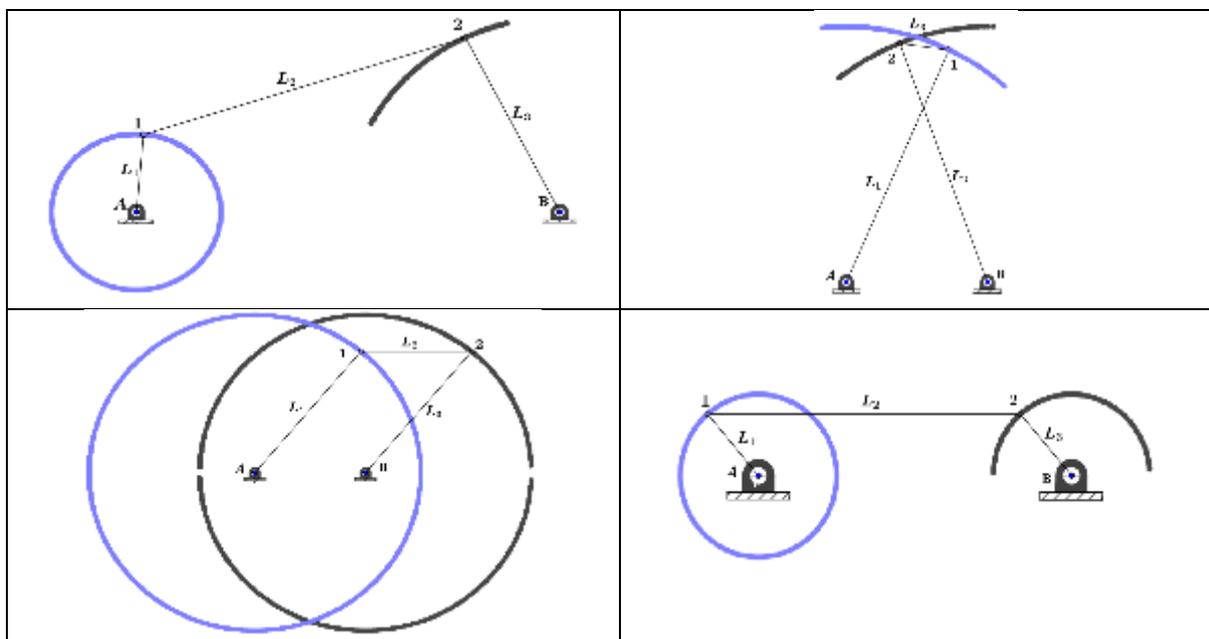


Figura 16 Tipos de mecanismos de cuatro barras

Determinaremos que la posición inicial del mecanismo es tal como se dispone el modelo de la Figura 15. Como habíamos mencionado anteriormente, para definir el número de ecuaciones de restricción es necesario saber los grados de libertad del mecanismo, que en este caso solo se tiene un grado de libertad, y el número de parámetros necesarios para definir el mecanismo, que son cinco: x_1, y_1, x_2, y_2 y α . Entonces el número de ecuaciones de restricción es 4. La posición del punto A y del punto B ya está determinadas, así como las longitudes de los segmentos.

Podemos definir que el vector de coordenadas sería:

$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \alpha \end{bmatrix} \dots (13)$$

Definiendo las ecuaciones de restricción:

$$(x_1 - x_A)^2 + (y_1 - y_A)^2 - L_1^2 = 0 \dots (14a)$$

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 - L_2^2 = 0 \dots (14b)$$

$$(x_2 - x_B)^2 + (y_2 - y_B)^2 - L_3^2 = 0 \dots (14c)$$

$$(x_1 - x_A) - L_1 \cos \alpha = 0 \dots (14d)$$

Estas ecuaciones están agrupadas en el vector de restricciones $\Phi(\mathbf{q})$.

$$\Phi(\mathbf{q}) = \begin{bmatrix} (x_1 - x_A)^2 + (y_1 - y_A)^2 - L_1^2 \\ (x_2 - x_1)^2 + (y_2 - y_1)^2 - L_2^2 \\ (x_2 - x_B)^2 + (y_2 - y_B)^2 - L_3^2 \\ (x_1 - x_A) - L_1 \cos \alpha \end{bmatrix} \dots (15)$$

Por lo tanto, para determinar la solución al sistema de ecuaciones de restricción.

$$\Phi(\mathbf{q}) = \mathbf{0} \dots (16)$$

El sistema de ecuaciones propuesto, es un sistema no lineal. Pero con la linealización de la ecuación 16 permite aplicar el desarrollo de la serie de Taylor para encontrar la posición inicial aproximada.

$$\Phi(\mathbf{q}) \cong \Phi(\mathbf{q}_0) + \Phi_q(\mathbf{q}_0)(\mathbf{q} - \mathbf{q}_0) = \mathbf{0} \dots (17)$$

En donde \mathbf{q}_0 está definido como el vector de coordenadas aproximadas:

$$\mathbf{q}_0 = \begin{bmatrix} x_1^0 \\ y_1^0 \\ x_2^0 \\ y_2^0 \\ \alpha \end{bmatrix} \dots (18)$$

Y el término $\Phi_q(\mathbf{q})$ es el jacobiano del vector de restricciones, que en otras palabras es la matriz que contiene las derivadas de las ecuaciones respecto a las variables que deseamos conocer.

$$\Phi_q(\mathbf{q}) = \begin{bmatrix} 2(x_1 - x_A) & 2(y_1 - y_A) & 0 & 0 & 0 \\ -2(x_2 - x_1) & -2(y_2 - y_1) & 2(x_2 - x_1) & 2(y_2 - y_1) & 0 \\ 0 & 0 & 2(x_2 - x_B) & 2(y_2 - y_B) & 0 \\ 1 & 0 & 0 & 0 & L_1 \sin \alpha \end{bmatrix} \dots (19)$$

En el problema de la posición inicial, se estiman las posiciones iniciales, y estos valores introducidos en las ecuaciones de restricción no serán igual a cero, con lo cual se puede reescribir la ecuación 17 de la siguiente manera:

$$\Phi_q(\mathbf{q}_0)(\mathbf{q} - \mathbf{q}_0) = -\Phi(\mathbf{q}_0) \dots (20)$$

Los valores de las variables del vector de coordenadas \mathbf{q} serán aproximados luego de haber resuelto la ecuación 20. Como este nuevo vector es una primera aproximación, se puede denotar de la siguiente manera:

$$\mathbf{q}_1 = \begin{bmatrix} x_1^1 \\ y_1^1 \\ x_2^1 \\ y_2^1 \\ \alpha \end{bmatrix} \dots (21)$$

Nótese que la variable que corresponde al ángulo, no cambia con respecto a \mathbf{q}_0 ya que es justo la posición en la que se quiere definir el mecanismo.

Si se resuelve la ecuación 20 de forma iterativa, podemos obtener un vector \mathbf{q} que sea óptimo para que los valores de las coordenadas cumplan con las restricciones (14). Con esta iteración de aproximación a la solución de nuestro sistema de ecuaciones no lineales, se aplica el método iterativo de Newton-Raphson.

En este caso, generalizamos:

$$\Phi_{\mathbf{q}}(\mathbf{q}_i)(\mathbf{q}_{i+1} - \mathbf{q}_i) = -\Phi(\mathbf{q}_i) \dots (21)$$

Donde $i=1, 2, 3, \dots$

Como el valor de la variable del grado de libertad, en este caso el ángulo, es conocido, se elimina la columna correspondiente a éste en el jacobiano y en el vector de coordenadas, solo para realizar estas iteraciones. Se verifica que al realizar el producto del jacobiano $\Phi_{\mathbf{q}}(\mathbf{q}_i)$ por $(\mathbf{q}_{i+1} - \mathbf{q}_i)$, y por el orden de $-\Phi(\mathbf{q}_i)$, ahora ya son compatibles en tamaño.

Desarrollando:

$$\begin{bmatrix} 2(x_1^i - x_A) & 2(y_1^i - y_A) & 0 & 0 \\ -2(x_2^i - x_1) & -2(y_2^i - y_1^i) & 2(x_2^i - x_1^i) & 2(y_2^i - y_1^i) \\ 0 & 0 & 2(x_2^i - x_B) & 2(y_2^i - y_B) \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^{i+1} - x_1^i \\ y_1^{i+1} - y_1^i \\ x_2^{i+1} - x_2^i \\ y_2^{i+1} - y_2^i \end{bmatrix} = - \begin{bmatrix} (x_1^i - x_A)^2 + (y_1^i - y_A)^2 - L_1^2 \\ (x_2^i - x_1^i)^2 + (y_2^i - y_1^i)^2 - L_2^2 \\ (x_2^i - x_B)^2 + (y_2^i - y_B)^2 - L_3^2 \\ (x_1^i - x_A) - L_1 \cos \alpha \end{bmatrix} - \alpha \begin{bmatrix} 0 \\ 0 \\ 0 \\ L_1 \sin \alpha \end{bmatrix}$$

2.3.3.1 Simulación del mecanismo y del problema de la posición inicial en MATLAB

Se realiza el modelamiento en Matlab, desde el cual podemos obtener datos que muestren el proceso iterativo para encontrar la posición inicial.

Determinaremos los puntos fijos: A(0,0) y B(8,0).

Y la longitud de las barras: L1=2; L2=6; L3=4. Valor del ángulo: $\alpha = \frac{\pi}{3}$.

Se establece como aproximaciones iniciales: $\mathbf{q}_0 = \begin{bmatrix} 1 \\ 2 \\ 6 \\ 3 \\ \pi/3 \end{bmatrix}$

A partir de estos valores, y casi próximos a los óptimos, se comenzará la iteración para encontrar los valores numéricos de las variables que cumplan las ecuaciones de restricción, puesto que los aproximados iniciales no los cumplirán, con una diferencia distinta y un poco alejada de cero. Cuando se realiza una iteración determinada, la norma del vector de restricciones $\Phi(\mathbf{q}_0)$ indicará que tan cerca a la solución óptima están las variables calculadas. La norma de este vector podría tomarse como un error (e), ya que este valor, y considerarlo como aceptable, no deberá de ser mayor al orden de 10^{-8} .

Tabla 1 Valores de variables en cada iteración para encontrar la posición óptima.

Iteración	x_1	y_1	x_2	y_2	α	e
0	1	1.75	6.75	4	1.0472	2.6348
1	1	1.7321	6.6472	3.7726	1.0472	0.0828
2	1	1.7321	6.6456	3.7637	1.0472	1.1353e-004
3	1	1.7321	6.6425	3.7637	1.0472	2.0726e-010

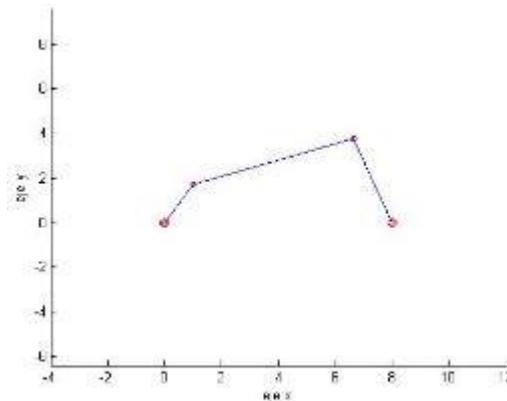


Figura 17 Posición Inicial

En la *Figura 17* se grafica el resultado de la solución del problema de la posición, obteniendo finalmente el vector de coordenadas:

$$\mathbf{q} = \begin{bmatrix} 1 \\ 1.7321 \\ 6.6425 \\ 3.7637 \\ \pi/3 \end{bmatrix}$$

Intuitivamente podemos determinar que cuando varía el valor del ángulo, esto es el grado de libertad, del mecanismo, obtendremos diferentes valores de las coordenadas. Haciendo una extensión de varios valores del grado de libertad, se podría montar una animación de movimientos finitos. A continuación se presentará la solución del problema de los desplazamientos finitos para el ejemplo del cuadrilátero articulado anterior. Con una variación del grado de libertad entre 22.5° hasta 172.5° .

RESULTADOS DE SIMULACIÓN EN MATLAB

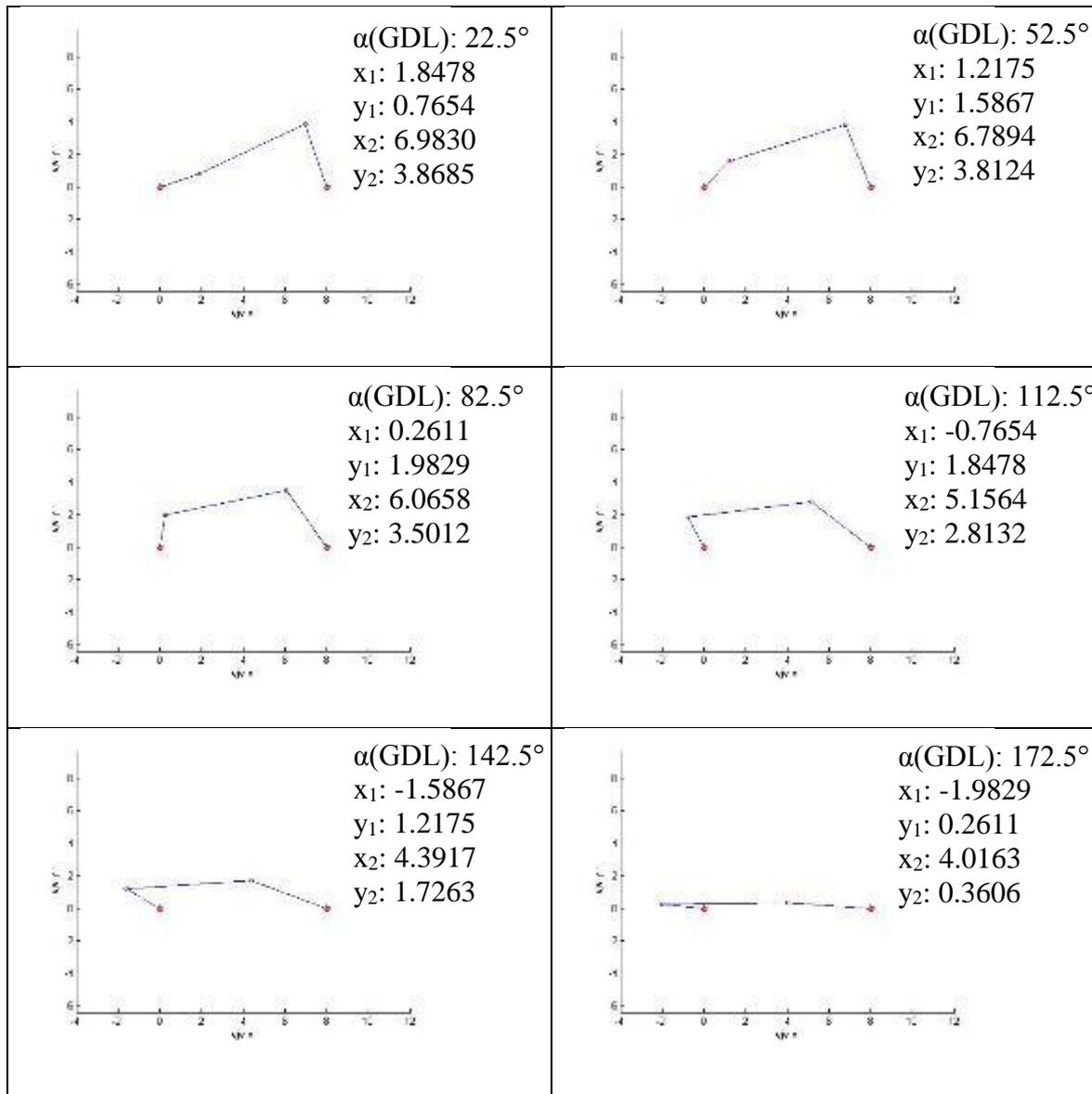


Figura 18 La simulación del movimiento de este mecanismo de cuatro barras, se ha establecido como un movimiento en sentido anti horario

2.3.3.2 Problema de la velocidad

Una vez determinadas las posiciones de las coordenadas de las variables, en función de los valores de los grados de libertad, también se puede saber las derivadas temporales (velocidades) de estas coordenadas; teniendo como dato las derivadas temporales de los grados de libertad de un determinado mecanismo.

Cuando se resolvió el problema de la posición, partíamos de la ecuación (16):

$$\Phi(q) = 0$$

Derivando esta ecuación, por regla de la cadena, se obtiene:

$$\Phi_q(q)\dot{q} = 0 \dots(22)$$

La matriz jacobiana es conocida, a partir del problema de la posición inicial. Esto permite obtener un sistema de ecuaciones lineales, cuyas incógnitas son las derivadas temporales de las posiciones de las coordenadas.

Del problema del cuadrilátero, se resuelve la ecuación (22), y como dato:

$$q = \begin{bmatrix} 1 \\ 1.7321 \\ 6.6425 \\ 3.7637 \\ \pi/3 \end{bmatrix}$$

Entonces,

$$\begin{bmatrix} 2 & 3.4642 & 0 & 0 & 0 \\ -11.285 & -4.0632 & 11.285 & 4.0632 & 0 \\ 0 & 0 & -2.715 & 7.5274 & 0 \\ 1 & 0 & 0 & 0 & 1.7321 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \dots(23)$$

Como se había mencionado, la velocidad del grado de libertad es conocida, y a partir de esto, se puede reescribir la ecuacione (23) para luego hallar los valores de las velocidades de las coordenadas.

$$\begin{bmatrix} 2 & 3.4642 & 0 & 0 \\ -11.285 & -4.0632 & 11.285 & 4.0632 \\ 0 & 0 & -2.715 & 7.5274 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = -\dot{\alpha} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.7321 \end{bmatrix} \dots(24)$$

Por ejemplo, si el valor de la velocidad de $\dot{\alpha} = 1$, entonces:

$$\dot{q} = \begin{bmatrix} -1.7320 \\ 1 \\ -1.2149 \\ -0.4372 \\ 1 \end{bmatrix}$$

2.3.3.3 Problema de la aceleración

Análogamente, las aceleraciones de las coordenadas se calculan con la derivada temporal de la velocidad de las mismas y conociendo el valor de la aceleración de los grados de libertad existentes en el sistema que se analiza. De la ecuación (22), se deriva para obtener:

$$\frac{\partial \Phi_q(q)\dot{q}}{\partial t} = \dot{\Phi}_q\dot{q} + \Phi_q(q)\ddot{q} = 0$$

$$\therefore \Phi_q(q)\ddot{q} = -\dot{\Phi}_q\dot{q}\dots(25)$$

En la ecuación (25), la matriz jacobiana es conocida, nuevamente, porque ya se resolvió el problema de la posición inicialmente. Al igual que $-\dot{\Phi}_q\dot{q}$ son conocidos porque están en función de las velocidades, previamente calculados. Seguiremos con el ejemplo del cuadrilátero articulado para resolver la ecuación (25). Primero obtenemos $-\dot{\Phi}_q\dot{q}$, como resultado de derivar la ecuación (19) por el vector de las derivadas temporales de las coordenadas:

$$-\dot{\Phi}_q\dot{q} = - \begin{bmatrix} 2\dot{x}_1 & 2\dot{y}_1 & 0 & 0 & 0 \\ -2(\dot{x}_2 - \dot{x}_1) & -2(\dot{y}_2 - \dot{y}_1) & 2(\dot{x}_2 - \dot{x}_1) & 2(\dot{y}_2 - \dot{y}_1) & 0 \\ 0 & 0 & 2(\dot{x}_2 - \dot{x}_B) & 2(\dot{y}_2 - \dot{y}_B) & 0 \\ 0 & 0 & 0 & 0 & (L_1 \cos \alpha)\dot{\alpha} \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \\ \dot{\alpha} \end{bmatrix}$$

$$-\dot{\Phi}_q\dot{q} = - \begin{bmatrix} 2(\dot{x}_1^2 + \dot{y}_1^2) \\ 2\{(\dot{x}_2^2 - \dot{x}_1^2) + (\dot{y}_2^2 - \dot{y}_1^2)\} \\ 2(\dot{x}_2^2 + \dot{y}_2^2) \\ (L_1 \cos \alpha)\dot{\alpha}^2 \end{bmatrix} = - \begin{bmatrix} 8.00 \\ 4.66 \\ 3.33 \\ 1 \end{bmatrix}$$

$$\Phi_q(q)\ddot{q} = \begin{bmatrix} 2 & 3.4642 & 0 & 0 & 0 \\ -11.285 & -4.0632 & 11.285 & 4.0632 & 0 \\ 0 & 0 & -2.715 & 7.5274 & 0 \\ 1 & 0 & 0 & 0 & 1.7321 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \\ \ddot{\alpha} \end{bmatrix} = - \begin{bmatrix} 8.00 \\ 4.66 \\ 3.33 \\ 1 \end{bmatrix}$$

Como las aceleraciones de las coordenadas se pueden determinar en función de la aceleración del grado de libertad,

$$\begin{bmatrix} 2 & 3.4642 & 0 & 0 \\ -11.285 & -4.0632 & 11.285 & 4.0632 \\ 0 & 0 & -2.715 & 7.5274 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = - \begin{bmatrix} 8.00 \\ 4.66 \\ 3.33 \\ 1 \end{bmatrix} - \ddot{\alpha} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.7321 \end{bmatrix}$$

Entonces, cuando $\ddot{\alpha} = 1.5$, el vector de las aceleraciones de las coordenadas sería:

$$\ddot{q} = \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} -3.5981 \\ -0.2321 \\ -3.4842 \\ -1.6968 \\ 1.5000 \end{bmatrix}$$

2.3.4 Manipulador plano

Este tipo de mecanismo consiste en tres puntos fijos y seis barras, las cuales tres de ellas comparten una misma unión, este punto en común sigue una trayectoria determinada. Para generalizar, se dispone del siguiente manipulador con las siguientes características:

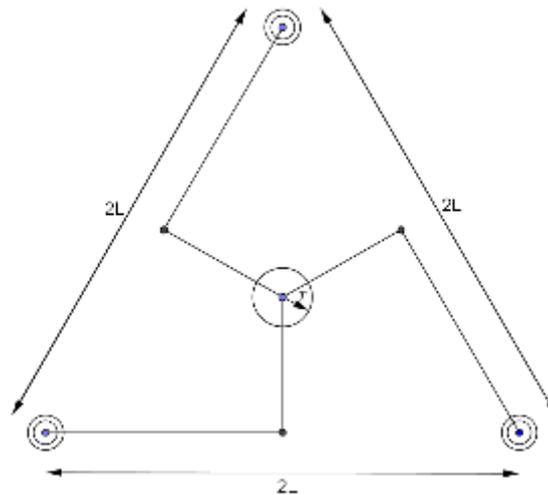


Figura 19 Manipulador plano

Modelamos el mecanismo señalando las coordenadas:

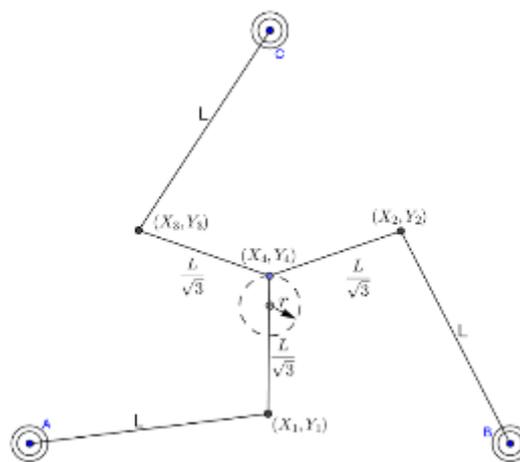


Figura 20 Modelamiento del Manipulador plano

Problema de la posición inicial

Establecemos los datos del mecanismo:

- $L = 2$
- Puntos fijos: $A(-L, \frac{-L}{\sqrt{3}})$; $B(L, \frac{-L}{\sqrt{3}})$; $C(0, \frac{2L}{\sqrt{3}})$
- $r = 0.25$
- Grado de libertad: Punto 4(X_4, Y_4)

Y las condiciones de velocidad y aceleración del grado de libertad:

$$\dot{x}_4 = 2 \cdot \pi \cdot r \cdot \cos(\alpha) \quad \dot{y}_4 = -2 \cdot \pi \cdot r \cdot \sin(\alpha)$$

$$\ddot{x}_4 = 8 \quad \ddot{y}_4 = 8$$

Valores iniciales de las coordenadas para iniciar la aproximación de la posición inicial:

$$\mathbf{q}_0 = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \\ 2 \\ -1 \\ 1 \\ 0 \\ 0.25 \end{bmatrix}$$

El vector de restricciones:

$$\Phi(\mathbf{q}) = \begin{bmatrix} (x_1 - x_A)^2 + (y_1 - y_A)^2 - L^2 \\ (x_2 - x_B)^2 + (y_2 - y_B)^2 - L^2 \\ (x_3 - x_C)^2 + (y_3 - y_C)^2 - L^2 \\ (x_4 - x_1)^2 + (y_4 - y_1)^2 - \left(\frac{L}{\sqrt{3}}\right)^2 \\ (x_4 - x_2)^2 + (y_4 - y_2)^2 - \left(\frac{L}{\sqrt{3}}\right)^2 \\ (x_4 - x_3)^2 + (y_4 - y_3)^2 - \left(\frac{L}{\sqrt{3}}\right)^2 \end{bmatrix}$$

Y como jacobiano de nuestro vector de restricciones tenemos:

$$\Phi_q(\mathbf{q}) = \begin{bmatrix} 2(x_1 - x_A) & 2(y_1 - y_A) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2(x_2 - x_B) & 2(y_2 - y_B) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(x_3 - x_C) & 2(y_3 - y_C) & 0 & 0 & 0 \\ -2(x_4 - x_1) & -2(y_4 - y_1) & 0 & 0 & 0 & 0 & 2(x_4 - x_1) & 2(y_4 - y_1) & 0 \\ 0 & 0 & -2(x_4 - x_2) & -2(y_4 - y_2) & 0 & 0 & 2(x_4 - x_2) & 2(y_4 - y_2) & 0 \\ 0 & 0 & 0 & 0 & -2(x_4 - x_3) & -2(y_4 - y_3) & 2(x_4 - x_3) & 2(y_4 - y_3) & 0 \end{bmatrix}$$

Tabla 2 Se muestra las iteraciones que convergen en la posición inicial idónea de las coordenadas con las que se modeló el mecanismo

Iteración	x ₁	y ₁	x ₂	y ₂	x ₃	y ₃	x ₄	y ₄	e
0	0	-1	1	2	-1	1	0	0.25	7.5854
1	-0.0131	-0.9083	1.3626	1.0131	-1.1612	0.6323	0	0.25	1.5800
2	-0.0157	-0.9046	1.1369	0.6917	-1.0918	0.6323	0	0.25	0.2182
3	-0.0157	-0.9046	1.0932	0.6295	-1.0896	0.6323	0	0.25	0.0082
4	-0.0157	-0.9046	1.0914	0.6270	-1.0896	0.6323	0	0.25	1.3365e-005
5	-0.0157	-0.9046	1.0914*	0.6270*	-1.0896*	0.6323	0	0.25	3.6054e-011

(*) Estos valores son iguales a la iteración número 4, a consecuencia de la aproximación a 4 decimales.

Entonces, luego de resolver la ecuación (20) para determinar dicha posición que cumple con las restricciones geométricas se obtiene en la 5ta iteración la posición inicial, con una tolerancia menor al orden de 10^{-10} .

$$\mathbf{q} = \begin{bmatrix} -0.0157 \\ -0.9046 \\ 1.0914 \\ 0.6270 \\ -1.0896 \\ 0.6323 \\ 0.0000 \\ 0.2500 \end{bmatrix}$$

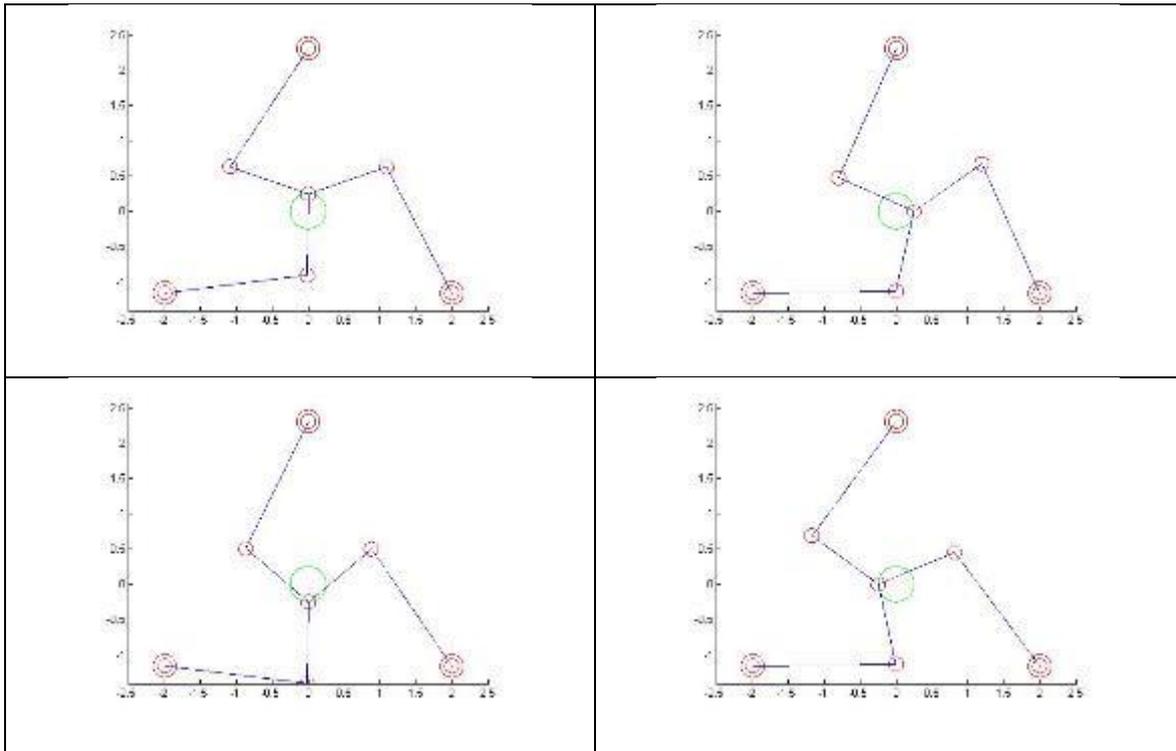


Figura 21 Diferentes posiciones del mecanismo, cuyo desplazamiento describe una circunferencia en sentido horario.

Para resolver el problema de la velocidad, tenemos como datos conocidos:

$$\dot{x}_4 = 2 \cdot \pi \cdot r \cdot \cos(\alpha) \quad \dot{y}_4 = -2 \cdot \pi \cdot r \cdot \sin(\alpha)$$

A continuación se desarrolla la ecuación (22):

$$\Phi_q(q)\dot{q} = 0$$

$$\begin{bmatrix} 2(x_1 - x_A) & 2(y_1 - y_A) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2(x_2 - x_B) & 2(y_2 - y_B) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(x_3 - x_C) & 2(y_3 - y_C) & 0 & 0 \\ -2(x_4 - x_1) & -2(y_4 - y_1) & 0 & 0 & 0 & 0 & 2(x_4 - x_1) & 2(y_4 - y_1) \\ 0 & 0 & -2(x_4 - x_2) & -2(y_4 - y_2) & 0 & 0 & 2(x_4 - x_2) & 2(y_4 - y_2) \\ 0 & 0 & 0 & 0 & -2(x_4 - x_3) & -2(y_4 - y_3) & 2(x_4 - x_3) & 2(y_4 - y_3) \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \\ \dot{x}_3 \\ \dot{y}_3 \\ \dot{x}_4 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Evidentemente, se desarrolla un sistema homogéneo, que podría resultar como resultado una solución trivial o indeterminada; sin embargo, separando las variables ya conocidas:

$$\begin{bmatrix} 2(x_1 - x_A) & 2(y_1 - y_A) & 0 & 0 & 0 & 0 \\ 0 & 0 & 2(x_2 - x_B) & 2(y_2 - y_B) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(x_3 - x_C) & 2(y_3 - y_C) \\ -2(x_4 - x_1) & -2(y_4 - y_1) & 0 & 0 & 0 & 0 \\ 0 & 0 & -2(x_4 - x_2) & -2(y_4 - y_2) & 0 & 0 \\ 0 & 0 & 0 & 0 & -2(x_4 - x_3) & -2(y_4 - y_3) \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \\ \dot{x}_3 \\ \dot{y}_3 \end{bmatrix} = -\dot{x}_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2(x_4 - x_1) \\ 2(x_4 - x_2) \\ 2(x_4 - x_3) \end{bmatrix} - \dot{y}_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2(y_4 - y_1) \\ 2(y_4 - y_2) \\ 2(y_4 - y_3) \end{bmatrix}$$

En consecuencia, podemos desarrollar ahora un sistema cuya solución no necesariamente es trivial o indeterminada. Como resultado de la resolución de esta ecuación, se obtiene que las velocidades de las coordenadas en la posición inicial son:

$$\dot{q} = \begin{bmatrix} -0.0027 \\ 0.0214 \\ 1.3355 \\ 0.6811 \\ 1.2792 \\ -0.8311 \\ 1.5708 \\ 0.0000 \end{bmatrix}$$

Además podemos representar la velocidad de las coordenadas del punto 1 en diferentes instantes de la simulación del movimiento mediante el siguiente gráfico.

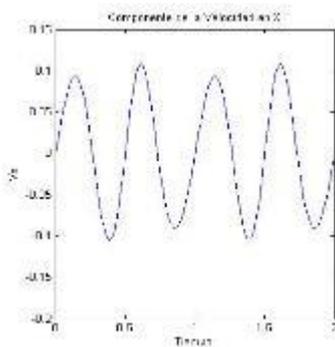


Figura 22 Componente de la velocidad en el eje X

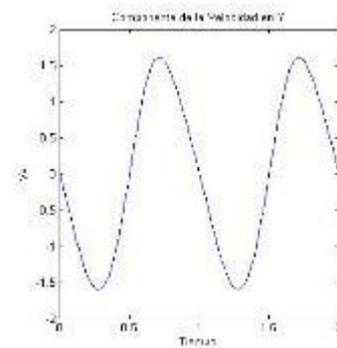


Figura 23 Componente de la velocidad en el eje Y

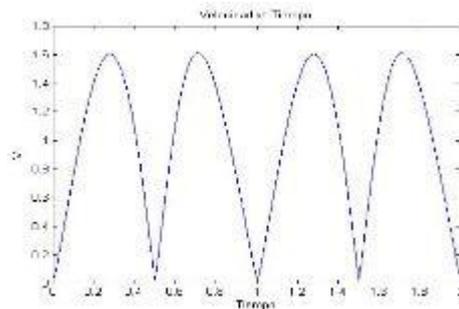


Figura 24 Módulo de la velocidad vs Tiempo

Tal como se muestra en las Figuras 22 y 23, las variaciones de la velocidad son periódicas. En general, el módulo de la velocidad del punto 1, Figura 24, también presenta este comportamiento periódico puesto que las componentes de la velocidad están en relación a una función trigonométrica, para efectos prácticos.

Una vez obtenida las componentes de la velocidad de punto 4, a consecuencia de esto, se conoce la aceleración también de este punto.

La ecuación 25 nos permite el cálculo de la derivada temporal de la velocidad de las coordenadas de los puntos para poder obtener las aceleraciones de las mismas.

$$\Phi_q(q)\ddot{q} = -\Phi_q\dot{q}$$

$$\begin{bmatrix} 2(x_1 - x_A) & 2(y_1 - y_A) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2(x_2 - x_B) & 2(y_2 - y_B) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(x_3 - x_C) & 2(y_3 - y_C) & 0 & 0 \\ -2(x_4 - x_1) & -2(y_4 - y_1) & 0 & 0 & 0 & 0 & 2(x_4 - x_1) & 2(y_4 - y_1) \\ 0 & 0 & -2(x_4 - x_2) & -2(y_4 - y_2) & 0 & 0 & 2(x_4 - x_2) & 2(y_4 - y_2) \\ 0 & 0 & 0 & 0 & -2(x_4 - x_3) & -2(y_4 - y_3) & 2(x_4 - x_3) & 2(y_4 - y_3) \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \\ \ddot{x}_3 \\ \ddot{y}_3 \\ \ddot{x}_4 \\ \ddot{y}_4 \end{bmatrix} = - \begin{bmatrix} 2(x_1^2 + y_1^2) \\ 2(x_2^2 + y_2^2) \\ 2(x_3^2 + y_3^2) \\ 2((x_4 - x_1)^2 + (y_4 - y_1)^2) \\ 2((x_4 - x_2)^2 + (y_4 - y_2)^2) \\ 2((x_4 - x_3)^2 + (y_4 - y_3)^2) \end{bmatrix}$$

Resolviendo el sistema, y agrupando las variables conocidas, en este caso las aceleraciones de las coordenadas del punto 4.

$$\begin{bmatrix} 2(x_1 - x_A) & 2(y_1 - y_A) & 0 & 0 & 0 & 0 \\ 0 & 0 & 2(x_2 - x_B) & 2(y_2 - y_B) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(x_3 - x_C) & 2(y_3 - y_C) \\ -2(x_4 - x_1) & -2(y_4 - y_1) & 0 & 0 & 0 & 0 \\ 0 & 0 & -2(x_4 - x_2) & -2(y_4 - y_2) & 0 & 0 \\ 0 & 0 & 0 & 0 & -2(x_4 - x_3) & -2(y_4 - y_3) \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \\ \ddot{x}_3 \\ \ddot{y}_3 \end{bmatrix} = \begin{bmatrix} 2(x_1^2 + y_1^2) \\ 2(x_2^2 + y_2^2) \\ 2(x_3^2 + y_3^2) \\ 2((x_4 - x_1)^2 + (y_4 - y_1)^2) \\ 2((x_4 - x_2)^2 + (y_4 - y_2)^2) \\ 2((x_4 - x_3)^2 + (y_4 - y_3)^2) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2(x_4 - x_1) \\ 2(x_4 - x_2) \\ 2(x_4 - x_3) \end{bmatrix} \begin{bmatrix} \ddot{x}_4 \\ \ddot{y}_4 \end{bmatrix}$$

Para cada instante de tiempo, se tiene una velocidad determinada de cada una de las coordenadas, así como se obtiene la aceleración de estas mismas. Entonces, como se conoce, por ejemplo, las velocidades de las coordenadas en la posición inicial, y como consecuencia, las aceleraciones de dichas coordenadas.

$$\ddot{q} = \begin{bmatrix} 0.9751 \\ -7.7381 \\ -2.9326 \\ -2.7569 \\ 3.7961 \\ -1.0787 \\ 0.0000 \\ -9.8696 \end{bmatrix}$$

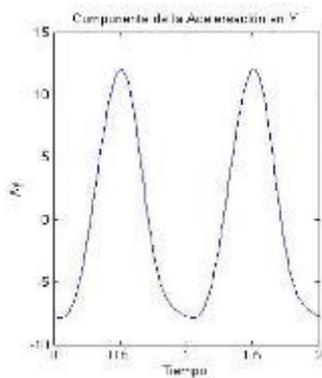


Figura 27 Componente de la aceleración en el eje Y

Las gráficas de las aceleraciones de las coordenadas del punto 1 se muestran a continuación:

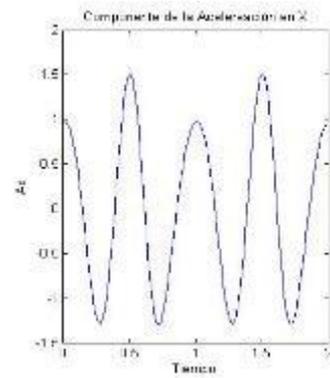


Figura 26 Componente de la aceleración en el eje X

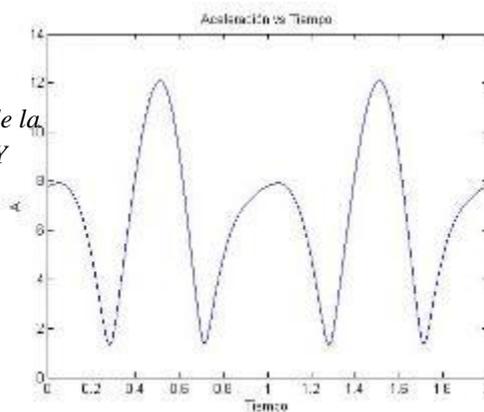


Figura 25 Módulo de la aceleración vs Tiempo

Capítulo 3

CINEMÁTICA Y DINÁMICA ORIENTADA A SEGMENTOS CORPORALES

Luego de realizar el estudio de mecanismos modelados por medio de sólidos rígidos, básicamente es la cinemática de estos sólidos como elementos del tipo barra. A continuación se procede a realizar el análisis de los segmentos corporales, no solo estudiando la cinemática, sino también, la dinámica de estos segmentos.

3.1 MODELIZACIÓN

Los parámetros inerciales de cada segmento corporal son complicados de determinar ya que son características únicas para cada individuo. Estos parámetros son la masa, posición del centro de gravedad y momento de inercia. Existen métodos para el cálculo de estos parámetros inerciales como son los métodos directos e indirectos.

- Métodos directos:
 - Mediciones dinámicas: en donde se emplean acelerómetros y transductores de fuerza para determinar el momento de inercia de cada segmento. Sin embargo, las mediciones resulta algo inexactas y requieren de mucho tiempo.
 - Tablas de reacciones: es un mecanismo que consiste en una tabla con dos apoyos, la cual calcula el peso ejercido por el cuerpo y el punto más próximo al centro de masa. La mayor desventaja de este método es que no puede calcular centro de gravedad de los segmentos en forma independiente.
 - Escáneres: se puede conocer la masa, centro de gravedad y momento de inercia de cada segmento por medio de rayos gamma, tomografía axial o resonancia magnética. Aunque pueda resultar un método exacto y confiable, no podría usarse con mucha frecuencia debido a la exposición de radiación y tener un costo alto.
- Métodos indirectos:
 - Métodos basados en aproximaciones geométricas: este método utiliza formas geométricas que se pueden describir de forma matemática. La desventaja de este método es el número de mediciones que son necesarias para el modelamiento. Algunas consideraciones de este método son de tener la misma densidad en todas las secciones de los segmentos corporales, además de ser simétricos.
 - Ecuaciones de regresión: Los datos utilizados en formular la ecuación de regresión, provienen de medirlos directamente en cadáveres de diferentes muestras de población. Este estudio sobre los cadáveres implica que éstos sean seccionados cuidadosamente, pesados, balanceados y medidos antropométricamente.

Es necesario escoger un método que sea personalizado para cada sujeto, que requiera poco tiempo, bajo costo, seguro y debe ser razonablemente exacto para estimar los parámetros inerciales de los segmentos.

Por lo anterior, y para este proyecto, es beneficioso utilizar métodos de regresión lineal, para estimar dichos parámetros inerciales, utilizando como variables, las medidas realizadas a los sujetos de estudio. Se ha demostrado que el resultado de estos métodos de regresión, son datos bastante precisos y confiables.

Tal como se muestra en la *Figura 28*, son 9 medidas por cada pierna, además del peso y la medida de la distancia anterior superior, o entre las espinas ilíacas ASIS por sus siglas en inglés.

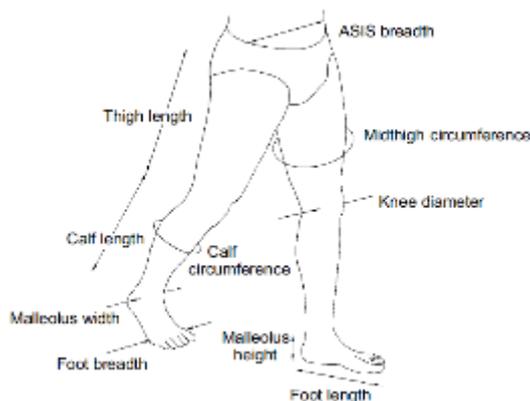


Figura 28 Medidas antropométricas de las extremidades inferiores, que son requeridas para predecir los parámetros de los segmentos del cuerpo, tales como masas y momentos de inercia. (Fuente: Dynamics of Human Gait)

A continuación, se presenta las medidas antropométricas necesarias para realizar las estimaciones antes mencionadas (Pàmies Vilà, 2012).

Tabla 3 Tabla de medidas antropométricas

M ₁	Masa total del cuerpo
M ₂	Ancho de cintura
M ₃	Longitud de muslo derecho
M ₄	Longitud de muslo izquierdo
M ₅	Perímetro del muslo derecho
M ₆	Perímetro del muslo izquierdo
M ₇	Longitud de la pantorrilla derecha
M ₈	Longitud de la pantorrilla izquierda
M ₉	Perímetro de la pantorrilla derecha
M ₁₀	Perímetro de la pantorrilla izquierda
M ₁₁	Diámetro de la rodilla derecha
M ₁₂	Diámetro de la rodilla izquierda
M ₁₃	Longitud del pie derecho
M ₁₄	Longitud del pie izquierdo
M ₁₅	Altura del tobillo derecho
M ₁₆	Altura del tobillo izquierdo
M ₁₇	Ancho del tobillo derecho
M ₁₈	Ancho del tobillo izquierdo
M ₁₉	Ancho del pie derecho
M ₂₀	Ancho del pie izquierdo

Estas mediciones pueden ser realizadas sin muchas complicaciones, ya que solo es necesario un centímetro y un pie de rey, para medir los diámetros respectivos.

Masa de cada segmento inferior

Se realiza la estimación de las masas de los segmentos de interés, para esto es necesario la masa total y las dimensiones de éstos segmentos. Una vez obtenido las medidas de un segmento determinado, se empleará la siguiente fórmula que implica la masa, el volumen y coeficientes de correlación (Vaughan, Davis, & O'Connor, 1999):

$$\text{Masa del segmento} = C1.\text{masa total del cuerpo} + C2.\text{longitud}^3 + C3\dots(26)$$

Donde C1, C2 y C3 son los coeficientes de correlación. Para nuestro propósito y luego basándonos en otras investigaciones, se optó por escoger el modelamiento de muslo y pantorrillas representadas con cilindros, y el del pie, como una pirámide.

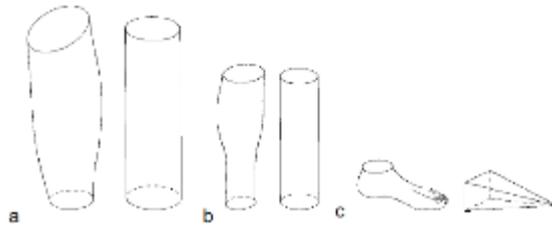


Figura 29 Segmentos corporales de extremidades inferiores y su comparación geométrica para el modelamiento: (a) muslo; (b) pantorrilla; (c) pie. (Fuente: Dynamics of Human Gait, Vaughan)

$$\text{Masa del cilindro} = \frac{\text{densidad}}{4\pi} (\text{longitud})(\text{circunferencia})^2$$

$$\text{Masa de una pirámide} = \frac{1}{3} (\text{densidad})(\text{ancho})(\text{largo})(\text{alto})$$

Los coeficientes de la ecuación 26 se obtienen luego de un análisis de correlación con los datos que fueron resultado de estudios forenses sobre los segmentos que queremos analizar, es decir los miembros inferiores. En consecuencia, se expresan las ecuaciones para el cálculo de la masa del muslo, pantorrilla y pie:

$$\text{Masa del muslo Derecho} = (0.1032)(M_1) + (12.76)(M_3)(M_5)^2 + (-1.023)$$

$$\text{Masa del muslo Izquierdo} = (0.1032)(M_1) + (12.76)(M_4)(M_6)^2 + (-1.023)$$

$$\text{Masa de la pantorrilla derecha} = (0.0226)(M_1) + (31.33)(M_7)(M_9)^2 + (0.016)$$

$$\text{Masa de la pantorrilla izquierda} = (0.0226)(M_1) + (31.33)(M_8)(M_{10})^2 + (0.016)$$

$$\text{Masa del pie derecho} = (0.0083)(M_1) + (254.5)(M_{17})(M_{15})(M_{13}) + (-0.065)$$

$$\text{Masa del pie izquierdo} = (0.0083)(M_1) + (254.5)(M_{18})(M_{16})(M_{14}) + (-0.065)$$

Desde M1 hasta M18 son las medidas necesarias e indicadas en la Tabla de medidas antropométricas.

Momento de inercia de cada segmento inferior

El momento de inercia es la medida de resistencia al movimiento angular, y su unidad es $\text{kg}\cdot\text{m}^2$. En consecuencia, para el cálculo del momento de inercia es lógico relacionarlo la masa del segmento y las dimensiones al cuadrado.

Se plantea que la ecuación de regresión lineal que puede expresar el momento de inercia sería la siguiente (Vaughan, Davis, & O'Connor, 1999):

$$\text{Momento de inercia de un segmento} = C4(\text{masa total})(\text{longitud})^2 + C5\dots(27)$$

Donde C4 y C5 son los coeficientes de regresión. De acuerdo al planteamiento del cálculo de la masa de los segmentos, seguiremos usando la aproximación de los muslos, pantorrillas y pie como cilindros para los dos primeros y de una pirámide recta para el pie.

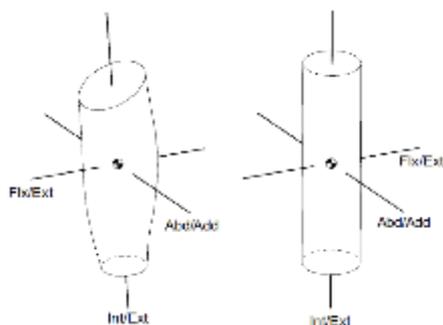


Figura 30 Ejes principales de giro del muslo. (Fuente: Dynamics of Human Gait, Vaughan)

En la *Figura 30* se muestra los posibles ejes en los cuales los segmentos pueden realizar sus movimientos.

Usando la definición matemática para determinar los momentos de inercia, por ejemplo de un cilindro de acuerdo a sus ejes de giro, son:

- *Momento de inercia de un cilindro con respecto al eje flexión/extensión =*

$$\frac{1}{12}(\text{masa})[(\text{longitud})^2 + 0.076(\text{circunferencia})^2]$$

- *Momento de inercia de un cilindro con respecto al eje abducción/aducción =*

$$\frac{1}{12}(\text{masa})[(\text{longitud})^2 + 0.076(\text{circunferencia})^2]$$

- *Momento de inercia de un cilindro con respecto al eje interno/externo =*

$$\frac{1}{8\pi^2}(\text{masa})(\text{circunferencia})^2$$

Para esta investigación, nos centraremos en los momentos de inercia con respecto al eje de Flexión y Extensión.

Luego de realizar el proceso de resolución para encontrar los coeficientes de correlación, se tiene finalmente las ecuaciones por segmento:

$$I_{Flex / Ext_muslo\ derecho} = 0.00762(M_1)[(M_3)^2 + 0.076(M_5)^2] + 0.01153$$

$$I_{Flex / Ext_muslo\ izquierdo} = 0.00762(M_1)[(M_4)^2 + 0.076(M_6)^2] + 0.01153$$

$$I_{Flex / Ext_pantorrilla\ derecha} = 0.00247(M_1)[(M_7)^2 + 0.076(M_9)^2] + 0.00511$$

$$I_{Flex / Ext_pantorrilla\ izquierda} = 0.00247(M_1)[(M_8)^2 + 0.076(M_{10})^2] + 0.00511$$

$$I_{Flex / Ext_pie\ derecho} = 0.00023(M_1)[4(M_{15})^2 + 3(M_{13})^2] + 0.00022$$

$$I_{Flex / Ext_pie\ izquierdo} = 0.00023(M_1)[4(M_{16})^2 + 3(M_{14})^2] + 0.00022$$

Desde M1 hasta M16 son las medidas necesarias e indicadas en la Tabla de medidas antropométricas.

3.2 CÁLCULO DE LA ORIENTACIÓN DE CADA SEGMENTO

En esta parte de la investigación, determinaremos la mejor manera de ubicar los marcadores (markers), los cuales nos permitirán establecer el centro de gravedad de los segmentos inferiores así como la orientación de los vectores que modelan los segmentos. A continuación, se presenta la disposición de los marcadores utilizados para terminar los segmentos en una captura de movimiento.

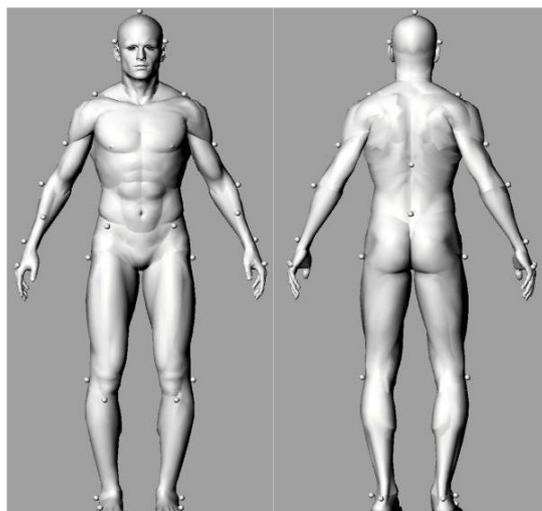


Figura 31 Posición de los marcadores utilizados en la captura de movimiento

Tabla 4 Mapeado de markers*

Marker	Posición	Marker	Posición
01	Exterior pie derecho	18	Codo derecho
02	Talón pie derecho	19	Antebrazo derecho
03	Tobillo derecho	20	Muñeca derecha
04	Tibia derecha	21	Nudillo meñique derecho
05	Exterior rodilla derecha	22	Nudillo índice derecho
06	Cabeza fémur derecho	23	Hombro izquierdo arriba
07	Hueso cadera derecha	24	Hombro izquierdo lateral
08	Exterior pie izquierdo	25	Codo izquierdo
09	Talón pie izquierdo	26	Antebrazo izquierdo
10	Tobillo izquierdo	27	Muñeca izquierda
11	Tibia izquierda	28	Nudillo meñique izquierdo
12	Exterior rodilla izquierda	29	Nudillo índice izquierdo
13	Cabeza fémur izquierdo	30	Espalda
14	Hueso cadera izquierda	31	Oreja derecha
15	Sacro	32	Cabeza arriba
16	Hombro derecho arriba	33	Oreja izquierda
17	Hombro derecho lateral		

*Según sistema usado en la sala de capturas del laboratorio de Ingeniería Mecánica de la Universidad A Coruña

La finalidad de una buena disposición de los marcadores es definir las articulaciones (joints) entre segmentos, en otras palabras las articulaciones del cuerpo. La información de las posiciones de los marcadores será utilizada para definir la posición, por ejemplo, del hombro, codo, muñeca, etc.

Se comienza por construir unos ejes ortogonales a partir de la posición de los marcadores. De esta manera podemos orientarnos y situar el punto en donde se ubica la articulación. Por ejemplo, para el pie derecho se tiene:

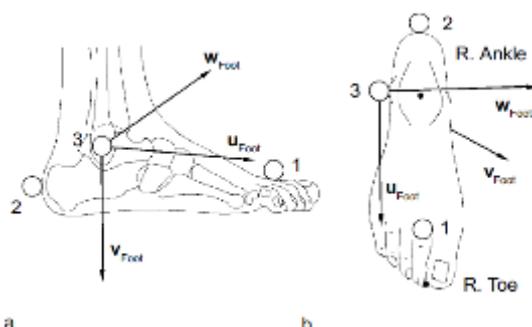


Figura 32 Los tres marcadores utilizados para determinar la posición del tobillo. (a) vista lateral del pie. (b) vista vertical del pie. (Fuente: Dynamics of Human Gait, Vaughan)

En la *Figura 32* se muestra el eje definido para establecer la posición del tobillo.

$$\hat{u}_{pie\ der.} = \frac{\vec{p}_1 - \vec{p}_2}{|\vec{p}_1 - \vec{p}_2|}$$

$$\hat{w}_{pie\ der.} = \frac{(\vec{p}_1 - \vec{p}_3) \times (\vec{p}_2 - \vec{p}_3)}{|(\vec{p}_1 - \vec{p}_3) \times (\vec{p}_2 - \vec{p}_3)|}$$

$$\hat{v}_{pie\ der.} = \hat{w}_{pie\ der.} \times \hat{u}_{pie\ der.}$$

Basados en los ejes ortogonales calculados, se puede establecer la posición del tobillo y del dedo del pie:

$$\vec{p}_{tob.der.} = \vec{p}_3 + 0.016(M_{13})\hat{u}_{pie\ der.} + 0.392(M_{15})\hat{v}_{pie\ der.} + 0.478(M_{17})\hat{w}_{pie\ der.}$$

$$\vec{p}_{dedo\ pie.der.} = \vec{p}_3 + 0.742(M_{13})\hat{u}_{pie\ der.} + 1.074(M_{15})\hat{v}_{pie\ der.} - 0.187(M_{19})\hat{w}_{pie\ der.}$$

Estimación del centro de gravedad

Para realizar la estimación del centro de gravedad de cada segmento es necesario tener como información la ubicación de las articulaciones anterior y posterior.

Estas estimaciones son proporcionales a la longitud del segmento, por ejemplo:

$$ratio_{CG_muslo\ der.} = 0.39$$

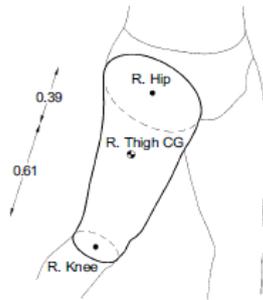


Figura 33 Estimación por ratio del centro de gravedad en el muslo. (Fuente: Dynamics of Human Gait, Vaughan)

$$\vec{p}_{CG_muslo\ der.} = \vec{p}_{cadera} + 0.39(\vec{p}_{rodilla} - \vec{p}_{cadera})$$

En resumen, se tiene los ratios para los miembros inferiores para las estimaciones de éstos:

$$ratio_{CG_muslo\ der.} = 0.39$$

$$ratio_{CG_pant.der.} = 0.42$$

$$ratio_{CG_pie\ der.} = 0.44$$

$$ratio_{CG_muslo\ izq.} = 0.39$$

$$ratio_{CG_pant.izq.} = 0.42$$

$$ratio_{CG_pie\ izq.} = 0.44$$

Cuyas ecuaciones para las estimaciones de los CG se determinan por las articulaciones distales y proximales:

$$\vec{p}_{CG_muslo} = \vec{p}_{cadera} + 0.39(\vec{p}_{rodilla} - \vec{p}_{cadera})$$

$$\vec{p}_{CG_pant.} = \vec{p}_{rodilla} + 0.42(\vec{p}_{tobillo} - \vec{p}_{rodilla})$$

$$\vec{p}_{CG_pie} = \vec{p}_{talón} + 0.44(\vec{p}_{dedo\ pie} - \vec{p}_{talón})$$

Hasta esta parte de la investigación, se ha mencionado sobre las estimaciones para la parte inferior del cuerpo.

Con relación a las mediciones de los segmentos superiores del cuerpo, se realizará una estimación de acuerdo a un modelo estándar, el cual representa el percentil 50 de la población, en donde se incluye unos coeficientes de escala de longitud, masa y momentos de inercia, tal como muestra la *Tabla 5*.

Tabla 5 Body Segments Parameters ("Kinematic Data Consistency in the Inverser Dynamic Analysis of Biomechanical Systems", M.P.T. Silva and J.A.C Ambrósio)

Elemento	Longitud (m)	dCG (m)	Masa (kg)	I _{flx/ext} (10 ⁻² kg.m ²)	I _{abd/add} (10 ⁻² kg.m ²)	I _{int/ext} (10 ⁻² kg.m ²)
muslo derecho	0.434	-	9.843	15.94	9.867	1.435
muslo izquierdo	0.434	-	9.843	15.94	9.867	1.435
pantorrilla derecha	0.439	-	3.626	3.83	3.14	1.086
pantorrilla izquierda	0.439	-	3.626	3.83	3.14	1.086
pie derecho	0.271	-	1.182	0.128	0.129	2.569
pie izquierdo	0.271	-	1.182	0.128	0.129	2.569
torso inferior	0.275	0.064	14.2	13.45	26.22	26.22
torso superior	0.294	0.101	24.95	37.19	24.64	19.21
cuello	0.122	0.061	1.061	0.215	0.268	0.268
cabeza	0.128	0.02	4.241	2.249	2.453	2.034
brazo derecho	0.295	0.153	1.992	1.356	1.492	0.248
antebrazo derecho	0.25	0.123	1.892	0.964	1.24	0.298
mano derecha	0.185	0.093	0.489	0.146	0.148	0.067
brazo izquierdo	0.295	0.153	1.992	1.356	1.492	0.248
antebrazo izquierdo	0.25	0.123	1.892	0.964	1.24	0.298
mano izquierda	0.185	0.093	0.489	0.146	0.148	0.067

Para el cálculo de la masa, centro de gravedad y momento de inercia de cada elemento se multiplica el valor correspondiente al modelo estándar por el valor del factor de escala correspondiente.

$$X_m = \frac{m}{m_{st}} \quad X_L = \frac{L}{L_{st}} \quad X_I = X_m \cdot X_L^2$$

Donde X_m , X_L , X_I representan los factores de escala de masa, longitud y momento de inercia respectivamente.

m_{st} , L_{st} representan las mediciones de masa y longitud del modelo estándar (percentil 50).

m, L representan las mediciones de masa y longitud del sujeto.

Dado que se utilizan diferentes métodos para calcular los parámetros de las partes superior e inferior del cuerpo, hay que asegurar que la masa total sea la correcta.

Para ello, se calcula el factor de escala de masas para los segmentos superiores como cociente entre la suma de sus masas en el modelo estándar, y la masa que queda al restar a la total la obtenida a partir de las ecuaciones de regresión para las piernas.

El factor de escala de longitud lo obtenemos como el cociente entre la longitud del muslo del sujeto y el modelo estándar.

El factor de escala de momentos de inercia es el resultado del producto del factor de escala de masa por el factor de escala de longitud al cuadrado, por análisis dimensional.

Para ubicar un sólido en el espacio, son necesario seis coordenadas independientes para definir su posición y orientación. En la determinación de la orientación se emplea un sistema de coordenadas locales que expresan la posición en relación a un sistema de referencia global.

Cuando se calcula la posición y orientación de los segmentos corporales, es necesario medir las posiciones de al menos tres puntos no alineados de dichos segmentos. La ubicación del origen de cada sistema local es en el centro de gravedad de cada segmento.

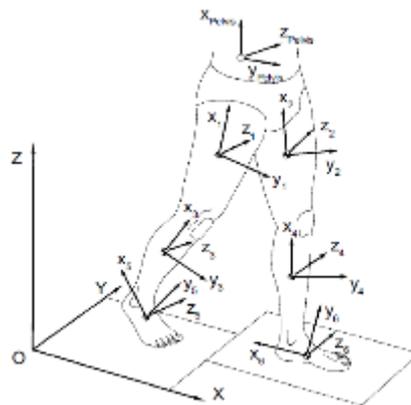


Figura 34 Orientación de los ejes locales ubicados en cada segmento, parte inferior; 1, muslo derecho; 2, muslo izquierdo; 3, pantorrilla derecha; 4, pantorrilla izquierda; 5, pie derecho; 6, pie izquierdo. (Fuente: *Dynamics of Human Gait, Vaughan*)

Por convención, el eje X de los ejes locales de cada segmento, se orienta en la dirección proximal-distal del elemento; y el eje Z en dirección medio-lateral.

Para el caso de los elementos de la pierna derecha, Muslo:

El eje X se orienta en la línea de acción del vector que forma los puntos de la rodilla y la cadera. El plano XZ está formado por la articulación de la cadera, el marcador del muslo (Marker 06, ver tabla de Mapeo de Markers) y la articulación de la rodilla. Por lo tanto, el

eje Y es perpendicular al plano XZ; el eje Z resulta del producto vectorial de ambos. En resumen:

$$\mathbf{i}_1 = \frac{(\vec{p}_{cadera\ der.} - \vec{p}_{rodilla\ der.})}{|\vec{p}_{cadera\ der.} - \vec{p}_{rodilla\ der.}|}$$

$$\mathbf{j}_1 = \frac{(\mathbf{Marker}_6 - \vec{p}_{cadera\ der.}) \times (\vec{p}_{rodilla\ der.} - \vec{p}_{cadera\ der.})}{|(\mathbf{Marker}_6 - \vec{p}_{cadera\ der.}) \times (\vec{p}_{rodilla\ der.} - \vec{p}_{cadera\ der.})|}$$

$$\mathbf{k}_1 = \mathbf{i}_1 \times \mathbf{j}_1$$

Las ecuaciones y estimaciones para determinar las posiciones de las articulaciones, tanto inferiores y superiores, están dentro del software utilizado en el sistema de capturas del laboratorio de Ingeniería Mecánica de la Universidad A Coruña. Por lo tanto, lo que se obtendrá luego de la captura de movimiento, será las posiciones de cada segmento con respecto a los ejes locales y también las coordenadas de las articulaciones con respecto a un eje global, las masas y momentos de inercia de los segmentos. Además de las matrices de transformación de los ejes locales con respecto al eje global. Toda esta información se obtiene con formato .mat el cual es compatible para ser usado en MATLAB.

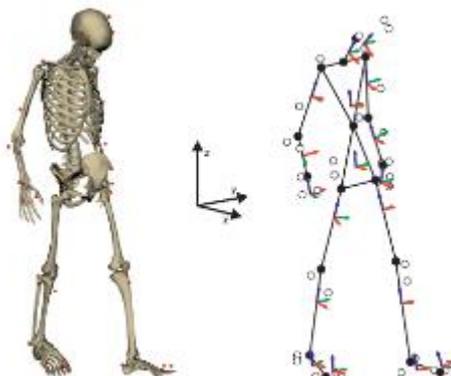


Figura 35 Ubicación de marcadores y orientación de segmentos, del sistema de capturas. (Fuente: Lugris, U., Carlín, J., Luaces, A., Cuadrado, J. (2013). *Gait Analysis System for Spinal Cord Injured Subjects Assisted by Active Orthoses and Crutches*)

Ponemos el siguiente ejemplo, la ecuación 28 establece la posición del centro de gravedad de la cabeza. Para definir esta ecuación, es necesario tener la posición global de la articulación proximal a la ubicación del punto que queremos definir, además de tener la matriz de giro que permita realizar la transformación del vector posición local, del eje local al eje global. De esta manera se obtendrá el nuevo vector de posición global del centro de gravedad.

$$\vec{r}_{CDG\ cabeza}^g = \vec{r}_{Base\ cabeza}^g + A_{cabeza} \cdot \vec{r}_{CDG\ cabeza}^l \dots (28)$$

Para un modelamiento correcto de la cabeza, manos y dedos de los pies, se expresa estas partes con el centro de gravedad correspondiente a éstas. De acuerdo a estas consideraciones, se tiene 23 segmentos que unen las 22 articulaciones establecidas. Al

realizar el modelamiento del cuerpo humano como un conjunto de elementos rígidos interconectados entre sí, estamos adoptando el enfoque estándar de la dinámica de sistemas multicuerpo.

Seguimos bajo el supuesto que son segmentos rígidos cuyas dimensiones y longitud no varían durante el movimiento, aunque en la realidad estos segmentos no son totalmente rígidos, por lo que se trata de una simplificación.

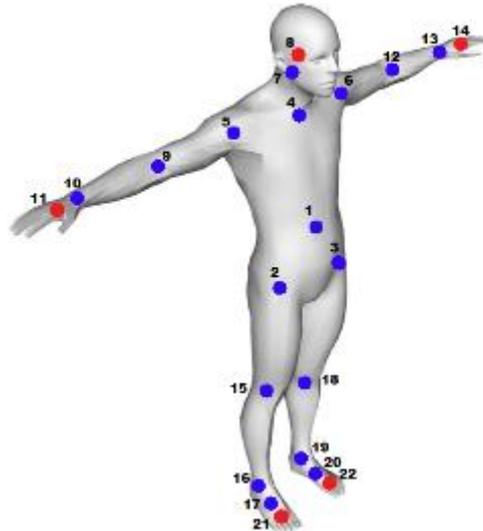


Figura 36 Ubicación de articulaciones (puntos azules) y centros de gravedad de cabeza, manos y dedos de los pies (puntos rojos).

Tabla 6 Listado de puntos para el modelamiento en Matlab

PM	Punto de modelo de Matlab
01	Articulación lumbar
02	Cadera derecha
03	Cadera izquierda
04	Base del cuello
05	Hombro derecho
06	Hombro izquierdo
07	Base de la cabeza
08	CGD de la cabeza
09	Codo derecho
10	Muñeca derecha
11	CDG de la mano derecha
PM	Punto de modelo de Matlab
12	Codo izquierdo
13	Muñeca izquierda
14	CDG de la mano izquierda
15	Rodilla derecha
16	Tobillo derecho

17	Articulación pie derecho
21	CGD punta pie derecho
18	Rodilla izquierda
19	Tobillo izquierdo
20	Articulación pie izquierdo
22	CGD punta pie izquierdo

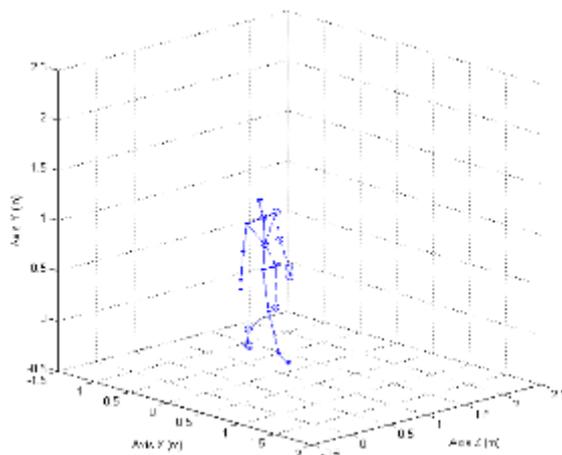


Figura 37 Modelo de segmentos y articulaciones en MATLAB.

Proyección en el plano sagital

Desde esta parte de la investigación, se trabajará en el plano sagital, esto implica la proyección de los puntos que previamente habíamos calculado por medio de las estimaciones y aproximaciones antes explicadas.

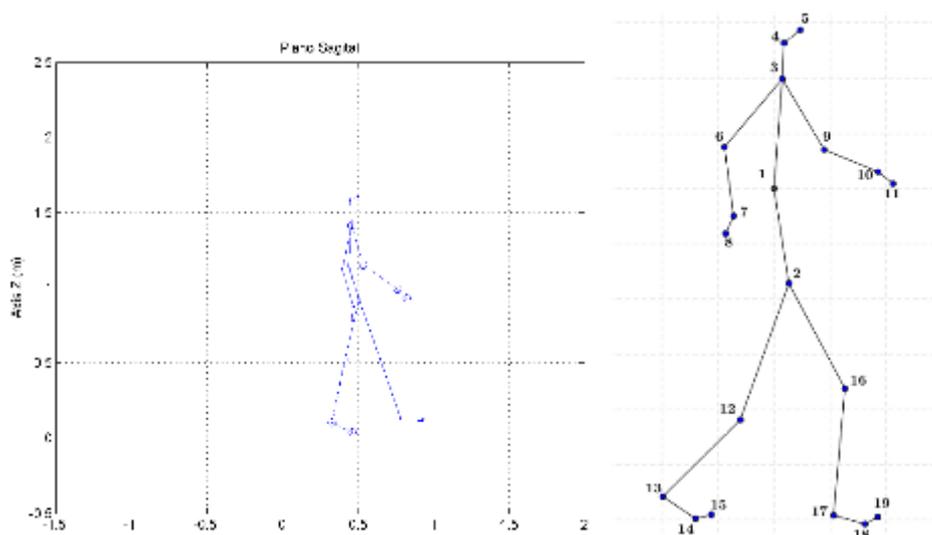


Figura 38 Modelo 2D en MATLAB (izquierda). Modelamiento de puntos y segmentos (derecha)

Las implicancias de este procedimiento son que los puntos 2 y 3, ver Figura 36, concurren en un solo punto. Así como los puntos 4, 5 y 6. Como resultado, el modelo final para posteriores análisis, es el de la Figura 38.

Consistencia cinemática

Cuando se realiza la captura del movimiento, en el resultado de estos datos es la composición de varias señales: el movimiento del sistema esquelético, el movimiento del marcador respecto al sistema esquelético y el ruido producido por el sistema de captura.

Esto sucede porque los marcadores son adheridos en un traje especial, como un traje de buceo. Existe un movimiento relativo entre el marcador y el sistema esquelético, este movimiento relativo será mayor a medida que la aceleración del movimiento se incremente. La causa de este problema es la deformación del traje, próximas a las articulaciones debido a efectos inerciales y contracciones musculares. Este tipo de movimiento produce un error al realizar el análisis dinámico inverso (ADI). Lo cual indica que debemos solucionar el problema de la consistencia cinemática que acarrea el movimiento relativo, ya que es la fuente de error más críticas en el análisis del movimiento humano. Cuando se realiza la captura de movimiento, los datos son ordenados de acuerdo a modelamiento planteado para esta investigación, posteriormente se proyectan los puntos en el plano sagital.

Se corrige las señales que se han obtenido en la captura de movimiento, y de acuerdo a las restricciones que tenemos, en este caso las restricciones de longitud de cada segmento corporal. Con esto se logra que las dimensiones del sistema esquelético sean constantes. Para ello, se minimiza de forma iterativa la suma de distancias entre las coordenadas medidas por el sistema de captura de movimiento y aquellas que cumplen las ecuaciones de restricción de la cadena cinemática.

Entonces, se incurre en resolver el siguiente problema de minimización:

$$\min_{\mathbf{q}} f(\mathbf{q}) = \frac{1}{2}(\mathbf{q}^* - \mathbf{q}_0)^T \mathbf{W}(\mathbf{q}^* - \mathbf{q}_0)$$

sujeto a $\Phi = 0$

donde \mathbf{q}_0 el vector que está formado por las coordenadas de las posiciones inicial de las articulaciones, \mathbf{q}^* el vector de las nuevas posiciones de acuerdo a cada iteración, \mathbf{W} es la matriz de ponderación la cual determina qué puntos no van a variar mucho con respecto a sus posiciones iniciales y Φ es el vector de restricciones cinemáticas no lineales.

$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \end{bmatrix} \quad \Phi = \begin{bmatrix} (x_1 - x_2)^2 + (y_1 - y_2)^2 - L_1^2 \\ (x_2 - x_3)^2 + (y_2 - y_3)^2 - L_2^2 \\ \vdots \end{bmatrix}$$

L_1 es la longitud entre las posiciones de las articulaciones 1 y 2, análogamente L_2 , longitud entre articulaciones 2 y 3.

Este problema de minimización se resuelve con el proceso iterativo de LaGrange aumentado:

$$(W + \Phi_q^T \alpha \Phi_q) \Delta q_{i+1} = -W(q_i - q_0) - \Phi_q^T (\alpha \Phi + \lambda_i)$$

$$\lambda_{i+1} = \lambda_i + \alpha \Phi \quad i = 1, 2, \dots$$

donde $\Delta q_{i+1} = q_{i+1} - q_i$, Φ_q es la matriz jacobiana de las restricciones cinemáticas del vector Φ , λ es el vector de los multiplicadores de LaGrange y α es el factor de penalidad, cuyo valor puede ser del orden entre 10^7 - 10^9 y dependiendo de este valor, afecta a la velocidad de la convergencia.

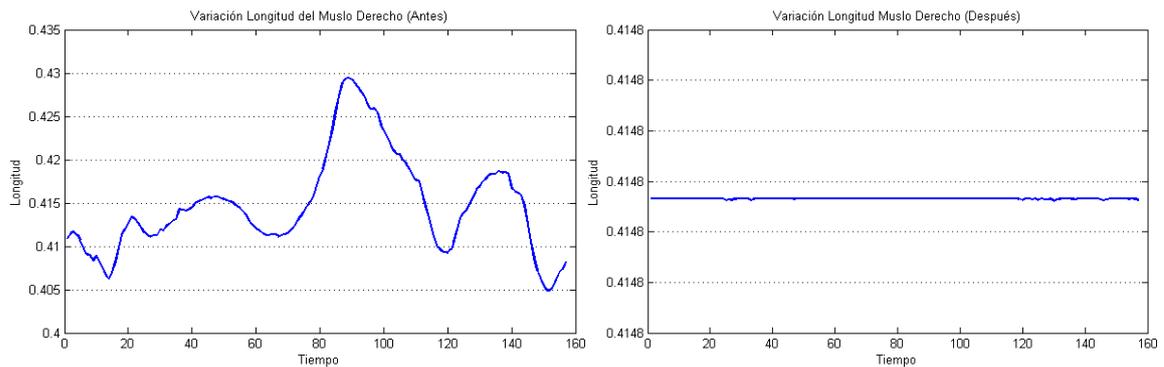


Figura 39 Gráficas de la variación de la longitud del muslo antes y después de resolver el problema de minimización.

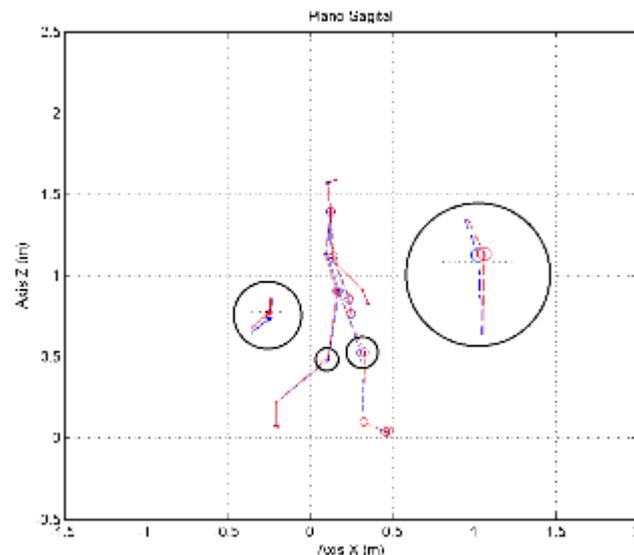


Figura 40 Corrección de posiciones de articulaciones. Azul: Antes de consistencia cinemática. Rojo: Después de consistencia cinemática.

En la Figura 40, se muestra el resultado antes y después del análisis cinemático, se puede ver que los puntos que no varían son los tobillos, asumiendo que durante toda la captura de movimiento, la ubicación de estas articulaciones son las obtenidas por las cámaras infrarrojas.

3.3 CÁLCULO DE LOS ÁNGULOS: CINEMÁTICA INVERSA

Luego de resolver el problema de posiciones de las articulaciones para que se cumplan las restricciones de longitud, es posible determinar la orientación de los segmentos y el giro que se ha producido entre dos elementos, para esto se tiene en cuenta del análisis en 2D, en el eje de flexión-extensión.

Para cada articulación se tiene dos sistemas de referencias: uno en el segmento proximal y otro en el distal. Por ejemplo, para la articulación de la rodilla, el segmento proximal es el muslo y el distal, la pantorrilla). Los ángulos articulares se definen como una rotación del segmento distal con respecto al segmento proximal, tomando como referencia el ángulo que forma la horizontal con el segmento distal.

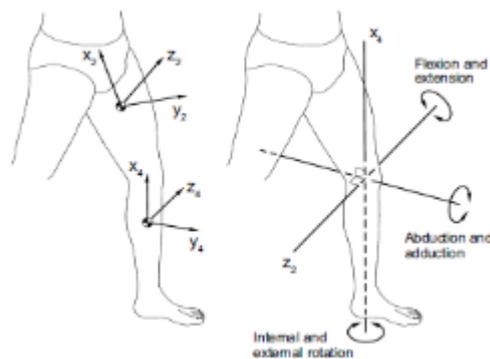


Figura 41 Ejes de rotación de la rodilla. A la izquierda, los ejes locales de los segmentos proximal y distal.

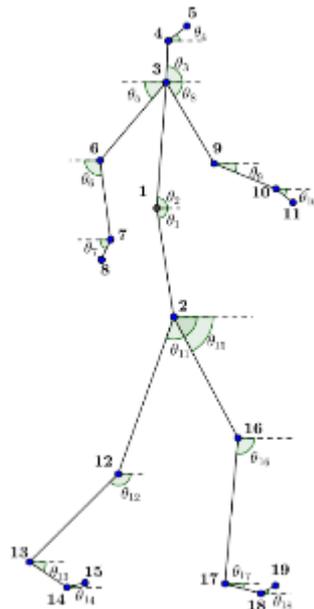


Figura 42 Modelamiento de ángulos de las articulaciones.

Con este planteamiento del nuevo modelo que incluye los ángulos, resolveremos la cinemática inversa, que implica hallar los ángulos que cumplan con las nuevas restricciones que se le agregará a nuestro vector de restricciones no lineales Φ , de antemano, también se modificará nuestro vector de variables q_0 ya que nuestras nuevas incógnitas, los ángulos, están ubicadas al final de las coordenadas antes halladas.

$$q_0^c = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ \theta_1 \\ \theta_2 \\ \vdots \end{bmatrix} \quad \Phi^c = \begin{bmatrix} (x_1 - x_2)^2 + (y_1 - y_2)^2 - L_1^2 \\ (x_2 - x_3)^2 + (y_2 - y_3)^2 - L_2^2 \\ \vdots \\ x_1 - x_2 + \cos(\theta_1) \cdot L_1 \\ x_1 - x_3 + \cos(\theta_2) \cdot L_2 \\ \vdots \end{bmatrix}$$

Cuando se redefine Φ , se considera las ecuaciones respectivas para la variación del ángulo en el cuadrante adecuado, es decir, que se usa la ecuación del seno o coseno, y en el caso en que la variación se realice en ambos cuadrantes, se usa una ecuación adicional que es redundante pero que se ajusta con la información necesaria para calcular los ángulos.

Una vez más se incurre en el problema de minimización, realizando iteraciones a partir de valores aproximados de los ángulos que se calculan. Para el planteamiento del problema, utilizamos la ecuación 20,

$$\Phi_q(q_0^c)(q - q_0^c) = -\Phi(q_0^c)$$

Los valores iniciales de los ángulos deben ser valores dentro de los rangos teóricos establecidos, esto asegura que la solución converja.

La tolerancia que parametriza las iteraciones para encontrar una solución óptima es del orden 10^{-10} , comparando este valor con la norma o módulo del vector Φ^c .

Luego de hallar los ángulos que cumplan con las restricciones, se realiza un filtrado de los datos, el cual elimina el ruido generado por la captura de datos. La finalidad de este procedimiento es que el movimiento de las posiciones de articulaciones sea más natural. Se considera que este filtro solo se aplicará a las coordenadas independientes del sistema de sólidos rígidos: al punto 1 del modelo, articulación lumbar, y a todos los ángulos calculados anteriormente.

Por último, se realiza un análisis de consistencia cinemática con los valores filtrados para conseguir minimizar los errores sistemáticos del sistema de captura.

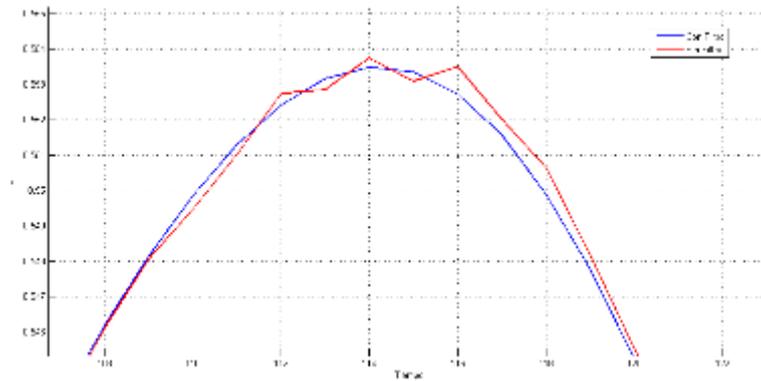


Figura 43 Filtro aplicado a una señal de captura de movimiento.

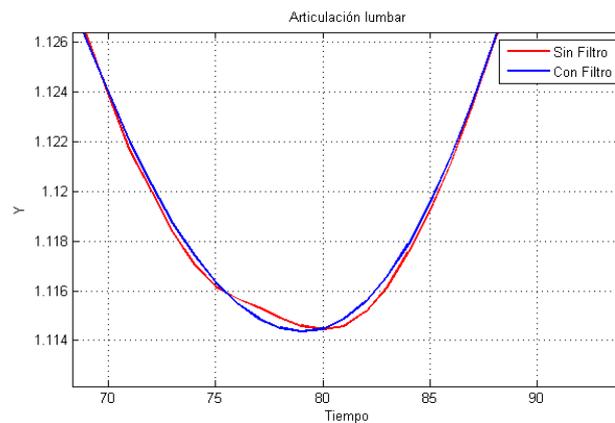


Figura 44 Coordenada Y de la articulación lumbar antes y después del filtrado y consistencia cinemática.

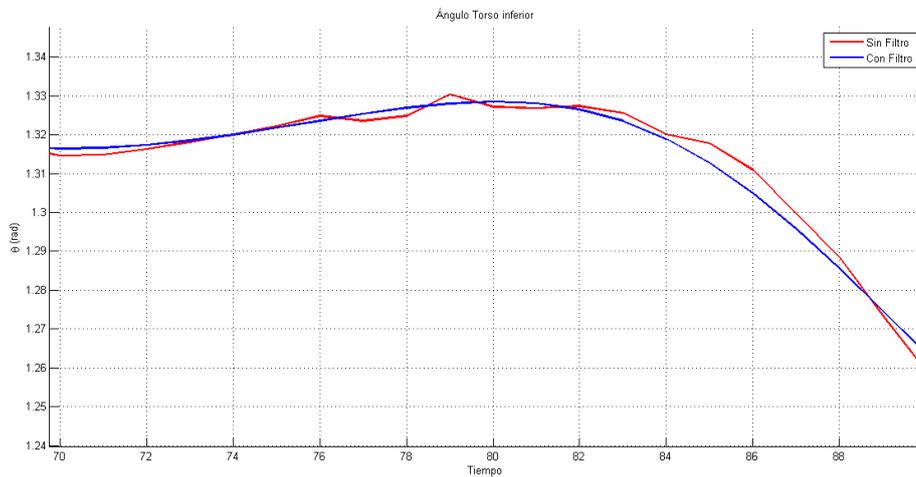


Figura 45 Variación del ángulo del torso inferior antes y después del filtrado y consistencia cinemática.

3.4 VELOCIDADES Y ACELERACIONES: DIFERENCIACIÓN NUMÉRICA

Para el cálculo de velocidades y aceleraciones de puntos determinados, se calcula por diferenciación numérica mediante el método de las diferencias finitas. Para medir la

velocidad de una posición determinada n se calcula la variación entre un instante anterior y posterior y se dividen entre el tiempo transcurrido entre ambos.

$$\frac{dx}{dt} = x' = \dot{x} = \frac{x_{n+1} - x_{n-1}}{\Delta t}$$

El cálculo de la aceleración no es más que la variación en velocidad entre un instante anterior y posterior, en consecuencia tenemos la siguiente expresión:

$$\frac{d^2x}{dt^2} = x'' = \ddot{x} = \frac{\dot{x}_{n+1} - \dot{x}_{n-1}}{\Delta t}$$

Sin embargo, el cálculo de la velocidad y aceleración no se realizan para todas las coordenadas. Se realiza la primera y segunda derivada de los datos filtrados correspondientes a las coordenadas independientes, es decir, a la posición de la articulación lumbar y los ángulos. Ya que luego se resolverá el problema de la velocidad y aceleración explicado en el acápite 2.3.3.2 y 2.3.3.3.

Utilizando la ecuación 22 se resuelve el problema de la velocidad, acomodando de acuerdo a los datos de velocidades que se conocen, queda resolver por iteraciones las velocidades de las otras variables en cada instante.

$$\Phi_q^c(q)\dot{q}^c = 0$$

$$\dot{q}_0^c = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \\ \vdots \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \end{bmatrix}$$

$$\Phi_q^c(q) = \begin{bmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) & -2(x_1 - x_2) & -2(y_1 - y_2) & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 2(x_1 - x_3) & 2(y_1 - y_3) & 0 & 0 & -2(x_1 - x_3) & -2(y_1 - y_3) & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots \\ 1 & 0 & -1 & 0 & 0 & 0 & \dots & -L_1 \text{sen}(\theta_1) & \dots & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & \dots & \dots & \dots & -L_2 \text{sen}(\theta_2) & \dots \\ \vdots & \vdots \end{bmatrix}$$

$$\Phi_q^c(q)\dot{q}^c = \begin{bmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) & -2(x_1 - x_2) & -2(y_1 - y_2) & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 2(x_1 - x_3) & 2(y_1 - y_3) & 0 & 0 & -2(x_1 - x_3) & -2(y_1 - y_3) & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots \\ 1 & 0 & -1 & 0 & 0 & 0 & \dots & -L_1 \text{sen}(\theta_1) & \dots & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & \dots & \dots & \dots & -L_2 \text{sen}(\theta_2) & \dots \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \\ \vdots \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \end{bmatrix}$$

Arreglando los términos para calcular las variables que necesitamos:

$$\begin{bmatrix} -2(x_1 - x_2) & -2(y_1 - y_2) & 0 & 0 & \dots \\ 0 & 0 & -2(x_1 - x_3) & -2(y_1 - y_3) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \\ \vdots \end{bmatrix} = -\dot{x}_1 \begin{bmatrix} 2(x_1 - x_2) \\ 2(x_1 - x_3) \\ \vdots \\ 1 \\ 1 \\ \vdots \end{bmatrix} - \dot{y}_1 \begin{bmatrix} 2(y_1 - y_2) \\ 2(y_1 - y_3) \\ \vdots \\ 0 \\ 0 \\ \vdots \end{bmatrix} - \dot{\theta}_1 \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -L_1 \text{sen}(\theta_1) \\ 0 \\ \vdots \end{bmatrix} - \dot{\theta}_2 \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \dots \\ -L_2 \text{sen}(\theta_2) \\ \vdots \end{bmatrix} \dots$$

Resolviendo esta ecuación matricial cuya matriz de variables es el vector que contiene los \dot{x} y \dot{y} . Mediante iteraciones podemos determinar el valor de estas variables en cada instante del tiempo.

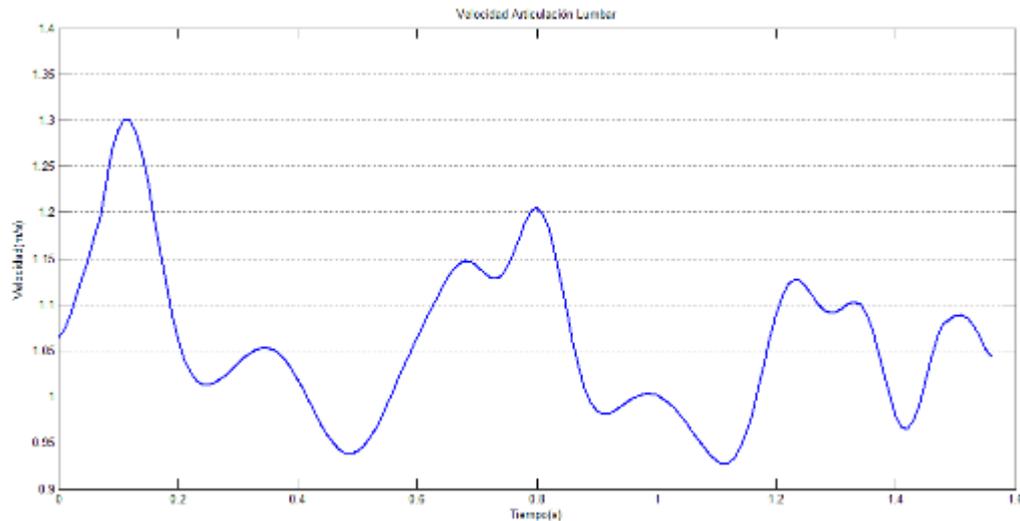


Figura 46 Velocidad de la articulación lumbar en una marcha normal.

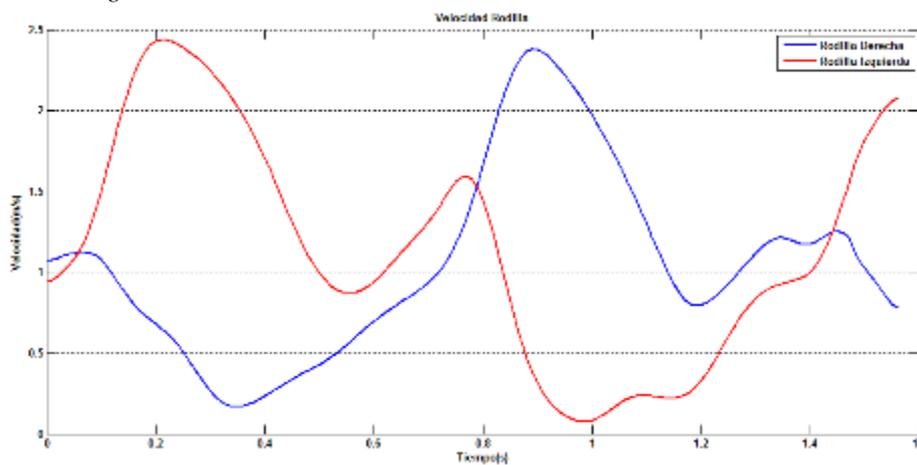


Figura 47 Velocidad de ambas rodillas en una marcha normal.

Para resolver el problema de la aceleración utilizamos la ecuación 25,

$$\Phi_q^c(q)\ddot{q}^c = -\dot{\Phi}_q^c\dot{q}^c$$

Se realiza la derivación de las velocidades de las coordenadas independientes para obtener las aceleraciones. Análogamente al proceso para hallar las velocidades de las coordenadas dependientes, también se realiza un arreglo de la ecuación anterior para calcular las aceleraciones de las coordenadas dependientes.

$$\ddot{\mathbf{q}}_0^c = \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \\ \vdots \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \vdots \end{bmatrix}$$

$$\Phi_q^c(\mathbf{q})\ddot{\mathbf{q}}^c = \begin{bmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) & -2(x_1 - x_2) & -2(y_1 - y_2) & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 2(x_1 - x_3) & 2(y_1 - y_3) & 0 & 0 & -2(x_1 - x_3) & -2(y_1 - y_3) & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots \\ 1 & 0 & -1 & 0 & 0 & 0 & \dots & -L_1 \sin(\theta_1) & \dots & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & \dots & \dots & -L_2 \sin(\theta_2) & \dots & 0 \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \\ \vdots \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \vdots \end{bmatrix}$$

$$-\Phi_q^c \dot{\mathbf{q}}^c = - \begin{bmatrix} 2(\dot{x}_1 - \dot{x}_2)^2 + 2(\dot{y}_1 - \dot{y}_2)^2 \\ 2(\dot{x}_1 - \dot{x}_3)^2 + 2(\dot{y}_1 - \dot{y}_3)^2 \\ \vdots \\ -L_1 \dot{\theta}_1 \cos(\theta_1) \\ -L_2 \dot{\theta}_2 \cos(\theta_2) \\ \vdots \end{bmatrix}$$

Como resultado se obtiene la siguiente ecuación:

$$\begin{bmatrix} -2(x_1 - x_2) & -2(y_1 - y_2) & 0 & 0 & \dots \\ 0 & 0 & -2(x_1 - x_3) & -2(y_1 - y_3) & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 \\ \vdots \end{bmatrix} = - \begin{bmatrix} 2(\dot{x}_1 - \dot{x}_2)^2 + 2(\dot{y}_1 - \dot{y}_2)^2 \\ 2(\dot{x}_1 - \dot{x}_3)^2 + 2(\dot{y}_1 - \dot{y}_3)^2 \\ \vdots \\ -L_1 \dot{\theta}_1 \cos(\theta_1) \\ -L_2 \dot{\theta}_2 \cos(\theta_2) \\ \vdots \end{bmatrix} - \begin{bmatrix} 2(x_1 - x_2) \\ 2(x_1 - x_3) \\ \vdots \\ 1 \\ 1 \\ \vdots \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \vdots \end{bmatrix} - \begin{bmatrix} 2(y_1 - y_2) \\ 2(y_1 - y_3) \\ \vdots \\ 0 \\ 0 \\ \vdots \end{bmatrix} \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \vdots \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ -L_1 \sin(\theta_1) \\ 0 \\ \vdots \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \vdots \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \dots \\ -L_2 \sin(\theta_2) \\ \vdots \end{bmatrix} \dots$$

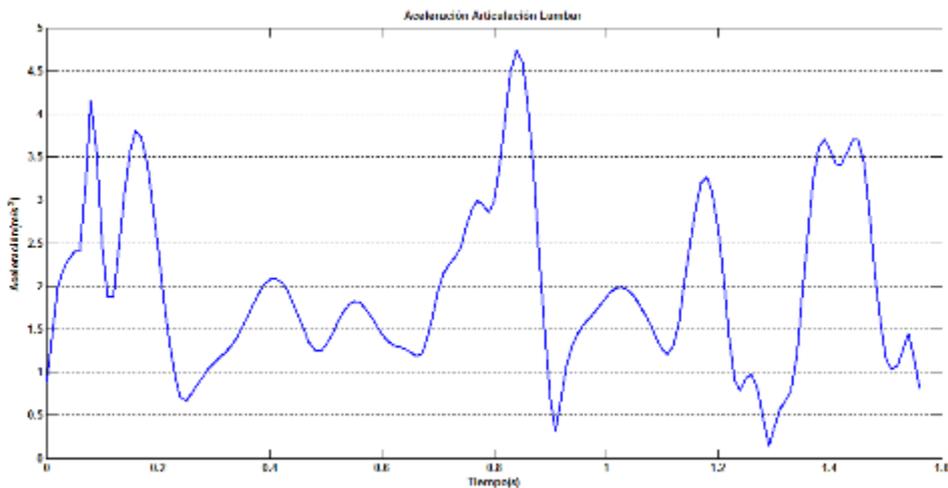


Figura 48 Aceleración lineal de la articulación lumbar en una persona sana.

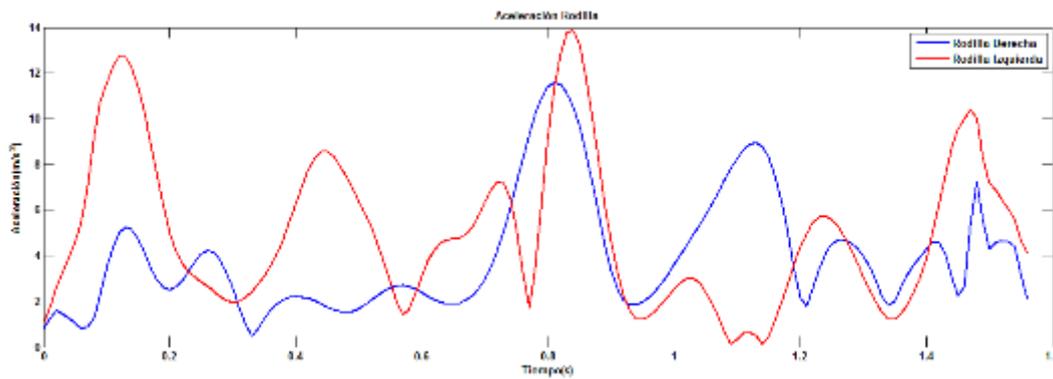


Figura 49 Aceleración lineal de las rodillas de una persona sana.

3.5 CÁLCULO DE REACCIONES Y ESFUERZOS MOTORES: DINÁMICA INVERSA

Cuando se tiene toda la información anterior: posición, velocidad y aceleración de las coordenadas independientes, se puede ahora calcular las fuerzas y momentos resultantes que actúan en las articulaciones.

La dinámica inversa (ID por sus siglas en inglés) implica resolver el problema de calcular los esfuerzos necesarios para conseguir que un segmento llegue a una posición con la velocidad y aceleración determinadas. El problema de ID es resuelto por medio de la formulación de la transformación de la velocidad conocida como Matriz \mathbf{R} , la cual provee de los esfuerzos motores requeridos para generar el movimiento, en la forma de fuerzas generalizadas aplicadas a las coordenadas independientes (Lugrís, Carlín, Luaces, & Cuadrado, 2013).

Determinamos \mathbf{z} como el vector que contiene a las coordenadas independientes, coordenadas de la articulación lumbar y ángulos. La formulación de la matriz \mathbf{R} está basada en la definición de la transformación de la velocidad de la matriz \mathbf{R} :

$$\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}} \dots (29)$$

Donde cada columna j de \mathbf{R} contiene las velocidades correspondientes a una unidad de velocidad de la j^{th} coordenada independiente de \mathbf{z}_j , mientras que las velocidades generalizadas restantes son ceros. El cálculo de la matriz \mathbf{R} involucra la solución del problema de la velocidad para cada grado de libertad (ángulo); esta ecuación está sujeto a:

$$\Phi_q \dot{\mathbf{q}} = \mathbf{0} \dots (30)$$

La velocidad de transformación puede ser aplicada a las ecuaciones de del movimiento de LaGrange:

$$\bar{\mathbf{M}}\ddot{\mathbf{q}} + \Phi_q^T \boldsymbol{\lambda} = \bar{\mathbf{Q}}_c + \bar{\mathbf{Q}}_m \dots (31)$$

Donde \bar{M} es la matriz de masas de los segmentos, \bar{Q}_c es el vector de las fuerzas generalizadas y \bar{Q}_m es el vector que contiene las reacciones del suelo y los momentos de las articulaciones, por último λ contiene la información de los multiplicadores de LaGrange.

Se deduce de la ecuación 29 y 30 que

$$\Phi_q R \dot{z} = \mathbf{0} \rightarrow \dot{z} \neq \mathbf{0} \rightarrow (\Phi_q R) = \mathbf{0}$$

Entonces se cumple:

$$(\Phi_q R)^T = R^T \Phi_q^T = \mathbf{0}$$

Si la ecuación de movimiento 31 son multiplicadas por R^T en ambos lados y arreglando los términos, se obtendría:

$$Q_m = R^T (\bar{M} \ddot{q} - \bar{Q}_c) \dots (32)$$

Donde:

$$M = \begin{bmatrix} m - 2m \frac{\bar{x}_G}{L} + \frac{I_1}{L^2} & 0 & m \frac{\bar{x}_G}{L} - \frac{I_1}{L^2} & -m \frac{\bar{y}_G}{L} \\ 0 & m - 2m \frac{\bar{x}_G}{L} + \frac{I_1}{L^2} & m \frac{\bar{y}_G}{L} & m \frac{\bar{x}_G}{L} - \frac{I_1}{L^2} \\ m \frac{\bar{x}_G}{L} - \frac{I_1}{L^2} & m \frac{\bar{y}_G}{L} & \frac{I_1}{L^2} & 0 \\ -m \frac{\bar{y}_G}{L} & m \frac{\bar{x}_G}{L} - \frac{I_1}{L^2} & 0 & \frac{I_1}{L^2} \end{bmatrix}$$

m : masa del elemento

\bar{x}_G y \bar{y}_G : coordenadas del centro de gravedad del elemento en el Sistema de referencia local.

I_1 : momento de inercia del elemento respecto al origen del sistema local.

L : longitud del elemento.

Por lo tanto, con la ecuación 32 se puede obtener directamente los momentos aplicados en las coordenadas independientes.

Con el planteamiento de nuestro modelo, se tiene 20 grados de libertad, definidos por 18 ángulos de orientación y 2 componentes de la posición de la articulación lumbar. Para el análisis de ID, es más práctico realizar la siguiente definición de los grados de libertad: 17 pares internos, 1 par externo y 2 fuerzas.

Cuando hay un solo pie de contacto con el suelo, esto pasa en la fase de balanceo, los momentos de las articulaciones pueden ser obtenidos sin los datos de la placa de fuerza, ya que no se tiene indeterminación para resolver el sistema, 20 grados de libertad y 20

magnitudes. Sin embargo, cuando se realiza la fase de doble apoyo, y más adelante el análisis que incluye dos muletas, se tendrá 3 datos redundantes debido a que tendremos 2 pares, más 4 fuerzas dando un total de 23 magnitudes y 20 grados de libertad.

Se establece el vector \mathbf{T} el cual contiene tres tipos de magnitudes: fuerzas externas, momentos externos y momentos internos. Y se puede vincular las fuerzas cartesianas \mathbf{T} con el vector de fuerzas generalizadas \mathbf{Q}_m (Lugrís, Carlín, Luaces, & Cuadrado, 2013)

$$\mathbf{B}^T \mathbf{T} = \mathbf{Q}_m \dots (33)$$

Donde \mathbf{B} es la matriz cuyas columnas contienen las velocidades $\dot{\mathbf{y}}$ obtenida cuando una la j^{th} coordenada generalizada es la unidad mientras que las demás son estáticas. Desde las velocidades de $\dot{\mathbf{y}}$ se derivan fácilmente de \mathbf{q} y $\dot{\mathbf{q}}$, y estos últimos están en la matriz \mathbf{R} .

$$\dot{\mathbf{y}} = \mathbf{B} \dot{\mathbf{z}}$$

Para que la ecuación 33 tenga una única solución, el número de variables en \mathbf{T} tiene que ser igual al número de grados de libertad. Sin embargo, \mathbf{T} contiene 3 variables de más con respecto a los grados de libertad. Esta solución única se obtiene con el problema de minimización:

$$\min_{\mathbf{T}} g(\mathbf{T}) = \frac{1}{2} \sum_{j=1}^6 w_j (\mathbf{T}_j - \mathbf{T}_j^*)^2$$

sujeto a $\mathbf{B}^T \mathbf{T} = \mathbf{Q}_m$

Donde w_j son los pesos asignados para cada desviación en la función $g(\mathbf{T})$ y \mathbf{T}_j^* son las medidas obtenidas por medio de placas de fuerza.

La resolución a este planteamiento del problema de minimización, se resuelve por multiplicadores de Lagrange:

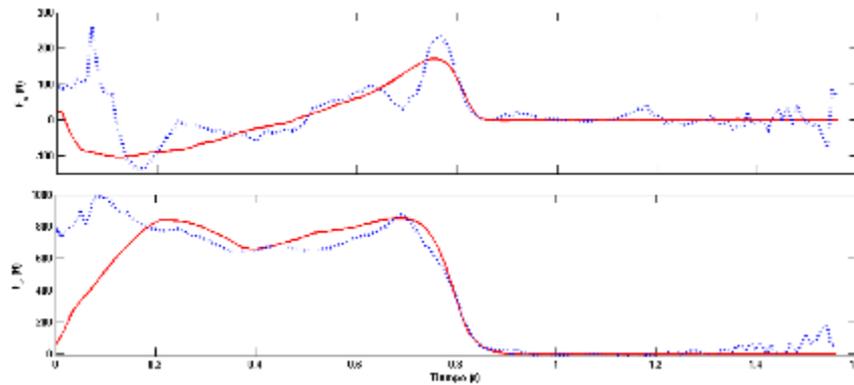
$$\begin{bmatrix} \mathbf{W} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{T} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{W} \mathbf{T}^* \\ \mathbf{Q}_m \end{bmatrix} \dots (34)$$

En este sistema lineal, \mathbf{W} es la matriz de pesos la cual solo tiene elementos distintos de cero, w_j , ubicados en la diagonal correspondientes a las posiciones de los 12 reacciones externas. \mathbf{T}^* es el vector que contiene las mismas posiciones de las 12 reacciones \mathbf{T}_j^* y $\boldsymbol{\lambda}$ son los multiplicadores de Lagrange.

La elección de los pesos w_j es importante para asegurar la consistencia de los resultados. Por ejemplo, si se determina el peso como la unidad, los resultados serán inconsistentes ya que podría resultar una reacción del suelo cuando en realidad no hay contacto con la superficie. Entonces se puede establecer los valores de los pesos como \mathbf{T}_j^{*-2} , para minimizar el error relativo a estos datos. Un buen uso del valor de los pesos para mantener los valores considerablemente cercanos a cero donde no hay contacto es $|\mathbf{T}_j^*|^{-0.5}$. Sin

embargo, por ser un modelo en 2D, las variaciones en los demás componentes podrías ser significativos.

Finalmente, el valor de $g(\mathbf{T})$ nos puede dar una medida de una buena calidad de captura de



movimientos.

Figura 50 Reacción del suelo (línea sólida: placas de fuerza, líneas punteadas: dinámica inversa)

3.0 PROTOTIPO ÓRTESIS ACTIVA

El prototipo de órtesis, el cual describiremos en este trabajo, es un prototipo desarrollado en el proyecto por Cuadrado, J et al. (2012) *Aplicación de técnicas de dinámica multicuerpo al diseño de órtesis activas para ayuda a la marcha*. XIX Congreso Nacional de Ingeniería Mecánica.

Esta órtesis es activa debido al motor rotativo ubicado en la articulación de la rodilla y por una serie de sensores que permiten el control de dicho prototipo.

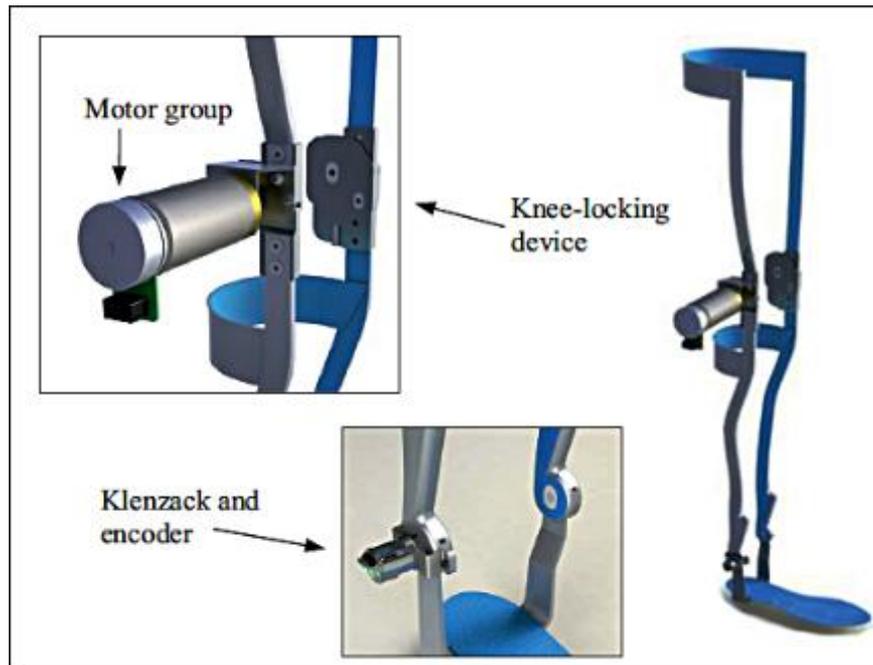


Figura 51 Prototipo de órtesis. (Fuente: Lugris, U. et al. (2013). Gait Analysis System for Spinal Cord Injured Subjects Assisted by Active Orthoses and Crutches)

3.6.1 Características

La prótesis consiste en un exoesqueleto de aluminio, el cual está compuesto por un motor eléctrico a la altura de las rodillas, junto con el motor eléctrico se instala un sistema de bloqueo de rodilla; a la altura del tobillo, un muelle Klenzack para evitar una marcha equina que de cierta manera evitará el tropiezo del paciente sobre su propio paso cuando el movimiento se encuentre en la fase de balanceo, y para medir el ángulo de la flexión, se adjunta un codificador para realizar dicha acción. También se configura un sensor plantar en la parte baja de la órtesis.

Un aspecto importante para la elaboración de esta órtesis, son las dimensiones del modelo. La órtesis es diseñada para un paciente determinado, quiere decir que depende de la longitud y grosos de las piernas del individuo; y que esto puede ir relacionado con el nivel de la lesión presentada en el paciente.



Figura 52 Prototipo de la órtesis usada en diferentes pacientes. (Fuente: Lugris, U. et al. (2013). Gait Analysis System for Spinal Cord Injured Subjects Assisted by Active Orthoses and Crutches)

La *Figura 52* muestra a dos pacientes con diferente nivel de lesión medular. El paciente de la izquierda tiene una lesión que deterioró el desarrollo muscular en la gran mayoría de sus extremidades inferiores, lo cual dificultó el diseño del prototipo de las órtesis a su medida. Dado que el exoesqueleto era parte de una donación de una empresa dedicada a la fabricación de este tipo de armazón, no se podía realizar las modificaciones para las medidas de dicho sujeto, pues perderían las propiedades mecánicas de este aparato. Por este motivo, se decidió cambiar de paciente, el cual cumpla las medidas aproximadas para el uso del aparato.

3.6.2 Funcionamiento

El prototipo de órtesis funciona de manera que pueda mover las piernas del paciente acercándose más a las características de una marcha normal. El sensor de contacto que va en la parte inferior de la órtesis manda una señal al motor y al mecanismo de bloqueo de la flexión de la rodilla, este sensor está configurado para trabajar con un cierto rango de presión para el sistema de bloqueo y desbloqueo de la órtesis. Una vez que no se detecta presión suficiente en dicho sensor, se desbloquea la flexión de la rodilla y el motor realiza el torque necesario para la fase de swing, en el tiempo necesario para el cual se configura dicho movimiento. El sistema del dispositivo de bloqueo evita el uso del motor para bloquear la flexión de la rodilla en la fase de apoyo, y realiza un bloqueo mecánico.

El ángulo de la rodilla sigue a un historial de variación introducido en el control de la órtesis, la variación del ángulo puede ser adaptada a diferentes velocidades de la marcha. Una aproximación más sofisticada usa la relación entre el ángulo de la rodilla en la fase de balanceo y el ángulo del tobillo opuesto, esta metodología permite al usuario cambiar dinámicamente la velocidad de la marcha.

3.6.3 Introducción de las órtesis y muletas en el modelo

Las órtesis formarán parte de las extremidades inferiores, las propiedades inerciales de los muslos, pantorrillas y pies serán modificados. Como se explicó en un inicio, el movimiento de traslación de la persona que utiliza las órtesis, será asistido por un par de muletas las cuales también son introducidas en el modelo como una extensión de las muñecas.

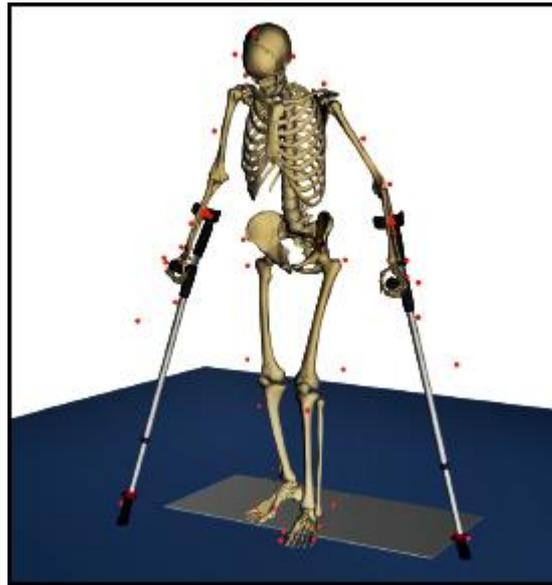


Figura 53 Posición de los markers. Persona con muletas y órtesis. (Fuente: Lugris, U. et al. (2013). Gait Analysis System for Spinal Cord Injured Subjects Assisted by Active Orthoses and Crutches)

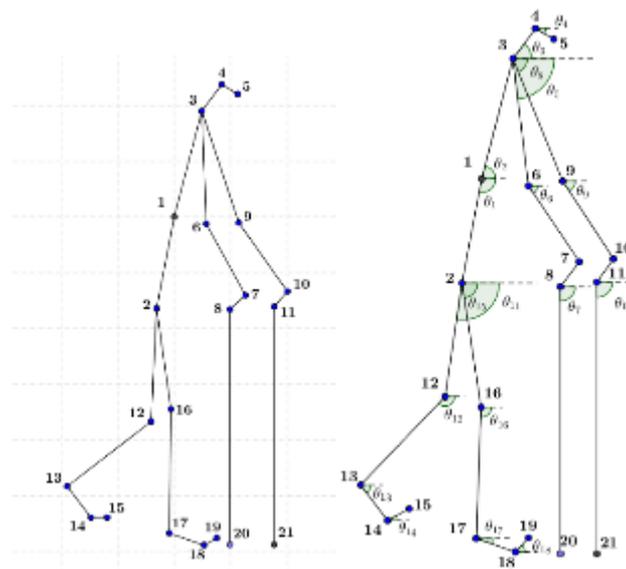


Figura 54 Modelamiento de puntos y segmentos de una persona que utiliza órtesis y muletas.

El nuevo modelamiento del sistema de sólidos rígidos es simplemente una extensión del modelo cuerpo completo que se estudió anteriormente. Se elimina el centro de gravedad de las manos y se representa como siguiente punto, al de la muñeca, la base de la empuñadura de la muleta. La nueva representación de puntos de las articulaciones se visualiza en la *Figura 55*.

Los puntos 7 y 10 son las articulaciones de las muñecas, y los puntos 8 y 11 son los puntos ubicados en la base de la empuñadura.



Figura 55 Ejemplo del modelamiento muleta-mano derecha.

Con las especificaciones anteriores, se puede resolver el problema la consistencia cinemática. Se crea un sólido rígido ficticio entre la muñeca y la base de la empuñadura ya que la distancia entre dos puntos es casi constante, y el siguiente sólido sería el correspondiente a este último punto con el final de la muleta.

Los cálculos para obtener las velocidades y aceleraciones de los puntos y centros de gravedad, también se obtienen de manera similar al modelo de cuerpo completo.

Para realizar la dinámica inversa, a parte de las fuerzas conocidas que son el peso de cada sólido rígido, también se introduce una función en la matriz de fuerzas conocidas Q_c , esta función determina los momentos que realizan los motores de las órtesis. El control de la órtesis se encarga de ejecutar el algoritmo adecuado para poder determinar la diferencia, en cada instante de tiempo, entre el ángulo que forma la rodilla y el ángulo establecido como óptimo. Finalmente, se calcula el torque necesario para realizar la aproximación al ángulo de referencia de una marcha normal. En este proceso de realizar el torque necesario para llegar al ángulo de referencia, también se controla el bloqueo de la rodilla, una vez alcanzado dicho ángulo, el bloqueo de la rodilla se activa bloqueando la flexión de esta articulación.

3.7 EFICIENCIA ENERGÉTICA EN LA MARCHA

Nuestro aparato locomotor no es tan eficiente, aunque si más versátil y seguro. Y nuestro centro de gravedad describe una trayectoria ondulada; en un movimiento eficiente, el centro de gravedad describiría una trayectoria lineal. Los mecanismos de optimización mejoran el rendimiento en la marcha, ya sea por deficiencia o exigencia física, a través de transferencias de energía y minimización del desplazamiento del centro de gravedad (Inman et al., 1981; Gage, 1990; Winter, 1992).

Existen diferentes metodologías en el análisis de la variación de energía en la marcha humana:

- Análisis de segmentos
- Dinámica inversa
- Energía metabólica por el ritmo cardíaco y velocidad
- Consumo de oxígeno y emisión de dióxido de carbono

Cuando una persona camina, existen dos formas de intercambio de energía: transformación de energía cinética y potencial gravitatoria, y transferencia de energía entre segmentos.

Una órtesis pasiva bloquea el movimiento de flexión de la rodilla; es posible que el paciente consiga una estabilidad junto con el uso de muletas. Sin embargo, al realizar un avance, con este tipo de órtesis, el consumo de energía será excesivo, puesto que tenderá a alzar más las caderas para ejecutar el balance de las piernas. Por otro lado, una órtesis activa realiza un torque necesario para que la rodilla se flexione al momento que el pie no tenga contacto con el suelo, y realice el movimiento de balance, un movimiento más natural. Además de bloquear la flexión de la rodilla cuando se detecte que el pie está en contacto con el suelo.

Si analizamos el centro de gravedad del cuerpo, se evidencia el intercambio de energías cinética y potencial gravitatoria. En la fase de doble apoyo, el centro de gravedad del cuerpo se encuentra su posición más baja y presenta su máxima velocidad hacia adelante, esto se traduce en una menor energía potencial gravitatoria y mayor energía cinética. En contraste, durante la mita de fase del apoyo monopodal, el centro de gravedad alcanza su máxima posición de su recorrido y la velocidad es mínima, es decir, máxima energía potencial gravitatoria y mínima energía cinética.

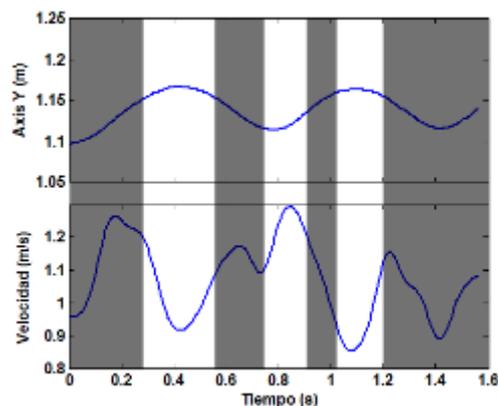


Figura 56 Altura y velocidad del punto de la articulación lumbar, cercana al CG.

blandas, la cual se transforma en energía cinética. Los músculos son como transductores que convierten la energía química en energía eléctrica, térmica y/o mecánica, a su vez los músculos contribuyen a suavizar la trayectoria del centro de masa.

Los factores biomecánicos que reducen y suavizan los desplazamientos verticales del centro de gravedad son:

- Rotación de la pelvis.
- El descenso de la pelvis hacia el lado oscilante.
- La flexión de la rodilla en el lado del apoyo.
- Los movimientos coordinados de rodilla, tobillo y pie.

Dentro de las características de la marcha del paciente, cuando utiliza una órtesis pasiva, se ve un movimiento excesivo de pelvis para poder realizar la fase de balanceo e implica un mayor esfuerzo para llevar el cuerpo hacia adelante, esfuerzo en abdomen y brazos para mantener el equilibrio del cuerpo soportado por las muletas.

Cuando se modela el cuerpo de una persona sana y paciente como un sistema de sólidos rígidos, es posible realizar el cálculo de Energía Mecánica en la marcha de ambos.

$$E_K = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \dots (35)$$

De esta manera se determina la energía cinética total (E_K) en cada instante de tiempo; la ecuación 35 permite el cálculo de la energía cinética de traslación y rotación, donde $\dot{\mathbf{q}}$ tiene la información de las velocidades de las coordenadas de los puntos que sirven para modelar todos los sólidos del sistema. En la matriz de masa \mathbf{M} aparecen los términos inerciales clásicos: masas de los elementos, posición de los centros de gravedad, momento de inercia y longitudes.

$$U_g = m \cdot g \cdot r_Y \dots (36)$$

También se realiza el cálculo de la Energía Potencial Gravitatoria (U_g), donde m es la masa del sólido, g la aceleración de la gravedad y r_Y coordenada vertical de la posición del centro de gravedad del sólido con respecto al eje global.

Capítulo 4

EXPERIMENTACIÓN Y RESULTADOS

4.1 CAPTURA DE DATOS

4.1.1 Pre proceso de datos

Se realiza una triangulación de los puntos que detecta el sistema de cámaras utilizadas dentro de la sala de captura, esta triangulación permitirá que se pueda reconstruir las trayectorias de los marcadores (*markers*), ya que conociendo la orientación y posición de cada uno, se puede identificar las trayectorias en 3D.

Es posible que durante la captura de movimiento, las trayectorias estén registradas por segmentos, esto se debe a intervalos de tiempo en el que los datos obtenidos por las cámaras son insuficientes para realizar la triangulación. Esto se produce porque la posición del marcador puede estar oculta a la mayoría de las cámaras, al estar tapado por el cuerpo.

Para corregir este tipo de falla, se realiza un análisis de las trayectorias de los marcadores. Pero dentro del sistema de captura, se tiene solucionado este problema; se ha introducido un código para determinar la trayectoria de un marcador en caso la trayectoria presente un hueco o salto en el proceso de captura.

El programa de captura no siempre ordena las trayectorias obtenidas de la misma manera, por lo que hay que identificar los marcadores. Si no se identifica los marcadores, el programa no identificará qué marcador pertenece a cada parte del cuerpo. Se utiliza dos técnicas para las identificaciones de los marcadores. La primera, consiste en el historial de la primera configuración de las identificaciones de los marcadores, estas ubicaciones se comparan con las nuevas ubicaciones de los marcadores del sujeto de ensayo. Los marcadores se ordenarán de forma que la distancia entre el marcador original y el de la captura sea mínima, teniendo en cuenta una corrección de posición en función del centro de gravedad de los marcadores. Cuando se tenga que realizar otra configuración de los marcadores, en el caso de una configuración que intervenga a un sujeto con órtesis y muletas, la configuración tiene que ser manual e ir separando cada marcador en función a sus coordenadas de la posición inicial.

Es importante la sincronización de entre las placas de fuerza y las cámaras. Esta sincronización consiste en golpear una de las placas con un mazo de goma. La peculiaridad del mazo utilizado para este procedimiento, es que tiene un marcador en la parte superior, entonces se hace coincidir el instante en el que se detecta el golpe en la placa con el instante en el que la coordenada z del marcador alcanza un mínimo.



Figura 57 Mazo utilizado para la sincronización de cámaras y placas de fuerzas.

Para calibrar las cámaras, es necesario realizar un barrido de todo el volumen de captura con la herramienta de la *Figura 57*.



Figura 58 Large Volume OptiWand Kit

El programa utilizado para la captura, analiza las trayectorias de los tres marcadores obtenidas por cada cámara. Además, se utiliza otra herramienta tipo escuadra, la cual determina el centro de referencia global y la orientación de sus ejes. Para este estudio, el origen de coordenadas se ubica en el medio de las posiciones de las placas de fuerzas.



Figura 59 Calibration Square

Finalmente, se procede a configurar el programa que controla las placas para que la orientación de sus ejes coincida con los ejes globales.

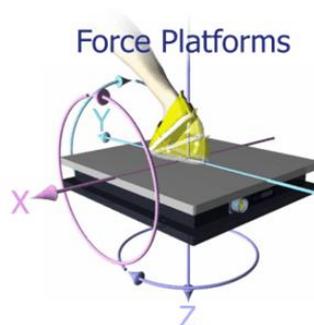


Figura 60 Configuración por defecto de los ejes de las placas.

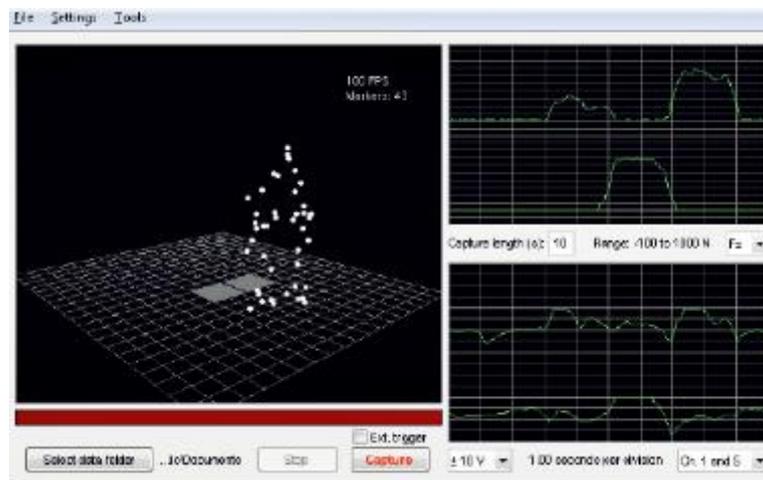


Figura 61 Software de captura

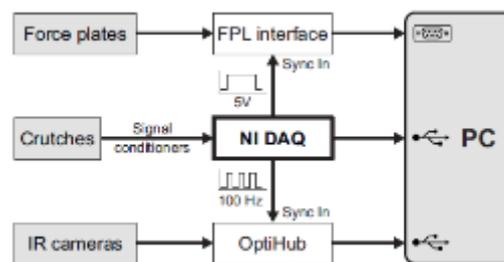


Figura 62 Diagrama de conexión de hardware

De acuerdo con la configuración de los hardwares para la captura de movimiento, se muestra el diagrama de la *Figura 62*. Las placas de fuerzas cuentan con una interface propia, así como también las cámaras de captura cuentan con una interface específica OptiHub. Estas dos interfaces están conectadas con un National Instrument USB DAQ. Para la investigación del proyecto general que desarrolla el grupo de investigación de las UDC y UPC, se implementó sensores y las muletas; y para efectos de esta tesis de pregrado, no se necesita expandir en el estudio de los sensores y la data que se pueda obtener de éstos.

4.1.2 Proceso de capturas

Una vez calibrado todo lo anterior, se procede a tomar las medidas antropométricas del individuo, explicado en el Capítulo 3, y el peso total del sujeto de estudio. Estos datos se introducen dentro de un archivo estándar, el cual es leído por el programa para estimar los parámetros biométricos. El sujeto se coloca el traje de captura y se ubican los marcadores en los puntos establecidos. En el caso del paciente se utilizó ropa adecuada para que no hubiese ruido en la captura de movimiento, ya que no se contaba con un traje especial

para una persona con las características físicas de padecer una lesión medular. Se sitúa al sujeto en el espacio de captura en la posición inicial, colocándose de tal forma que el primer paso coincida con la primera placa. El operador ejecuta el programa Even Gosht que iniciará el programa de capturas de las cámaras y de las placas, y se procederá a la sincronización de placas y cámaras. Por último, se da una señal para que el sujeto camine sobre las placas.

El proceso que sigue los datos hasta obtener los resultados de simulación se explica en el *Esquema 1*.

Denominaremos Programa de Captura al algoritmo de Matlab que es utilizado para obtener los ficheros necesarios que serán los inputs para el Programa de Simulación, también desarrollado en Matlab. Uno de los productos de esta investigación es el Programa de Simulación, por otro lado el Programa de Captura ya es un código existente propiedad del Laboratorio de Ingeniería Mecánica de la Universidade A Coruña.

Programa de Captura

Los inputs del programa de captura son los datos obtenidos de las capturas de los marcadores, que corresponden a las trayectorias de éstos (C3D), los valores de fuerzas y momentos en las placas (Placas), y la tabla de medidas antropométricas del sujeto de estudio (APM).

La función **Reconstrucción** utiliza los datos de trayectorias C3D, se analiza y se comprueba que no tengan variaciones de posición (saltos) entre dos instantes de tiempo consecutivos, dentro de un límite razonable. Si se determina que hay un salto, la función busca cual es el marcador con el que se ha podido intercambiar la información de la trayectoria; y si no encuentra concordancia con alguna información, se determina por aproximación a una función polinómica la trayectoria perdida. Luego estos datos reconstruidos son filtrados con el algoritmo de filtrado *ssa*. El algoritmo de **ordenar_datos** ordena los datos de acuerdo a lo modelado para identificar los marcadores, con respecto al eje z, los de menor valor corresponderán con los pies, posteriormente pantorrillas, muslos, cadera, torso, brazos, cuello y cabeza. Luego por el valor del eje y, separando el lado derecho e izquierdo, así como identificando los marcadores de cada brazo y pierna. Finalmente se ordena con respecto al eje x para aquellos marcadores que pueda quedar en alguna posición de duda. La función **Seleccionar_datos** realiza cuatro acciones: sincronización de los datos de las cámaras de captura y las placas de fuerzas, recortar los datos de las trayectorias y placas durante el intervalo de tiempo de la marcha, filtro de las señales obtenidas por las placas y calcular el centro de presión en estas (FPL). En una última estancia, la función **Trayectorias** se encarga de determinar la posición de los extremos de los segmentos que representan el cuerpo (r_0) y los ejes de referencia locales (r_l), así como la información para formar las matrices de rotación de los ejes (AM) y las posiciones de los centros de gravedad locales (r_g).

La función **BSP** es la que realiza los cálculos para estimar los parámetros biométricos (MI), como las masas y momentos de inercia de los segmentos.

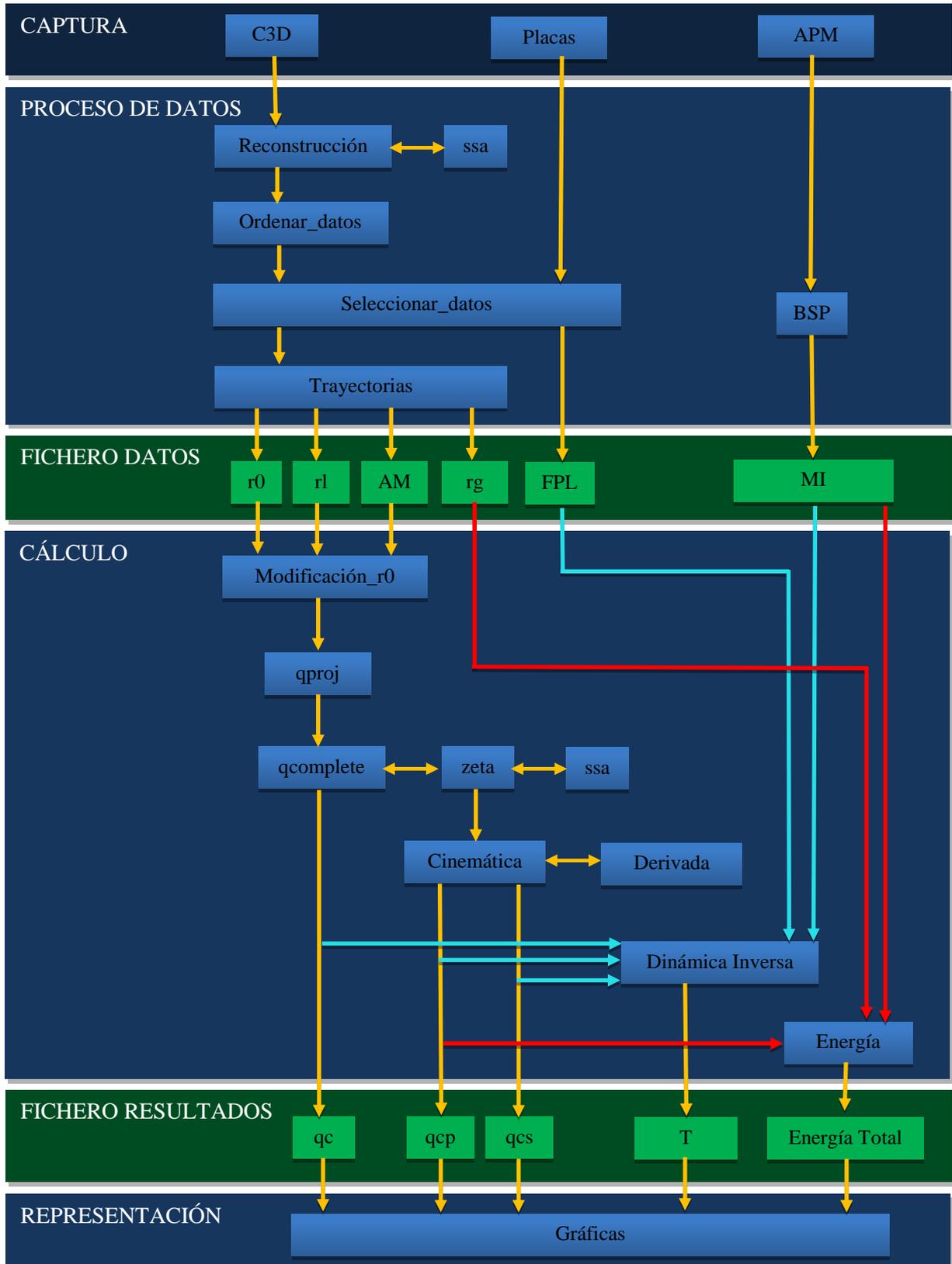


Figura 63 Programa cálculo de esfuerzos articulares y cálculo de energía en Matlab.

Programa de Cálculo

El inicio de los cálculos para poder alcanzar el objetivo de este proyecto, comienza cuando se establecen las posiciones de los puntos de los segmentos que se modelan. La función **modificación_r0** realiza los cálculos para definir las posiciones de todas las articulaciones de acuerdo a la modelización establecida, así como las longitudes de los segmentos y el paso de 3D a 2D en la modelización e identificación de las articulaciones en esta última dimensión. Posteriormente, se realiza la consistencia cinemática, con la función **qproj**, se establece las nuevas posiciones de las articulaciones en 2D y las longitudes de los segmentos de acuerdo a las ecuaciones de restricción, asegurando la consistencia cinemática

La función **qcomplete** permite integrar en un solo vector las coordenadas y ángulos que modelan todo el sistema. Y se separa las variables determinadas como los grados de libertad (**zeta**), que son los que determinan el sistema de sólidos rígidos en cualquier instante de tiempo. Se aplica el filtro sobre los datos de los grados de libertad, con la función **ssa**; una vez filtrado los grados de libertad, se introducen en el vector de coordenadas y ángulos, y de nuevo se realiza la consistencia cinemática de acuerdo a los nuevos valores de los grados de libertad (**qc**). Se realiza el cálculo cinemático usando la función **derivada** para conocer las velocidades (**qcp**) y aceleraciones (**qcs**) de las uniones de los sólidos del sistema.

Para realizar la dinámica inversa, es necesario tener como inputs **qc**, **qcp**, **qcs**, datos de las placas de fuerzas FPL y parámetros inerciales del fichero MI. El resultado de realizar la Dinámica inversa, se obtiene el fichero T, el cual contiene la información de las fuerzas de reacción del suelo y los momentos de las articulaciones del sistema multicuerpo.

Por último, se realiza los cálculos de **Energía** con la información de los ficheros que contienen los datos de los centros de gravedad, velocidades de los puntos que modelan los segmentos corporales, la masa y momentos de inercia de los segmentos que se obtiene del fichero MI.

Con todos los ficheros que se obtienen luego de los cálculos, se puede generar una simulación de la marcha de la persona o paciente, según sea el estudio, así como las velocidades y aceleraciones de las articulaciones, así como las variaciones de los ángulos de dichas articulaciones y los momentos que se generan. También se puede generar la variación de la energía del mecanismo a través de gráficas que separen la energía potencial y cinética.

4.2 ANÁLISIS DE RESULTADOS

En la simulación de la marcha de una persona sana y el paciente, se puede describir las fases del movimiento, realizando una comparación cualitativa, señalando el trabajo que realiza las órtesis para que el paciente realice una marcha semejante a la normal. Se diferencia la parte derecha del cuerpo como líneas sólidas y la parte izquierda, como líneas discontinuas.

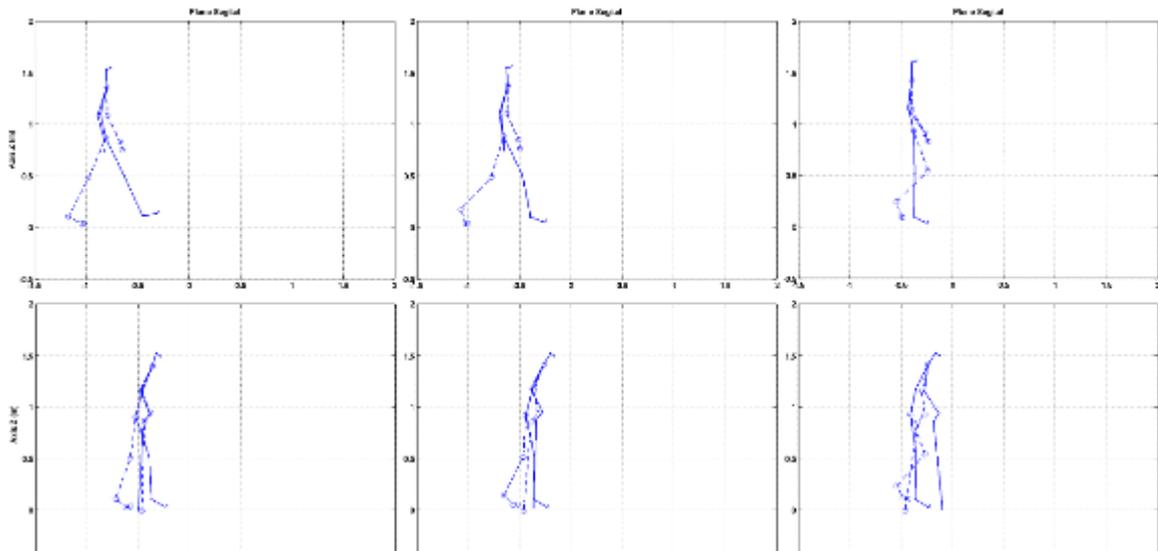
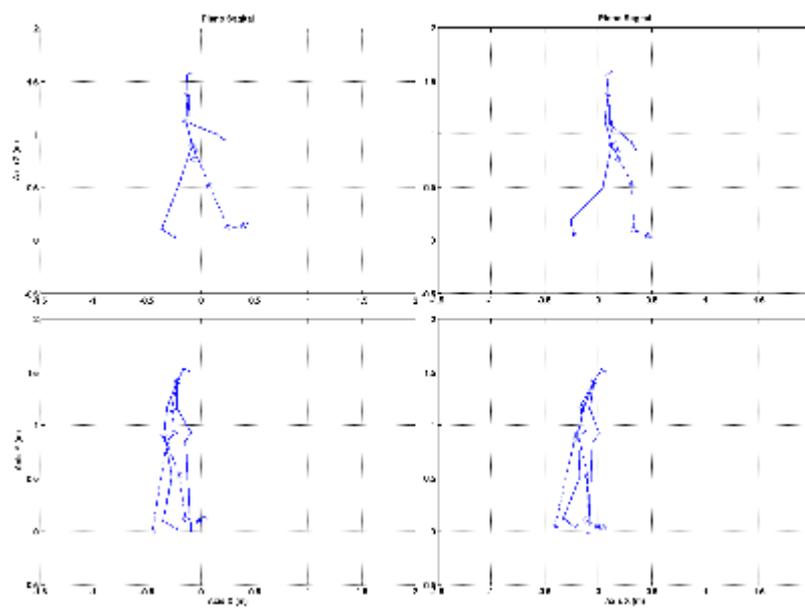


Figura 64 Simulación de la marcha de una persona sana (parte superior) y de un paciente con órtesis (parte inferior). Parte I

En la parte inicial del movimiento, se presenta el momento del contacto inicial. Para el paciente es necesario que las muletas estén casi al mismo nivel para poder conseguir el



equilibrio necesario en esta parte del movimiento.

El siguiente movimiento se caracteriza cuando el talón izquierdo comienza a perder contacto con el suelo. Una vez que el pie izquierdo pierde el contacto con el suelo, la órtesis realiza el balanceo de la pierna izquierda. La posición de las muletas cambia, ya que el paciente busca un mejor apoyo para realizar el paso con ayuda de la nueva posición de su cadera. Una vez que el control haya determinado que se recorrió el ángulo de referencia, comienza el apoyo del pie izquierdo y el talón del pie derecho se prepara para comenzar a perder contacto.

El paciente se inclina hacia adelante para seguir la marcha, llevando la muleta izquierda hacia adelante, preparando el equilibrio para cuando la órtesis de la pierna derecha realice la fase del swing.

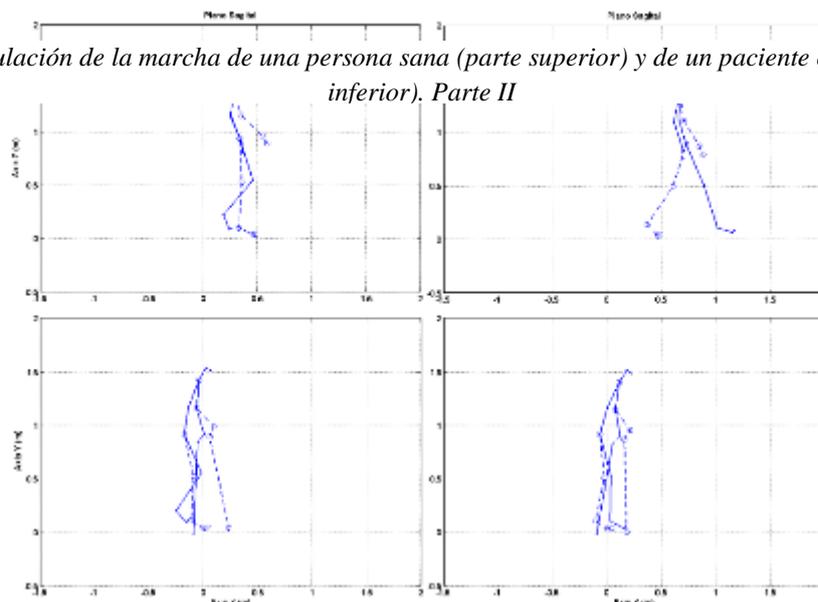
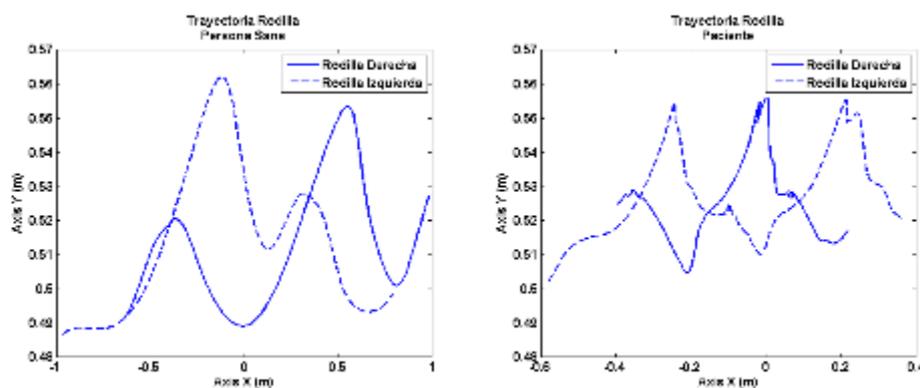


Figura 65 Simulación de la marcha de una persona sana (parte superior) y de un paciente con órtesis (parte inferior). Parte II

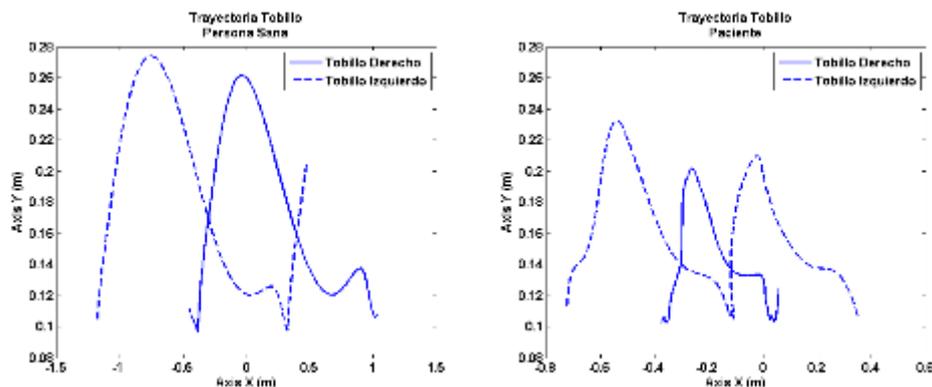
Figura 66 Simulación de la marcha de una persona sana (parte superior) y de un paciente con órtesis (parte inferior). Parte III

Finalmente, el pie derecho se apoya sobre el suelo y el control de la órtesis detecta esta reacción y permite el bloqueo de la rodilla derecha para seguir con el movimiento de la



pierna izquierda, completando el ciclo de la marcha.

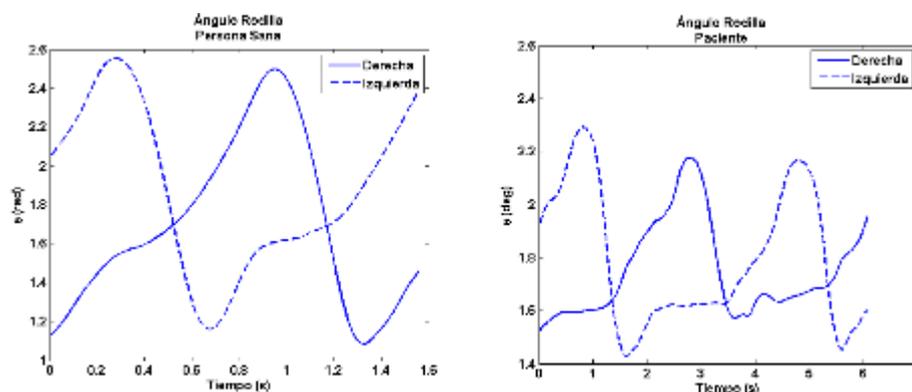
Se observa que en la *Figura 67* de la trayectoria de la rodilla del paciente que usa la órtesis, presenta picos debido a que en esa posición el motor de la órtesis realiza el torque necesario para el giro de la rodilla en la fase del swing. El uso de la órtesis contempla el ritmo de la forma en que una rodilla se desplaza con respecto de la otra, con un timing



necesario para asemejar la marcha normal.

Los tobillos del paciente realizan un desplazamiento limitado por el muelle Klenzack; sin embargo, la forma del movimiento se asemeja al desplazamiento de una persona sana.

El desplazamiento del ángulo de la rodilla, cuando se utiliza una órtesis, viene determinado por una gráfica de un ángulo de rodilla de una persona sana. Como se muestra en la *Figura 69* las gráficas tienen igual forma, la diferencia entre los máximos y mínimos, entre ambas,



radica en la necesidad de la flexión y extensión de la rodilla para realizar el movimiento.

También se obtiene la gráfica con la que el ángulo cambia con respecto al tiempo, es decir la velocidad de cambio de este ángulo. La gráfica de la velocidad del ángulo de la rodilla

en el paciente que tiene la órtesis tiene un comportamiento no tan parecido a la gráfica de la velocidad del ángulo de la rodilla de una persona sana, debido a que el control realiza la corrección de la variación del ángulo. Como se espera, la velocidad cambia repentinamente una vez alcanzado el ángulo objetivo. Los mínimos relativos de la gráfica, representan la fase del balanceo en cada rodilla, derecha o izquierda ya sea el caso.

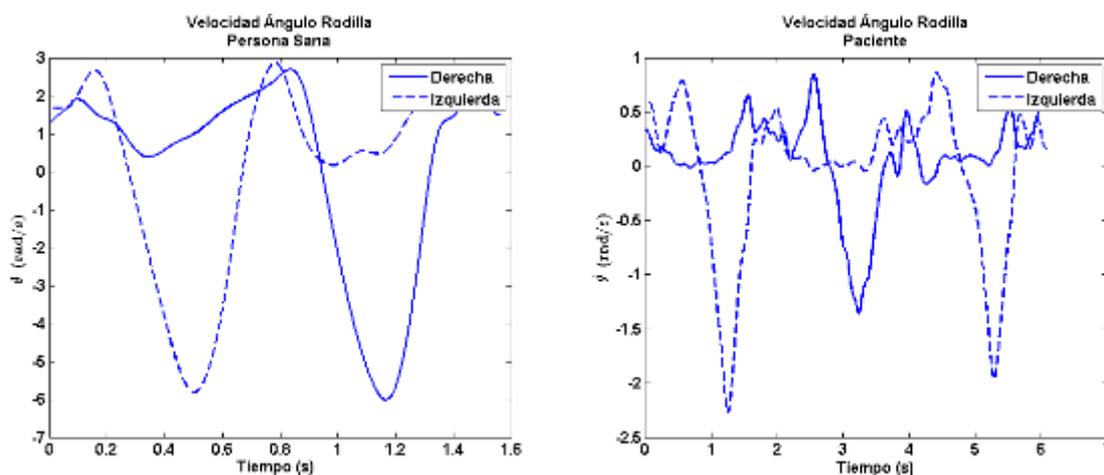


Figura 70 Velocidad del ángulo de la rodilla

En las gráficas de la aceleración, se puede observar los cambios constantes de aceleración del ángulo de la rodilla, en el caso del paciente, no se puede observar un comportamiento similar a la aceleración del ángulo de la rodilla de una persona normal. Se deduce que las causas de esta irregular forma de la gráfica es la disminución de variables en la consistencia cinemática, es decir, a consecuencia del paso de 3D a 2D. Además del ajuste del equilibrio cuando el paciente utiliza la órtesis junto con las muletas.

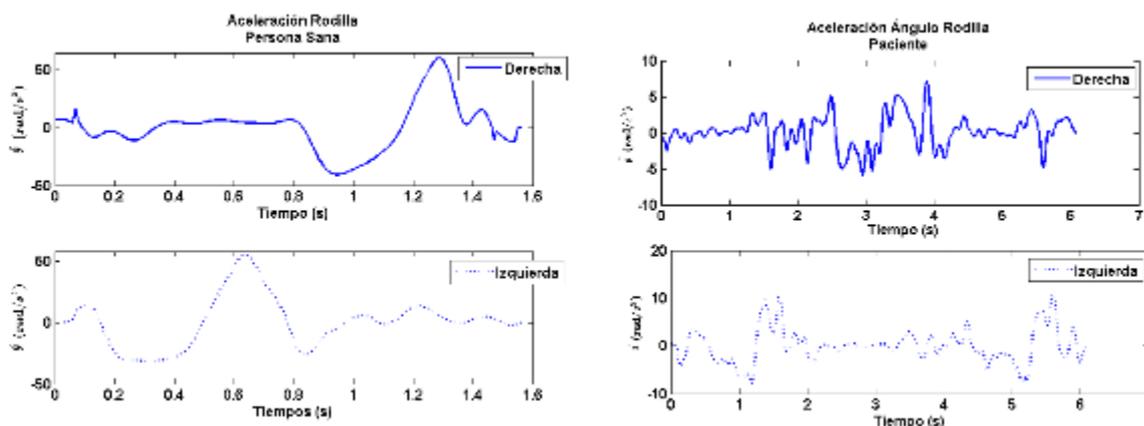


Figura 71 Aceleración del ángulo de la rodilla

Del análisis cinemático anterior, se constata que el comportamiento del sistema multicuerpo de una persona que usa una órtesis es parecido al movimiento de traslación de una persona sana, considerando las limitaciones naturales del uso de la órtesis en el movimiento.

A continuación, se muestra las gráficas del resultado de los cálculos de la dinámica inversa, comparando los sistemas multicuerpo de una persona sana y del paciente.

REACCIÓN EN EL EJE HORIZONTAL

PERSONA SANA

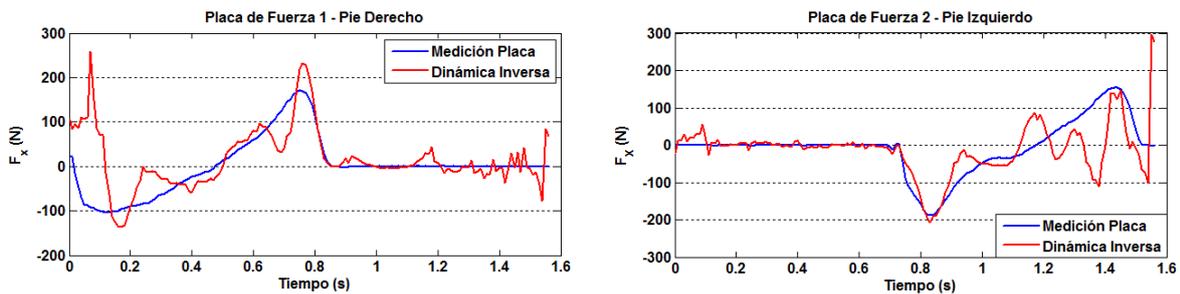


Figura 72 Medición de Fuerza en placa y Dinámica Inversa en el eje X, en una persona sana

PACIENTE CON ÓRTESIS

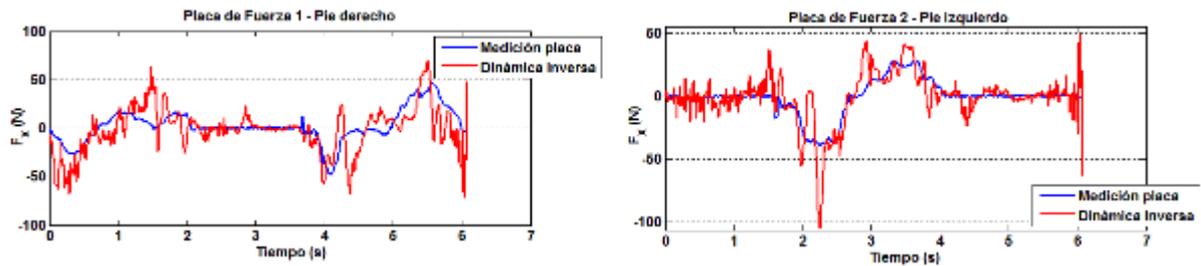


Figura 73 Medición de Fuerza en placa y Dinámica Inversa en el eje X, en un paciente con órtesis

REACCIÓN EN EL EJE VERTICAL

PERSONA SANA

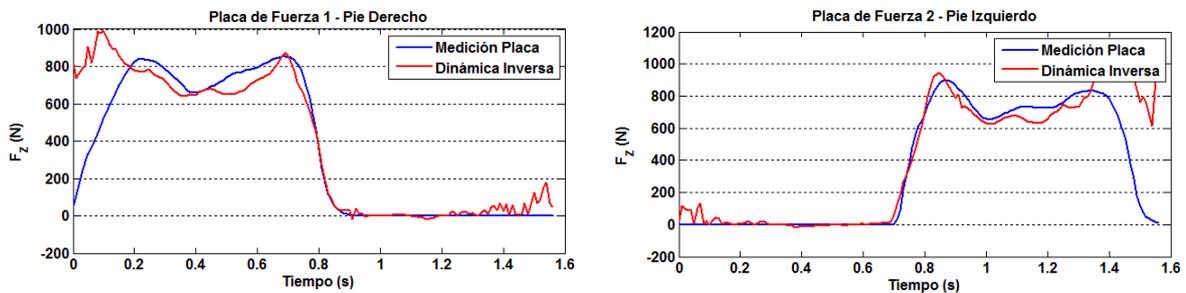


Figura 74 Medición de Fuerza en placa y Dinámica Inversa en el eje Z, en una persona sana

PACIENTE CON ÓRTESIS

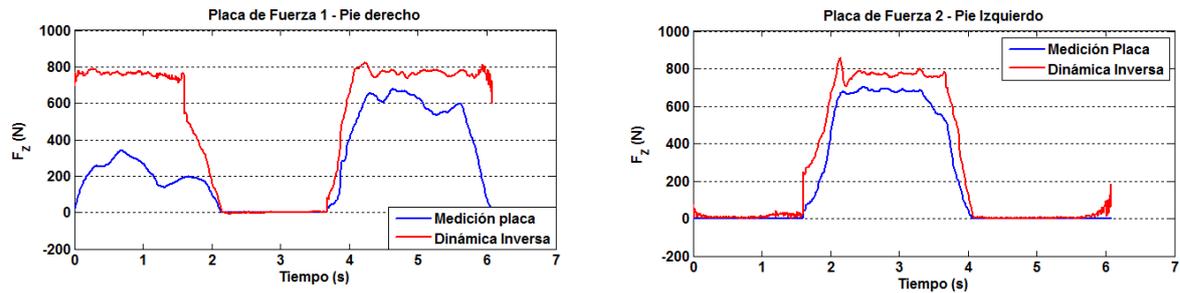


Figura 75 Medición de Fuerza en placa y Dinámica Inversa en el eje Z, en un paciente con órtesis

MOMENTOS CON RESPECTO AL TERCER EJE

PERSONA SANA

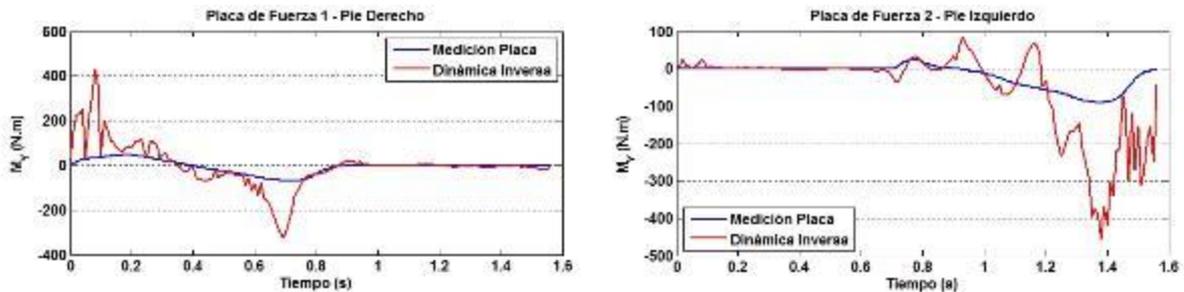


Figura 76 Medición de Momento en placa y Dinámica Inversa en el eje Y, en una persona sana.

PACIENTE CON ÓRTESIS

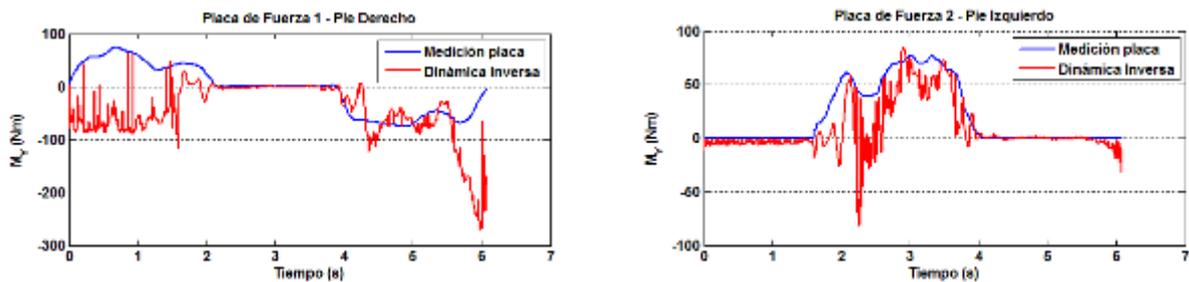


Figura 77 Medición de Momento en placa y Dinámica Inversa en el eje Y, en un paciente con órtesis.

Al momento de realizar la dinámica inversa, se toma como prioridad la aproximación de los datos de dicho análisis dinámico inverso cercano a cero cuando no se detecta contacto de los pies en las placas de fuerza, esto se logra por medio de los penalizadores en el problema de minimización por multiplicadores de Lagrange, ecuación (34). De cierta manera, la dinámica inversa no resultará en la fase de balanceo, ya que trata de resolver un sistema de cadena abierta, por la pérdida de reacciones en una pierna. Por otro lado, cuando se realiza la fase de doble apoyo, la dinámica inversa funciona aceptablemente.

PLACAS DE FUERZA-PERSONA SANA

Se tiene la información de las placas de fuerza, las cuales nos da la información de la fuerza ejercida por la persona al caminar. Usando Matlab, se reproduce una animación la cual muestra el vector de fuerza, con origen el centro de presión (COP, por sus siglas en inglés) registrada por la placa. El sentido de esta fuerza está orientado al CG del cuerpo y es independiente para cada miembro inferior. La fuerza, en la pierna de apoyo, va aumentando y llega en su punto máximo cuando la otra pierna comienza a perder contacto con el suelo. Cuando se produce la fase del balanceo, esta fuerza disminuye pero vuelve a aumentar cuando la pierna deja la fase del balance y comienza la fase de doble apoyo. Mientras transcurre la fase de doble apoyo, la fuerza comienza a disminuir rápidamente.

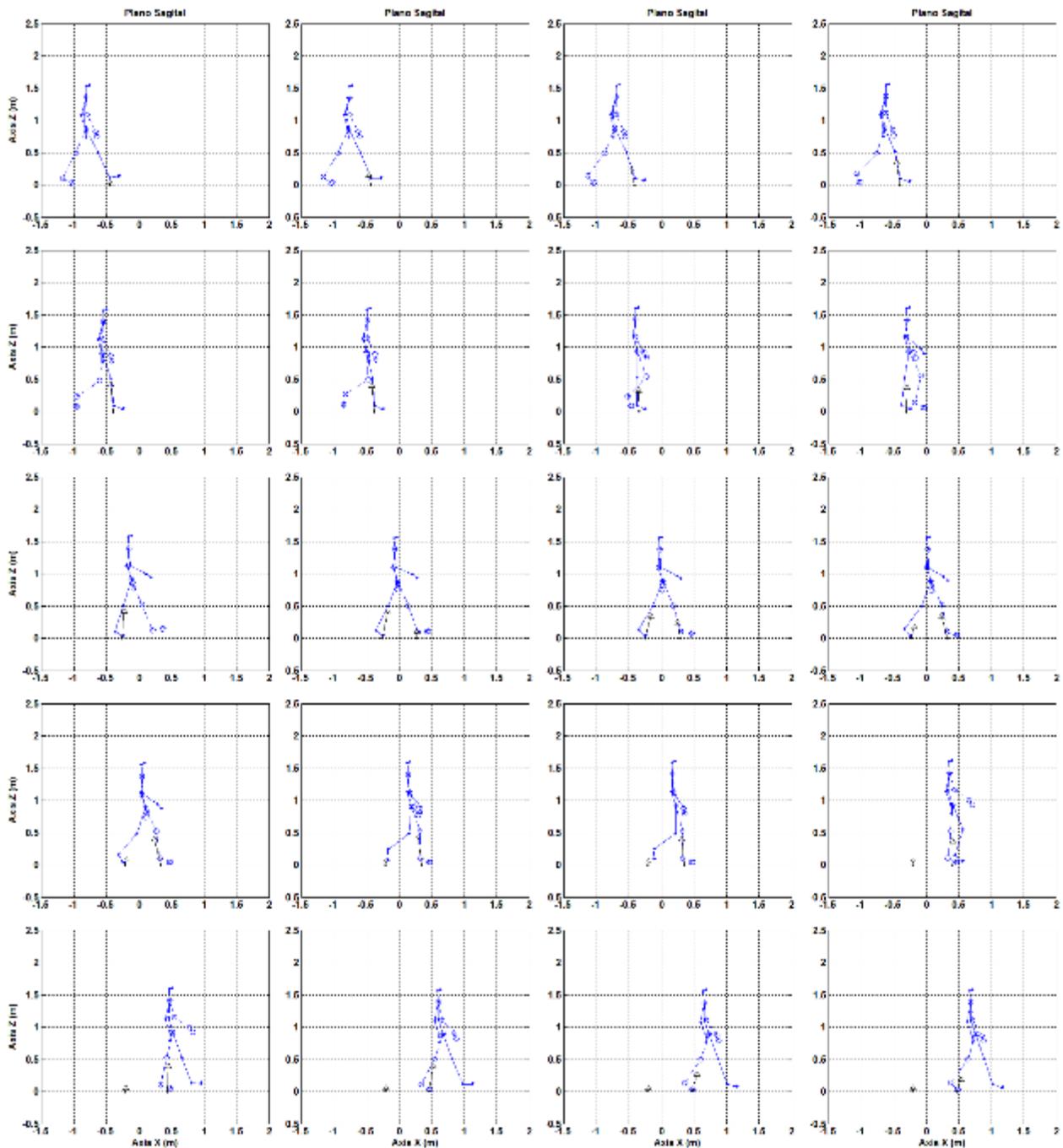


Figura 78 Simulación Placas de Fuerza, en Persona Sana.

Análogamente, se vuelve a repetir este proceso en la pierna contraria.

PLACAS DE FUERZA-PACIENTE

Para el caso del paciente, las placas detectan un comportamiento muy similar en la marcha asistida con muletas y órtesis activa. Una de las características principales de estas reacciones es que la carga está distribuida, no solo en las extremidades inferiores, sino también en las muletas, es por eso se registra menor magnitud de las fuerzas en las extremidades inferiores. Se puede verificar que al inicio del movimiento, se registra una fuerza pequeña, ya que el paciente muestra un poco de duda al comenzar a caminar, y mayor fuerza en las muletas (no registrado en este estudio). Luego de la parte inicial del movimiento, las fuerzas registradas tienen un comportamiento de una marcha normal. Sin embargo, el COP desde el cual se modela el origen de la fuerza de la pierna de apoyo varía de posición, casi oscilante, y esto es a consecuencia de la capacidad del paciente para

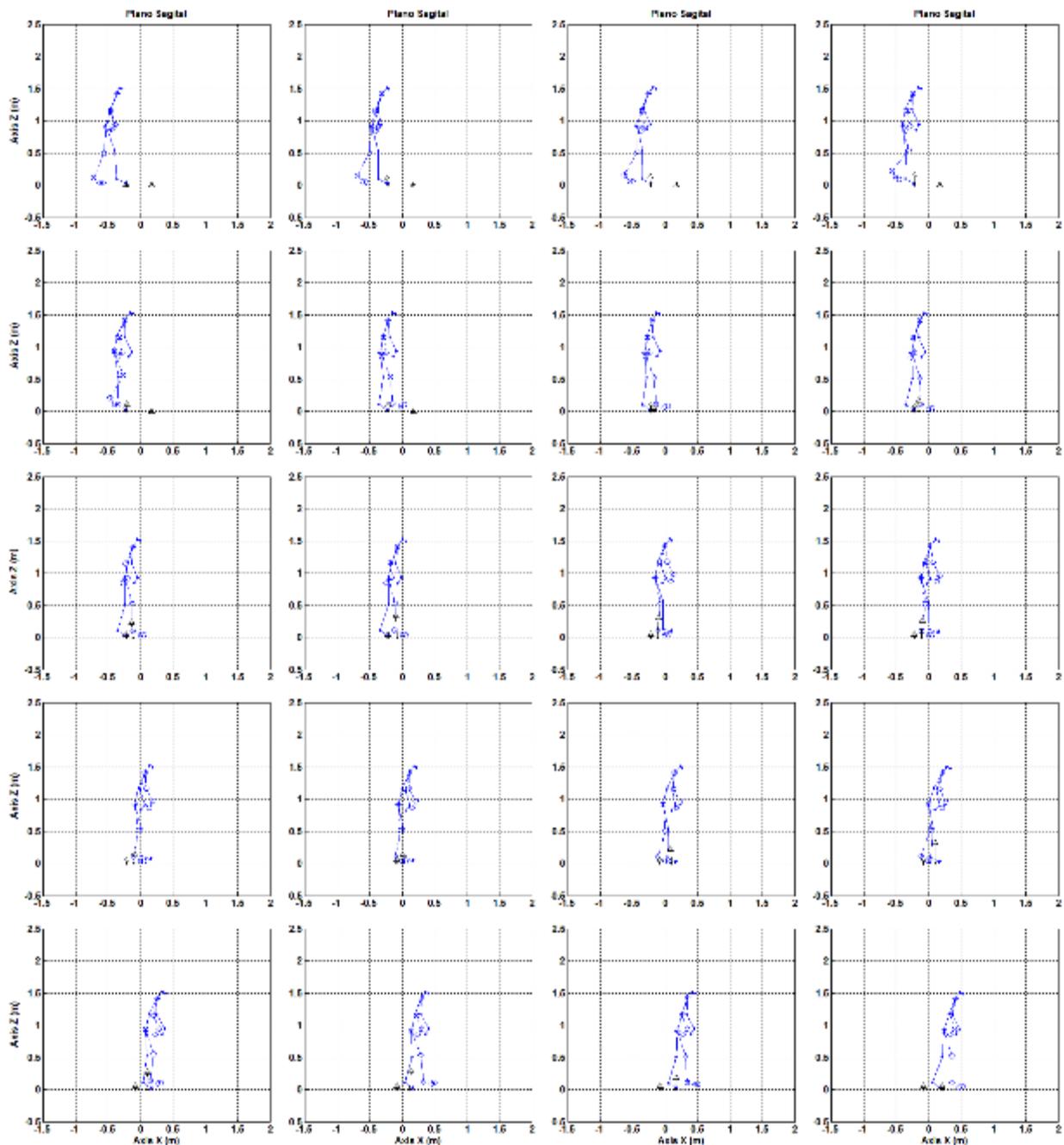


Figura 79 Simulación Placas de Fuerza, en Paciente con órtesis.

sobrellevar el movimiento de la órtesis activa, que implica el torque que se genera en la rodilla.

Las variaciones que aparecen en los resultados de las *Figuras 80 y 81* son debido a que el modelo está planteado en el plano sagital, obviando los movimientos en el plano frontal y trasversal. Y además, las articulaciones tanto de la cadera, rodilla y tobillo son modeladas como articulaciones simples, sin desplazamiento fuera del plano sagital.

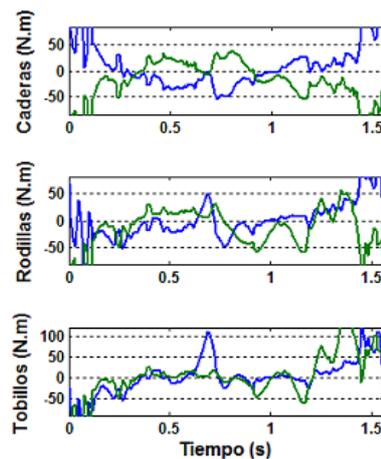


Figura 80 Momento de Flexión-Extensión en articulaciones inferiores de una persona sana. (Línea azul: pierna derecha, Línea verde: pierna izquierda)

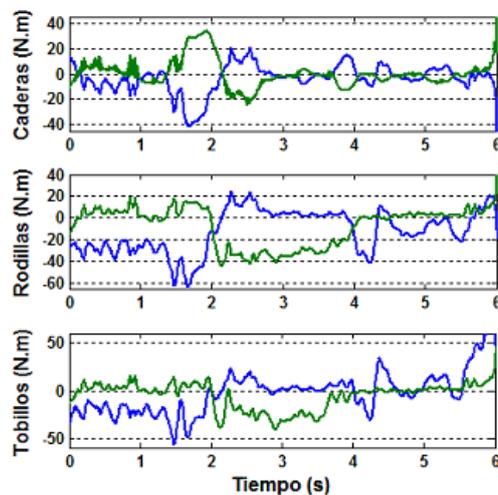


Figura 81 Momento de Flexión-Extensión en articulaciones inferiores del paciente. (Línea azul: pierna derecha, Línea verde: pierna izquierda)

Por último, el tiempo en los que se ejecuta los momentos, en el caso del paciente con órtesis, son más prolongados que en una marcha normal. Sin embargo, el modelo propuesto es una primera mejor aproximación para modelar la marcha, ya sea de una persona sana o un paciente.

Energía

Finalmente, se calcula la energía cinética, potencial gravitatoria y energía total para los casos de una persona sana y el paciente con órtesis.

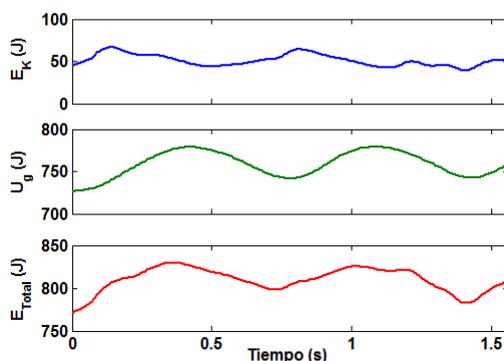


Figura 82 Cálculo Energético en una persona sana. Energía Cinética (Línea azul), Energía Potencial Gravitatoria (Línea verde), Energía Total (Línea roja)

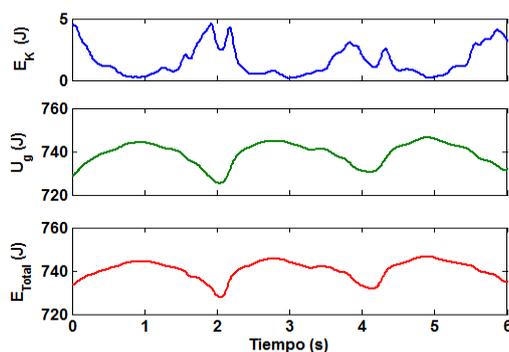


Figura 83 Cálculo Energético en un paciente. Energía Cinética (Línea azul), Energía Potencial Gravitatoria (Línea verde), Energía Total (Línea roja)

La energía cinética para ambas capturas, indudablemente, es diferente, ya que el paciente no tiene una velocidad de movimiento normal, puesto que no tiene las facultades necesarias para hacerlo. Sin embargo, la forma de la gráfica de las energías potenciales, para ambos casos, siguen la misma tendencia.

$$\begin{aligned} \text{Persona Sana: } \bar{E}_k &= 51.26 \text{ J} \\ \bar{U}_g &= 758.36 \text{ J} \\ \bar{E}_{Total} &= 809.62 \text{ J} \end{aligned}$$

$$\begin{aligned} \text{Paciente: } \bar{E}_k &= 1.57 \text{ J} \\ \bar{U}_g &= 738.86 \text{ J} \\ \bar{E}_{Total} &= 740.43 \text{ J} \end{aligned}$$

Para la persona sana en estudio, el promedio de la Energía Potencial (U_g) representa el 93.67% del promedio de la Energía total (E_T). Para el paciente, el promedio U_g es de 99.79% del promedio de E_T . Las cifras promedio de U_g para ambos casos, son cercanas y esto lo podemos interpretar como un mismo comportamiento en el movimiento de los segmentos corporales, principalmente de los segmentos inferiores, demostrando que la órtesis genera un movimiento más natural o parecido al de una persona sana.

CONCLUSIONES Y RECOMENDACIONES

- Con el análisis energético se puede determinar cualitativamente una aproximación de la marcha de un paciente con una órtesis a una marcha normal. Principalmente, en la energía potencial, la cual indica una mejor postura en la marcha del paciente.
- El control de la órtesis es adecuada para el tipo de lesión del paciente en estudio. Es importante que este control sea adaptable para otros tipos de marcha, independiente para cada paciente.
- El uso de la órtesis activa mejora la calidad de la marcha en un paciente, frente al uso de una órtesis pasiva. Ya que se puede monitorear la adaptación del paciente y ayuda a entender la interacción entre paciente y órtesis. Con esta ventaja, se podría generar un plan de negocio para poner al mercado órtesis activas de bajo costo.
- La técnica utilizada para modelar los segmentos corporales y articulaciones, es un método válido para el análisis de la marcha. Matlab se muestra como una herramienta potente para realizar los cálculos necesarios y resolver los diferentes problemas cinemáticos y dinámicos.
- El uso de simulaciones permite estudiar fenómenos o eventos los cuales no ocurren frecuentemente, son costosas reproducir más de una vez, o como en el caso de esta investigación, la molestia del paciente de caminar varias veces, provocando la fatiga de la persona.
- El siguiente paso para esta investigación, es trabajar con el modelo en 3D, incluso agregando un modelo de contacto pie-suelo para mejorar los resultados en la Dinámica Inversa.
- Plantear nuevos escenarios de acción para la órtesis activa, ya no solo caminar en un lugar plano, sino en situaciones como subir por una rampa, subir escalones, etc.
- El uso de la técnica de modelamiento y captura de movimiento para estudios de patologías en la marcha, así como también, actividades deportivas, como levantamiento de pesas, para el estudio de posibles lesiones y estrategias de prevención.
- Incluir dentro del modelamiento el uso de las muletas, para estudios posteriores a este, ya que es necesario incluir las reacciones de los extremos para realizar la dinámica inversa.

BIBLIOGRAFÍA

- Bureau of Labor Statistics. (2008).
- Callacná, R. (2005). *Aplicación de dinámica orinetada al objeto en el ámbito de movimiento de vehículos guiados*. Tesis de Pregrado, Piura.
- Christopher L. Vaughan, Brian L. David, Jeremy C. O'Connor. (1999). *Dynamics of Human Gait* (2 ed.). Cape Town, South Africa.
- Cuadrado Aranda, J., Lugrís Armesto, U., Alonso Sánchez, F., & Font-Llagunes, J. (2012). La Aplicación de técnicas de dinámica multicuerpo al diseño de órtesis activas para ayuda a la marcha. *XIX Congreso Nacional de Ingeniería Mecánica, España*. Castellón.
- Cuadrado, J. (1999). *Cinemática y Dinámica de Máquinas y Mecanismos por Computador*. Ferrol, La Coruña, España.
- García de Jalón, J., & Bayo, E. (1994). *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time challenge*. New York: Springer-Verlag.
- Lugrís, U., Carlín, J., Luaces, A., & Cuadrado, J. (2013). *Gait Analysis System for Spinal Cord Injured Subjects Assited by Active Orthoses and Crutches*. University of La Coruña, Escuela Politécnica Superior, Ferrol-España.
- Pàmies Vilà, R. (2012). *Application of Multibody Dynamics Techniques to the Analysis of Human Gait*. A thesis submitted for the degree of Doctor for the Universitat, UNIVERSITAT POLITÈCNICA DE CATALUNYA, Biomedical Engineering Doctoral Programme, Barcelona.
- Sears, F. W., Zemansky, M. W., y Young, H. D. (2009). *Física universitaria* (Vol. 1). México: Pearson.
- Silva, & Ambrósio, J. (n.d.). *Kinematic Data Consistency in the Inverser Dynamic Analysis of Biomechanical Systems*. Master Thesis Preparation.
- Vaughan, C. L., Davis, B. L., & O'Connor, J. C. (1999). *Dynamics of Human Gait* (2nd ed.). (C. Vaughan, Ed.) Cape Town: Kiboho Publishers.
- Whittle, M. W. (2007). *Gait Analysis* (4 ed.). Philadelphia: Butterworth Heinemann Elsevier.
- Winter, D. A. (2005). *Biomechanics and motor control of human movement* (3 ed.). New Jersey: John Wiley & Sons.

ANEXOS

A.1 Características de la sala de captura y el equipo

La sala de capturas utilizada para este estudio consta de 12 cámaras instaladas sobre una estructura metálica de perfiles de aluminio. Dichas cámaras están colocadas y orientadas en forma óptima para enfocar la totalidad del espacio de captura, evitando puntos ciegos en caso de que algún marcador pudiese quedar tapado por el cuerpo.

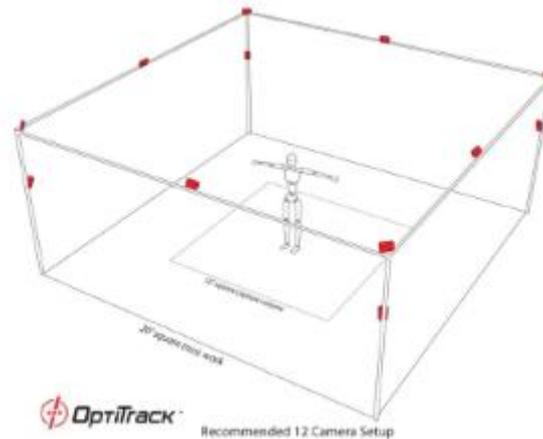


Figura A.1 1 Ubicación de cámaras

Se tiene instaladas dos placas de fuerzas en una tarima de madera, de manera que las placas y la superficie de la tarima estén a un mismo nivel para que la diferencia de altura no afecte a la marcha.

Se puede configurar las placas de tal manera que pueda tomar medidas de acuerdo a las características de la marcha; es decir, para una marcha normal, las placas se pueden disponer una a continuación de otra, incluso con un espacio entre ellas, y para una marcha de un paciente con órtesis, las placas se disponen una al lado de otra, ya que el paso de este tipo de persona son más pequeños que lo normal.

Las cámaras están conectadas a un computador dentro del laboratorio, donde está instalado el software específico de cada dispositivo y desde ese computador se controla y monitorea la captura.

Software de captura

OPTITRACK ARENA

Es el software empleado para controlar las cámaras para la captura de movimiento. Este tipo de aplicación es de animación gráfica ya que permite capturar movimientos naturales de personajes reales y dispone de la posibilidad de temporizar la captura para que el propio operador pueda ser el actor, lo que permite mayor flexibilidad. Otra de las ventajas es que se puede exportar los datos obtenidos de cada captura a otros programas, para posterior análisis.

Características clases y ventajas

- **Control programable:** temporizador al momento de la captura.
- **Sencilla configuración:** capacidad de conectar hasta 24 cámaras y obtener volúmenes de captura hasta 36 m². El tiempo de calibración suele ser menos de 5 minutos.
- **Captura de datos en tiempo real:** permite visualizar el movimiento en tiempo real.
- **Flexibilidad para la exportación de datos:** las exportaciones en formatos habituales como BVH, C3D y en tiempo real a MotionBuilder 7.5.

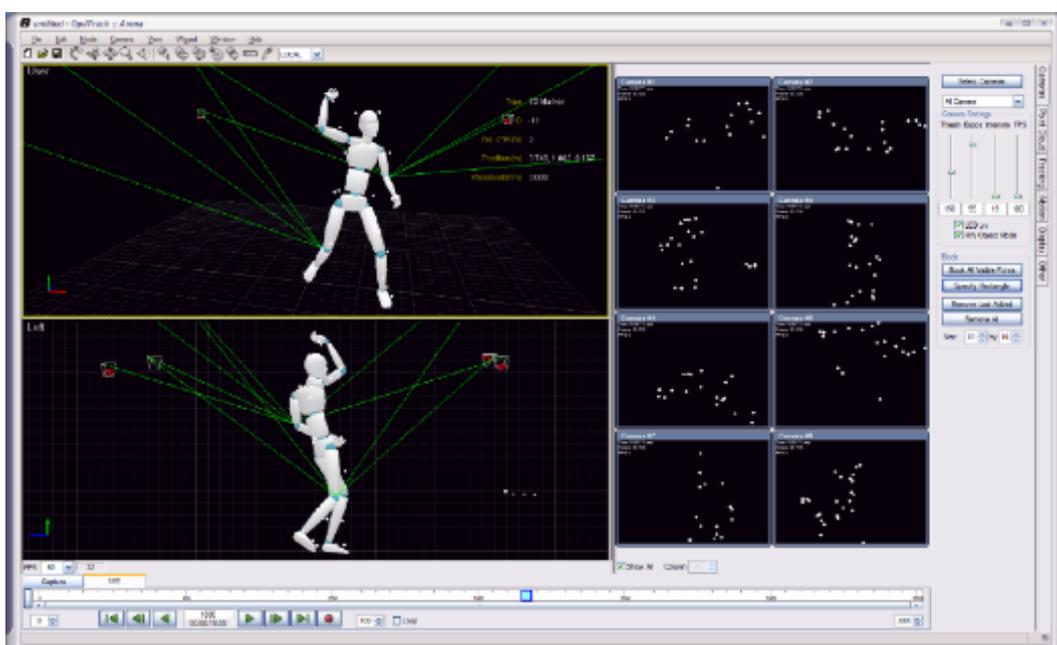


Figura A.1 2 Screenshot software Optitrack Arena

AMTI NETFORCE

Software empleado para la adquisición de datos medidos por las placas de fuerza AMTI. Puede utilizarse para aplicaciones de biomecánica y mediciones industriales. Netforce puede ejecutarse en varios ordenadores. El programa es sencillo e intuitivo que permite configurar las placas y obtener fuerzas y momentos ejercidos sobre las mismas. Los datos pueden ser importados para su análisis o utilizarse como variables de otros programas.

Características

- Visualización de datos en tiempo real.
- Ideal para la marcha y el equilibrio.
- Almacenamiento y exportación de datos.
- Soporta Ethernet, RS-232 y DT3002.

EVENTGHOST (Versión 0.3.7)

Software libre que permite ejecutar los programas de forma automática. EventoGhost ejecuta una secuencia de pasos, en las que va activando los botones de cada uno de los programas. Este programa permite tener un control remoto del equipo o activar simultáneamente varios programas.

MATLAB

Matlab es un software de cálculo matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio, parecido a C++. Las prestaciones básicas de este software se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otro lenguajes y con otros dispositivos de hardware.

Equipo empleado

Cámaras

Nº de cámaras: 12

Modelo: OptiTrak Flex: V100

Resolución: 640x480

Frecuencia: 100Hz

Número de LED's 26



Placas de fuerza

Modelo: AccuGait

Dimensiones: 50x50 cm

Altura: 4,4 cm

Fuerza máxima soportada: 3 000 N



Sujeciones de cámaras

Se utilizan para fijar las cámaras a la estructura tubular montada en el laboratorio, permiten orientar la cámara en la dirección deseada.



Traje de captura OptiTrack

Traje compuesto por un pantalón y una chaqueta de tejidos elásticos junto a una gorra donde se pueden adherir unos velcros, que soportan los marcadores.



Equipos de Calibración

Compuesto por dos herramientas:

Una barra extensible con un marcador en el extremo, que se emplea para comprobar el volumen de captura y realizar el calibrado.

Y una escuadra que se emplea para situar el origen y los ejes globales.



Marcadores

Marcadores reflectantes

Diámetro = 9,5 mm

Disponen de una base de velcro para pegarlos al traje.



A.2 Presupuesto para equipar una sala de capturas

		€/unidad	Total
Captura óptica			
Cámara V100 y accesorios	12		10.000,00 €
Cableado	1		239,40 €
Hub USB	2		30,00 €
OptiHub	1		231,93 €
Bolsa markers	1		240,00 €
Traje	1		215,00 €
Estructura	1		1.000,00 €
TrackingTools	1		618,48 €
Suelo			
Placa AccuGait	2	9.615,02 €	19.230,04 €
Caja conexión	1	2.133,58 €	2.133,58 €
Tarima	1	700,00 €	700,00 €
Muletas			
Muleta	2	4,25 €	8,50 €
Roseta extensométrica 90°	12	25,57 €	306,85 €
Banda extensométrica recta	4	17,70 €	70,80 €
Bobina cable 4 conductores (30m)	1	99,36 €	99,36 €
Bobina cable 3 conductores (30m)	1	90,51 €	90,51 €
Acc. extensometría (adhesivos etc.)	1	413,00 €	413,00 €
Adquisición muletas			
Amplificador RDP 628	8	637,20 €	5.097,60 €
Fuente alimentación RDP 631	1	560,50 €	560,50 €
Chasis RDP 604	1	1.062,00 €	1.062,00 €
Tarjeta de adquisición USB-6212	1	1.560,17 €	1.560,17 €
		TOTAL	43.907,71 €

A.3 Presupuesto para la elaboración de un prototipo de órtesis activa

Razón	Medida	Precio unitario	Cantidad	Total
-------	--------	-----------------	----------	-------

Aluminio 2011	kg	7,60 €	0,5	3,80 €
Acero S235	kg	2,00 €	0,01	0,02 €
Acero inoxidable X20Cr13	kg	4,30 €	0,7	3,01 €
Mecanizado soporte motor	horas	50,00 €	2	100,00 €
Mecanizado soporte muslo	horas	50,00 €	4	200,00 €
Mecanizado soporte pantorrilla	horas	50,00 €	3	150,00 €
Mecanizado eje articulación	horas	50,00 €	3,5	175,00 €
Mecanizado cala regulación	horas	30,00 €	1	30,00 €
Motor	unidades	115,85 €	1	115,85 €
Reductor	unidades	373,37 €	1	373,37 €
Mecanizado eje reductor	horas	50,00 €	0,5	25,00 €
Sistema de bloqueo	unidades	1.500,00 €	1	1.500,00 €
Encoder	unidades	63,01 €	1	63,01 €
Cojinete	unidades	6,40 €	1	6,40 €
Pareja engranajes cónicos	unidades	62,35 €	1	62,35 €
Mecanizado engranajes	horas	50,00 €	1,5	75,00 €
Elementos normalizados	unidades	5,00 €	1	5,00 €
Material esqueleto (AFO+titanio)	unidades	2.000,00 €	1	2.000,00 €
Construcción esqueleto	horas	50,00 €	6	300,00 €
Controladora	unidades	1.460,00 €	1	1.460,00 €
Batería de alto rendimiento	unidades	290,00 €	1	290,00 €
Giróscopo	unidades	200,00 €	1	200,00 €
Sensor plantar	unidades	33,00 €	3	99,00 €
Cableado, conectores y otros	unidades	140,00 €	1	140,00 €
TOTAL				7.376,81 €

A.4 Técnicas de Captura y Medición

Captura de movimiento

Son distintos métodos los que se han utilizado hasta el día de hoy en la captura de movimiento para poder estudiar la marcha humana. Para conocer la posición del cuerpo a lo largo del movimiento se han empleado, por ejemplo, grabación por medio de una

cámara de video normal, la secuencia de la marcha y representando en un mismo gráfico la posición del cuerpo en cada intervalo de tiempo.

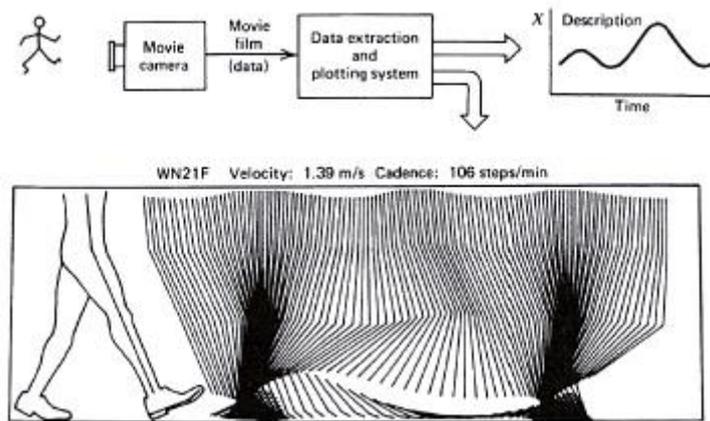


Figura A.4 1 Sistema para la representación del movimiento en una sola pierna en el plano sagital.

El problema principal es la forma de automatizar el proceso y digitalizar los datos para poder trabajar con ellos. Es por eso que se describe dos tipos de mecanismos para el estudio del movimiento: técnicas directas y sistemas ópticos.

Técnicas de medición directa

- **Goniómetros**

Son instrumentos de medida que se colocan en las articulaciones para medir el giro producido en estas articulaciones. El instrumento se coloca de forma que el eje de giro esté alineado con el eje de giro de la articulación. Pueden utilizar un potenciómetro o codificador digital.

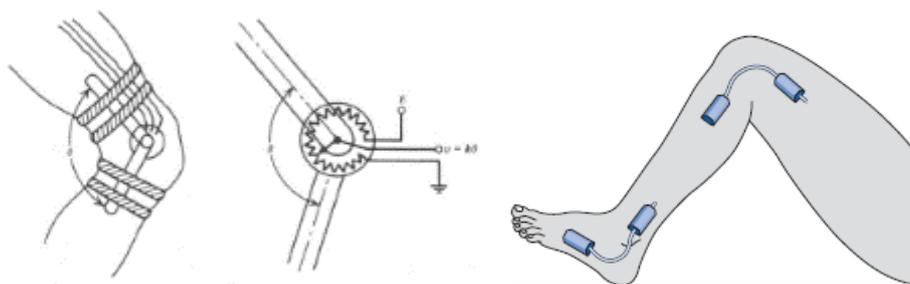


Figura A.4 2 Diferentes tipos de goniómetros. Goniómetro en base a un potenciómetro (Izquierda). Goniómetro flexible (Derecha).

potenciómetro: es una resistencia que varía según el ángulo de giro, este ángulo se relaciona con el potencial de salida tras el paso de la corriente eléctrica.

Codificador digital: dispositivo que genera impulsos digitales al detectar el giro de un disco patrón.

- **Acelerómetros**

Dispositivos que miden la aceleración. Utilizan un transductor de presión que mide la fuerza ejercida por una pequeña masa cuando se produce la aceleración de la misma, $F=m.a$. El transductor devuelve una señal de tensión que es directamente proporcional a la aceleración producida.

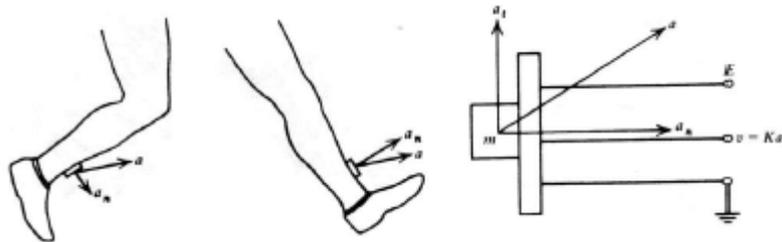


Figura A.4 3 Funcionamiento del acelerómetro

La aceleración medida por el acelerómetro es la correspondiente en la dirección normal, y para poder calcular la aceleración total, es necesario conocer la orientación del dispositivo en el espacio. Actualmente se emplean acelerómetros triaxiales y giróscopos en numerosos dispositivos electrónicos como en teléfonos móviles y mandos de video juegos.

- **Giróscopo**

Es un dispositivo micro-electromecánico que proporciona de forma precisa su orientación en el espacio en un sistema de referencia absoluto. Está formado por 3 sensores de velocidad angular con los que monitorea los cambios rápidos en la orientación. También utiliza sensores de aceleración y norte magnético para proporcionar una referencia estable. Con la conexión a un computador, y con el software adecuado, proporciona en tiempo real, de manera precisa y estable en el tiempo, la orientación en el espacio.

El hardware utilizado puede ser con conexión inalámbrico Bluetooth o por cable USB.

Sistemas ópticos

Es importante para el observador del movimiento, tener un seguimiento visual de la marcha. Debido a la complejidad de muchos movimientos, el único sistema que puede capturar todos los datos es un sistema de captura de video.

Nos limitaremos en mencionar tres sistemas de captura de imágenes: cámaras cinematográficas, de video y sistema optoelectrónicos.

- **Cámaras cinematográficas**

Suelen utilizarse para visualizar el movimiento del cuerpo en 2D. Existen una variedad de tamaños de película, desde los 8mm hasta los 70mm. Para efectos del estudio de la marcha, las películas de 16mm son adecuadas ya que cumplen con las características de ser económicas, manejables y ofrecen una resolución adecuada, además de ser capaces de grabar a mayor velocidad.

- **Videocámaras**

Las videocámaras, a diferencia de las cámaras cinematográficas, usa una frecuencia fija, por ejemplo, una frecuencia de 50 ó 60 Hz. Existen algunos problemas cuando se intenta trabajar con las cámaras de video tradicionales. La fuerte señal que refleja un marcador produce una distintiva imagen circular cuando éste está quieto. Sin embargo, cuando se mueve rápido la imagen se registra de forma borrosa y se produce una especie de rastro.

Posteriormente, se desarrolló la cámara de infrarrojos, que no capta la luz visible y no se ve influenciada por la reflexión de otros objetos que no sean los marcadores, facilitando las capturas.

- **Marcadores pasivos**

Algunos sistemas de tracking óptico utilizan marcadores planos a modo de pegatinas, que requieren mucho tiempo de procesamiento de imagen. La calidad del seguimiento suele ser sensible a los cambios de condiciones de luz ambiental. Los marcadores pasivos utilizados son catadióptricos. Es decir, son reflectores de la iluminación IR en la dirección de la fuente.

- **Marcadores activos**

Los marcadores activos tienen la característica de emitir luz infrarroja (IR) propia de LEDs.

- LED estándar: pueden ser cubierto de policarbonato para su protección. Estos marcadores son simples y robustos. Tiene una visibilidad buena a larga distancia. El rango angular de los mismos es de $\pm 60^\circ$.
- LED individual con difusor: se emplean para una óptima visibilidad en el rango angular; sin embargo, se limita la distancia entre el marcador y la cámara de seguimiento.

- **Técnicas Optoeléctricas**

Los sistemas de imagen optoeléctricos son tecnología de los últimos años, esta tecnología tiene ciertas ventajas sobre las cámaras cinematográficas y de video. El desarrollo más reciente de este sistema ha evolucionado en una cámara de 3D llamada OPTOTRAK. Este sistema consiste en tres cámaras montadas en línea en un bastidor. Las cámaras de los extremos tienen sus lentes apuntando ligeramente hacia adentro, y la del centro, en posición horizontal.

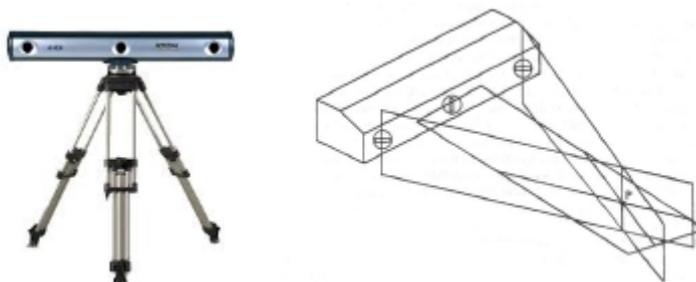


Figura A.4 4 OptoTrak

Ventajas:

- Todos los datos obtenidos están referidos a un sistema de ejes globales.
- La mayoría no están limitados a un número máximo de marcadores.
- Son sistemas razonablemente económicos.
- Las cámaras de video y cinematográficas pueden ser útiles para reproducir el movimiento de forma visual, para propósitos educativos o análisis cualitativos del movimiento global del cuerpo.

Desventajas:

- El tiempo necesario para el tratamiento de las imágenes de video puede ser elevado y pueden producirse errores.
- Cuando se usan marcadores activos, la colocación de los LEDs y su cableado puede ser costosa y el número de ellos suele estar limitado.
- Algunos sistemas de imagen no pueden ser utilizados con luz natural.

A.5 Programación de problemas cinemáticos y dinámicos en MATLAB

CUADRILÁTERO ARTICULADO

```
function iterar
%solucion que resuelve la posicion del problema del cuadrilatero
articulado

%constantes
```

```

xa=0; ya=0; xb=10; yb=0;
L1=2; L2=8; L3=5;

%valor de la variable del grado de libertad
alfa0=pi/3; delta_alfa=5*pi/180; n=72;

%velocidad y aceleración del grado de libertad
alfap=1; alfas=0;

%variables iniciales
x1=0; x2=10; y1=2; y2=5; alfa=alfa0;

%valor inicial de q
q=[x1 y1 x2 y2 alfa]';

for i=1:3*(n+1)

    alfa=alfa0+(i-1)*delta_alfa; q(5)=alfa;

    % PROBLEMA DE POSICION

    if abs(tan(alfa)) < 1
        phi=[(x1-xa)^2+(y1-ya)^2-L1^2;
              (x2-x1)^2+(y2-y1)^2-L2^2;
              (x2-xb)^2+(y2-yb)^2-L3^2;
              (y1-ya)-L1*sin(alfa)];
    else
        phi=[(x1-xa)^2+(y1-ya)^2-L1^2;
              (x2-x1)^2+(y2-y1)^2-L2^2;
              (x2-xb)^2+(y2-yb)^2-L3^2;
              (x1-xa)-L1*cos(alfa)];
    end
    error=norm(phi);

    while error > 10^-8

        if abs(tan(alfa)) < 1
            Phiq=[2*(x1-xa) 2*(y1-ya) 0 0 0;
                  -2*(x2-x1) -2*(y2-y1) 2*(x2-x1) 2*(y2-y1) 0;
                  0 0 2*(x2-xb) 2*(y2-yb) 0;
                  0 1 0 0 -L1*cos(alfa)];
        else
            Phiq=[2*(x1-xa) 2*(y1-ya) 0 0 0;
                  -2*(x2-x1) -2*(y2-y1) 2*(x2-x1) 2*(y2-y1) 0;
                  0 0 2*(x2-xb) 2*(y2-yb) 0;
                  1 0 0 0 L1*sin(alfa)];
        end

        A=Phiq(:,1:4);
        b=-phi;
        dq=A\b;
        dq=[dq;0];
        q=q+dq;
        x1=q(1); y1=q(2); x2=q(3); y2=q(4); alfa=q(5);

        if abs(tan(alfa)) < 1
            phi=[(x1-xa)^2+(y1-ya)^2-L1^2;

```

```

        (x2-x1)^2+(y2-y1)^2-L2^2;
        (x2-xb)^2+(y2-yb)^2-L3^2;
        (y1-ya)-L1*sin(alfa)];
else
    phi=[(x1-xa)^2+(y1-ya)^2-L1^2;
        (x2-x1)^2+(y2-y1)^2-L2^2;
        (x2-xb)^2+(y2-yb)^2-L3^2;
        (x1-xa)-L1*cos(alfa)];
end
error=norm(phi);

end

figure(1); clf;
hold on

% Puntos fijos.
circunferencia(xa, ya, 0.08*L1, 'r');
circunferencia(xa, ya, 0.05*L1, 'r');
circunferencia(xb, yb, 0.08*L1, 'r');
circunferencia(xb, yb, 0.05*L1, 'r');

% Puntos moviles.
circunferencia(x1, y1, 0.05*L1, 'r');
circunferencia(x2, y2, 0.05*L1, 'r');

% Barras.
x=[xa x1 x2 xb]; y=[ya y1 y2 yb]; plot(x,y, 'b');

hold off
axis([-2*L1, xb+2*L1, -0.4*(xb+4*L1), 0.6*(xb+4*L1)]);

% PROBLEMA DE VELOCIDAD

if abs(tan(alfa)) < 1
    Phiq=[2*(x1-xa) 2*(y1-ya) 0 0 0;
        -2*(x2-x1) -2*(y2-y1) 2*(x2-x1) 2*(y2-y1) 0;
        0 0 2*(x2-xb) 2*(y2-yb) 0;
        0 1 0 0 -L1*cos(alfa)];
else
    Phiq=[2*(x1-xa) 2*(y1-ya) 0 0 0;
        -2*(x2-x1) -2*(y2-y1) 2*(x2-x1) 2*(y2-y1) 0;
        0 0 2*(x2-xb) 2*(y2-yb) 0;
        1 0 0 0 L1*sin(alfa)];
end

A=Phiq(:,1:4);
b=-alfap*Phiq(:,5);
qp=A\b;
qp=[qp;alfap];
x1p=qp(1); y1p=qp(2); x2p=qp(3); y2p=qp(4); alfap=qp(5);

% PROBLEMA DE ACELERACION

if abs(tan(alfa)) < 1
    phiqpqp=[2*(x1p^2+y1p^2);
        2*((x2p-x1p)^2+(y2p-y1p)^2);
        2*(x2p^2+y2p^2);

```

```

        L1*(alfap^2)*sin(alfa)];
else
    phiqqqp=[2*(x1p^2+y1p^2);
             2*((x2p-x1p)^2+(y2p-y1p)^2);
             2*(x2p^2+y2p^2);
             L1*(alfap^2)*cos(alfa)];
end

b=-phiqqqp-alfas*Phiq(:,5);
qs=A\b;
qs=[qs;alfas];
x1s=qs(1); y1s=qs(2); x2s=qs(3); y2s=qs(4); alfas=qs(5);

end

end

function circunferencia(xC,yC,r,color)
%función que dibuja una circunferencia con centro en %(xC,yC), radio r y
%color color

%constantes
n=32;

%valores iniciales
theta=0;

for i=1:n+1

    x(i) = xC+r*cos(theta);
    y(i) = yC+r*sin(theta);

    theta = theta + 2*pi/n;

end

plot(x,y,color);

```

MANIPULADOR PLANO

```

function manipulador

%solución que resuelve el problema del manipulador plano

%constantes
L=2;
xA=-L; yA=-L/sqrt(3); xB=L; yB=-L/sqrt(3); xC=0; yC=2*L/sqrt(3);

```

```

r=0.25; n=72;

%valor de la variable de los grados de libertad
x4=0; y4=r;

%velocidad y aceleración de las variables de los grados de libertad
x4p=1; y4p=1; x4s=8; y4s=8;

%variables iniciales
x1=0; y1=-1; x2=1; y2=2; x3=-1; y3=1; alfa0=0; delta_alfa=2*pi/n;
alfa=alfa0;

%valor inicial de q
q=[x1 y1 x2 y2 x3 y3 x4 y4]';

for i=1:2*n+1

    alfa=alfa0+(i-1)*delta_alfa;

    x4=r*sin(alfa);
    y4=r*cos(alfa);

    q(7)=x4;
    q(8)=y4;

    % PROBLEMA DE POSICION

    phi=[(x1-xA)^2+(y1-yA)^2-L^2;
          (x2-xB)^2+(y2-yB)^2-L^2;
          (x3-xC)^2+(y3-yC)^2-L^2;
          (x4-x1)^2+(y4-y1)^2-(L/sqrt(3))^2;
          (x4-x2)^2+(y4-y2)^2-(L/sqrt(3))^2;
          (x4-x3)^2+(y4-y3)^2-(L/sqrt(3))^2];

    error=norm(phi);

    while error > 10^-8

        Phiq=[2*(x1-xA) 2*(y1-yA) 0 0 0 0 0 0;
              0 0 2*(x2-xB) 2*(y2-yB) 0 0 0 0;
              0 0 0 0 2*(x3-xC) 2*(y3-yC) 0 0;
              -2*(x4-x1) -2*(y4-y1) 0 0 0 0 2*(x4-x1) 2*(y4-y1);
              0 0 -2*(x4-x2) -2*(y4-y2) 0 0 2*(x4-x2) 2*(y4-y2);
              0 0 0 0 -2*(x4-x3) -2*(y4-y3) 2*(x4-x3) 2*(y4-y3)];

        A=Phiq(:,1:6);
        b=-phi;
        dq=A\b;
        dq=[dq;0];
        dq=[dq;0];
        q=q+dq;
        x1=q(1); y1=q(2); x2=q(3); y2=q(4); x3=q(5); y3=q(6); x4=q(7);
        y4=q(8);

```

```

    phi=[(x1-xA)^2+(y1-yA)^2-L^2;
          (x2-xB)^2+(y2-yB)^2-L^2;
          (x3-xC)^2+(y3-yC)^2-L^2;
          (x4-x1)^2+(y4-y1)^2-(L/sqrt(3))^2;
          (x4-x2)^2+(y4-y2)^2-(L/sqrt(3))^2;
          (x4-x3)^2+(y4-y3)^2-(L/sqrt(3))^2];

    error=norm(phi);

end

figure(1); clf;
hold on

% Puntos fijos.
circunferencia(xA,yA,0.08*L,'r');
circunferencia(xA,yA,0.05*L,'r');
circunferencia(xB,yB,0.08*L,'r');
circunferencia(xB,yB,0.05*L,'r');
circunferencia(xC,yC,0.08*L,'r');
circunferencia(xC,yC,0.05*L,'r');

% Puntos moviles.
circunferencia(x1,y1,0.05*L,'r');
circunferencia(x2,y2,0.05*L,'r');
circunferencia(x3,y3,0.05*L,'r');
circunferencia(x4,y4,0.05*L,'r');

%trayectoria
circunferencia(0,0,r,'g');

% Barras.
barrax=[xA x1 x4]; barray=[yA y1 y4]; plot(barrax,baray,'b');
barra2x=[xB x2 x4]; barra2y=[yB y2 y4];
plot(barra2x,barra2y,'b');
barra3x=[xC x3 x4]; barra3y=[yC y3 y4];
plot(barra3x,barra3y,'b');

hold off

axis([-0.5+xA,0.5+xB,-0.25+yA,0.25+yC]);

% PROBLEMA DE VELOCIDAD

x4p=2*pi*r*cos(alfa);
y4p=-2*pi*r*sin(alfa);

Phiq=[2*(x1-xA) 2*(y1-yA) 0 0 0 0 0 0;
       0 0 2*(x2-xB) 2*(y2-yB) 0 0 0 0;
       0 0 0 0 2*(x3-xC) 2*(y3-yC) 0 0;
       -2*(x4-x1) -2*(y4-y1) 0 0 0 0 2*(x4-x1) 2*(y4-y1);
       0 0 -2*(x4-x2) -2*(y4-y2) 0 0 2*(x4-x2) 2*(y4-y2);
       0 0 0 0 -2*(x4-x3) -2*(y4-y3) 2*(x4-x3) 2*(y4-y3)];

A=Phiq(:,1:6);

```

```

b=- (x4p*Phiq(:,7)+y4p*Phiq(:,8));
qp=A\b;
qp=[qp;x4p];
qp=[qp;y4p];
x1p=qp(1); y1p=qp(2); x2p=qp(3); y2p=qp(4); x3p=qp(5);
y3p=qp(6); x4p=qp(7); y4p=qp(8);

v1=[x1p y1p];

mv1 = norm(v1);

x1pg(i)=x1p;
y1pg(i)=y1p;
modulo_vell(i)=mv1;
tg(i)=alfa/(2*pi);

% PROBLEMA DE ACELERACION
vel4=sqrt(x4p^2+y4p^2);
x4s=-vel4^2/r*sin(alfa);
y4s=-vel4^2/r*cos(alfa);

phiqqqp=[2*(x1p^2+y1p^2);
          2*(x2p^2+y2p^2);
          2*(x3p^2+y3p^2);
          2*((x4p-x1p)^2+(y4p-y1p)^2);
          2*((x4p-x2p)^2+(y4p-y2p)^2);
          2*((x4p-x3p)^2+(y4p-y3p)^2)];

b=-phiqqqp-x4s*Phiq(:,7)-y4s*Phiq(:,8);
qs=A\b;
qs=[qs;x4s];
qs=[qs;y4s];
x1s=qs(1); y1s=qs(2); x2s=qs(3); y2s=qs(4); x3s=qs(5);
y3s=qs(6); x4s=qs(7); y4s=qs(8);

x1sg(i)=x1s;
y1sg(i)=y1s;

a1=[x1s y1s];

ma1 = norm(a1);

modulo_ace11(i)=ma1;

end

figure(2); clf;
subplot(1,2,1),plot(tg,x1pg),xlabel('Tiempo'),
ylabel('Vx'),title('Componente de la Velocidad en X');
subplot(1,2,2), plot(tg,y1pg),xlabel('Tiempo'),
ylabel('Vy'),title('Componente de la Velocidad en Y');

figure(3); clf;
d=derivar(tg,x1pg);

```

```

figure(4);clf;
subplot(1,2,1),plot(tg,x1sg),xlabel('Tiempo'),
ylabel('Ax'),title('Componente de la Acelereación en X');
subplot(1,2,2), plot(tg,y1sg),xlabel('Tiempo'),
ylabel('Ay'),title('Componente de la Acelereación en Y');

figure(5); clf;
plot(tg,modulo_vell);xlabel('Tiempo'),ylabel('V'),title('Velocidad
vs Tiempo');

figure(6); clf;
plot(tg,modulo_acell);xlabel('Tiempo'),ylabel('A'),title('Acelerac
ión vs Tiempo');

RMS=(d-x1sg);

y=sqrt(mean(RMS.^2)) %muestra el RMS

```

```
end
```

```

function circunferencia(xC,yC,r,color)
%función que dibuja una circunferencia con centro en %(xC,yC), radio r y
%color color

%constantes
n=32;

%valores iniciales
theta=0;

for i=1:n+1

    x(i) = xC+r*cos(theta);
    y(i) = yC+r*sin(theta);

    theta = theta + 2*pi/n;

end

plot(x,y,color);

```

```

function [df]=derivar(X,Y)

%Funcion que permite derivar dy/dx

N=numel(X);

df(1)=(Y(2)-Y(1))/(X(2)-X(1));

```

```
df(N)=(Y(N)-Y(N-1))/(X(N)-X(N-1));  
  
for n=2:N-1  
  
df(n)=(Y(n+1)-Y(n-1))/(X(n+1)-X(n-1));  
  
end  
  
plot(X,df)  
  
end
```

MODELO COMPLETO

```
%Import captures data  
AM=importdata('AM.dat');  
r1=importdata('r1.dat');  
r0=importdata('r0.dat');  
MI=importdata('MI.dat');  
rg=importdata('rg.dat');  
fp11=importdata('FPL1.dat');
```

```

fpl2=importdata('FPL2.dat');
t=r0(:,1);

[r] = modificacion_r0(r0,AM,r1);

[l] = longitudes(r);

[q_proj] = qproj(r,l);

for i=1:1
markers(r,q_proj);
end

[qc] = qcomplete(q_proj,l,t);
qc_antes=qc;

[z] = zeta(qc);% vector de 20x1

%Filtrado de z
z_fil = z;
for i = 1:20
    for j = 1:3
        z_fil(:,i) = ssa(z_fil(:,i),9);
    end
end

%construccion de la nueva qc con valores filtrados
qc(:,1:2) = z_fil(:,1:2);
qc(:,39:56) = z_fil(:,3:20);

%consistencia cinemática con los valores filtrados de z de los valores
%independientes
for i = 1:length(t)
    [Phic,Phiqc] = restric_c(qc(i,:),l);
    error = norm(Phic); tol = 1e-14;
    while error > tol

        inc = 3:38;
        A = Phiqc'*Phiqc;
        B = -Phiqc'*Phic;
        deltaq = A(inc, inc)\B(inc);
        qc(i,inc) = qc(i,inc)+deltaq';

        [Phic, Phiqc] = restric_c(qc(i,:),l);

        error = norm(Phic);

    end

end

%Primera Derivada de z
z_fil_p = zeros(length(t),20);

%Segunda Derivada de z

```

```

z_fil_s = zeros(length(t),20);

for i = 1:20

    z_fil_p(:,i) = derivar(t,z_fil(:,i));
    z_fil_s(:,i) = derivar(t,z_fil_p(:,i));

end

%Calculo de qcp a partir de z_fil_p
qcp = zeros(length(t),56);
qcp(:,1:2) = z_fil_p(:,1:2);
qcp(:,39:56) = z_fil_p(:,3:20);

%Calculo de qcs a partir de z_fil_s
qcs = zeros(length(t),56);
qcs(:,1:2) = z_fil_s(:,1:2);
qcs(:,39:56) = z_fil_s(:,3:20);

%Cálculo phiqqqp
order = [1 2 1 3 3 4 4 5 3 6 6 7 7 8 3 9 9 10 10 11 2 12 12 13 13 14 14
15 2 16 16 17 17 18 18 19];
phiqqqp = zeros(36,1);
for i = 1:length(t)
[Phic,Phiqc] = restric_c(qc(i,:),1);

    inc = 3:38;
    A = Phiqc;
    B = -Phiqc(:, [1:2,39:56])*z_fil_p(i,:)' ;
    C = A(:,inc)\B;
    qcp(i,inc)=C';

    p = @(k) 2*k-1:2*k; m = @(c,g) qcp(c,p(order(g)));

%armado del vector phiqqqp(1:18,1)
a=0;
for j=1:2:36
    a=a+1;
    f = m(i,j)-m(i,j+1);
    phiqqqp(a,:)=2*(f*f');

end

%armado del vector phiqqqp(19:36,1)

phiqqqp(19,:) = -1(1)*qcp(i,39)^2*cos(qcp(i,39));
phiqqqp(20,:) = -1(2)*qcp(i,40)^2*cos(qcp(i,40));
phiqqqp(21,:) = -1(3)*qcp(i,41)^2*cos(qcp(i,41));
phiqqqp(22,:) = 1(4)*qcp(i,42)^2*sin(qcp(i,42));
phiqqqp(23,:) = 1(5)*qcp(i,43)^2*sin(qcp(i,43));
phiqqqp(24,:) = -1(6)*qcp(i,44)^2*cos(qcp(i,44));
phiqqqp(25,:) = -1(7)*qcp(i,45)^2*cos(qcp(i,45));
phiqqqp(26,:) = 1(8)*qcp(i,46)^2*sin(qcp(i,46));
phiqqqp(27,:) = -1(9)*qcp(i,47)^2*cos(qcp(i,47));
phiqqqp(28,:) = -1(10)*qcp(i,48)^2*cos(qcp(i,48));

```

```

    phiqpqp(29,:) = -1(11)*qcp(i,49)^2*cos(qcp(i,49));
    phiqpqp(30,:) = -1(12)*qcp(i,50)^2*cos(qcp(i,50));
    phiqpqp(31,:) = 1(13)*qcp(i,51)^2*sin(qcp(i,51));
    phiqpqp(32,:) = -1(13)*qcp(i,51)^2*cos(qcp(i,51));
    phiqpqp(33,:) = 1(14)*qcp(i,52)^2*sin(qcp(i,52));
    phiqpqp(34,:) = -1(14)*qcp(i,52)^2*cos(qcp(i,52));
    phiqpqp(35,:) = -1(15)*qcp(i,53)^2*cos(qcp(i,53));
    phiqpqp(36,:) = -1(16)*qcp(i,54)^2*cos(qcp(i,54));
    phiqpqp(37,:) = 1(17)*qcp(i,55)^2*sin(qcp(i,55));
    phiqpqp(38,:) = -1(17)*qcp(i,55)^2*cos(qcp(i,55));
    phiqpqp(39,:) = 1(18)*qcp(i,56)^2*sin(qcp(i,56));
    phiqpqp(40,:) = -1(18)*qcp(i,56)^2*cos(qcp(i,56));

    B = -phiqpqp-Phiqc(:, [1:2, 39:56])*z_fil_s(i,:)' ;
    C = A(:,inc)\B;
    qcs(i,inc)=C';
end

%Modulo velocidad
modulo_velocidad=zeros(length(t),19);
for i=1:length(t)

    for j=1:19
        modulo_velocidad(i,j)=sqrt((qcp(i,2*j-1))^2+(qcp(i,2*j))^2);
    end

end

figure(4);clf
plot(t,modulo_velocidad(:,1))

figure(7);clf
hold on
plot(t,modulo_velocidad(:,12))
xlabel('Tiempo(s)');
ylabel('Velocidad(m/s)');
title('Velocidad Rodilla Derecha');

plot(t,modulo_velocidad(:,16),'r')
hold off

% End velocidad

%Modulo aceleracion

modulo_aceleracion=zeros(length(t),19);
for i=1:length(t)

    for j=1:19
        modulo_aceleracion(i,j)=sqrt((qcs(i,2*j-1))^2+(qcs(i,2*j))^2);
    end

end

figure(10);clf
plot(t,modulo_aceleracion(:,1))

figure(11);clf

```

```

hold on
plot(t,modulo_aceleracion(:,12))
xlabel('Tiempo(s)');
ylabel('Aceleración(m/s^2)');
title('Aceleración Rodilla');

plot(t,modulo_aceleracion(:,16),'r')
hold off

%End Aceleracion

grafica_trayectorias(r,q_proj,qc);

figure(3);clf
subplot(7,2,1)
plot(t,qc(:,23),'b',t,qcp(:,23),'r',t,qcs(:,23),'g');
grid on
title('x12 vs t')
lighting gouraud
set(gcf,'Renderer','OpenGL')

subplot(7,2,2)
plot(t,qc(:,24),'b',t,qcp(:,24),'r',t,qcs(:,24),'g');
grid on
title('y12 vs t')
lighting gouraud

subplot(7,2,3)
plot(t,qc(:,31),'b',t,qcp(:,31),'r',t,qcs(:,31),'g');
grid on
title('x16 vs t')
lighting gouraud

subplot(7,2,4)
plot(t,qc(:,32),'b',t,qcp(:,32),'r',t,qcs(:,32),'g');
grid on
title('y16 vs t')
lighting gouraud

subplot(7,2,5)
plot(t,qc(:,25),'b',t,qcp(:,25),'r',t,qcs(:,25),'g');
grid on
title('x13 vs t')
lighting gouraud

subplot(7,2,6)
plot(t,qc(:,26),'b',t,qcp(:,26),'r',t,qcs(:,26),'g');
grid on
title('y13 vs t')
lighting gouraud

subplot(7,2,7)
plot(t,qc(:,33),'b',t,qcp(:,33),'r',t,qcs(:,33),'g');
grid on
title('x17 vs t')
lighting gouraud

subplot(7,2,8)

```

```

plot(t,qc(:,34),'b',t,qcp(:,34),'r',t,qcs(:,34),'g');
grid on
title('y17 vs t')
lighting gouraud

subplot(7,2,9:10)
plot(t,qc(:,50),'b',t,qcp(:,50),'r',t,qcs(:,50),'g');
grid on
title('theta 1')
lighting gouraud

subplot(7,2,11:12)
plot(t,qc(:,51),'b',t,qcp(:,51),'r',t,qcs(:,51),'g');
grid on
title('theta 2')
lighting gouraud

subplot(7,2,13:14)
plot(t,qc(:,52),'b',t,qcp(:,52),'r',t,qcs(:,52),'g');
grid on
title('theta 3')
lighting gouraud
legend('posicion','velocidad','aceleracion')

%Cálculo de matriz de masa
%masas de segmentos
m=masas(MI);
%CG locales de segmentos
cg_l=CG(rg);
[alfa,rg_g]=grafica_CG(qc,cg_l,l);

%Primera Derivada de rg_g

    rg_gp = zeros(length(t),36);

    for i = 1:36

        rg_gp(:,i) = derivar(t,rg_g);

    end

%Momento de inercia con respecto a un extremo
I=Inercia(MI,cg_l,m);

[M]=Matriz_de_masa(m,cg_l,I,l);

%Matriz de fuerzas
[Qc]=matriz_de_fuerzas(m,l,alfa);

%Dinámica Inversa
tic
[T]=Dinamica_Inversa(t,qc,qcs,z_fil_p,fp11,fp12,l,M,Qc);
toc

T(:,16)=ssa(T(:,16),3);
T(:,20)=ssa(T(:,20),3);

```

```

T(:,17)=ssa(T(:,17),3);
T(:,21)=ssa(T(:,21),3);
T(:,18)=ssa(T(:,18),3);
T(:,22)=ssa(T(:,22),3);

figure(12);
subplot(3,1,1)
plot(t,T(:, [16 20])/2);ylabel('Caderas (N.m)');ylim([-80 80]);xlim([0
1.56]);
subplot(3,1,2)
plot(t,T(:, [17 21])/2);ylabel('Rodillas (N.m)');ylim([-80 80]);xlim([0
1.56]);
subplot(3,1,3)
plot(t,T(:, [18 22])/2);xlabel('Tiempo (s)');ylabel('Tobillos
(N.m)');ylim([-80 80]);xlim([0 1.56]);

for i=1:3
grafica_fuerza_placas(r,fp11,fp12);
end

%Energía
ener_cine=zeros(length(t),1); ener_pot_grav=zeros(length(t),1);g=9.81;
ener_tot=zeros(length(t),1);
for i=1:length(t)
ener_cine(i,1)=0.5*qcp(i,:)*M*qcp(i,:);
for j=1:length(m)
ener_pot_grav(i,1)=ener_pot_grav(i,1)+m(j)*g*(rg_g(i,2*j));
end
ener_tot(i,1)=ener_cine(i,1)+ener_pot_grav(i,1);
end

figure(6);clf
subplot(3,1,1)
plot(t,ener_cine);title('Energía Cinética');xlabel('time
(s)');ylabel('Energy (J)');

subplot(3,1,2)
plot(t,ener_pot_grav);title('Energía Potencial
Gravitatoria');xlabel('time (s)');ylabel('Energy (J)');

subplot(3,1,3)
plot(t,ener_tot);title('Energía Total');xlabel('time
(s)');ylabel('Energy (J)');

function cg_1 = CG(rg)
%Funcion que permite determinar los centros de gravedad locales
cg_1= zeros(18,2);
rg(:,2)=[];%eliminando la columna Y, paso de 2D a 3D;

cg_1(1,:) = rg(1,:);
cg_1(2,:) = rg(2,:);
cg_1(3,:) = rg(3,:);

```

```

cg_1(4,:) = rg(4,:);
cg_1(5,:) = rg(5,:);
cg_1(6,:) = rg(6,:);
cg_1(7,:) = rg(7,:);
cg_1(8,:) = rg(8,:);
cg_1(9,:) = rg(9,:);
cg_1(10,:) = rg(10,:);
cg_1(11,:) = rg(11,:);
cg_1(12,:) = rg(12,:);
cg_1(13,:) = rg(13,:);
cg_1(14,:) = rg(17,:);
cg_1(15,:) = rg(14,:);
cg_1(16,:) = rg(15,:);
cg_1(17,:) = rg(16,:);
cg_1(18,:) = rg(18,:);

```

```
end
```

```
function [df]=derivar(X,Y)
```

```
%Funcion que permite derivar dy/dx
```

```
N=numel(X);
```

```
df(1)=(Y(2)-Y(1))/(X(2)-X(1));
```

```
df(N)=(Y(N)-Y(N-1))/(X(N)-X(N-1));
```

```
for n=2:N-1
```

```
% df(n)=(Y(n+1)-Y(n-1))/((X(n+1)-X(n-1)));
```

```
df(n)=(Y(n+1)-Y(n))/((X(n+1)-X(n)));
```

```
end
```

```
% plot(X,df)
```

```
end
```

```
function [T]= Dinamica_Inversa(t,qc,qs,zp,fp11,fp12,l,M,Qc)
```

```
T=zeros(length(qc),23);
```

```
Qm = zeros(length(qc),20);
```

```
xy = @(n) 2*n-1:2*n;
```

```
pv = @(w,v) w*[-v(2) v(1)];
```

```
ang = @(n) 38+n;
```

```

yp=zeros(length(qc),23);
order = [2 1 3 2 4 3 5 2 6 5 7 6 8 2 9 8 10 9 11 1 12 11 13 12 14 13 15 1
16 15 17 16 18 17];
B = zeros(23,20);
T_ast=zeros(length(qc),23);
W = zeros(23);
for i=1:length(qc)

    R = Matriz_R(qc(i,:),1);
    Qm(i,:) = R'*(M*qs(i,:)'-Qc);

    %Cálculo de la matriz B
    for j=1:20
        B(1:2,j)=R(xy(13),j)'+pv(R(ang(13),j),[fpl1(i,7) fpl1(i,8)]-
qc(i,xy(13))));
        B(3,j)=R(ang(13),j);
        B(4:5,j)=R(xy(17),j)'+pv(R(ang(17),j),[fpl2(i,7) fpl2(i,8)]-
qc(i,xy(17))));
        B(6,j)=R(ang(17),j);
        for m=1:17

            B(6+m,j)=R(ang(order(2*m-1)),j)-R(ang(order(2*m)),j);

        end

    end

    yp(i,:)=B*zp(i,:)';

    %Problema de minimización, con restriccion [W B;B' 0]*[T lambda]=[WT*;Qm]
    T_ast(i,1)=fpl1(i,1);
    T_ast(i,2)=fpl1(i,3);
    T_ast(i,3)=fpl1(i,5);%momento con respecto a Y
    T_ast(i,4)=fpl2(i,1);
    T_ast(i,5)=fpl2(i,3);
    T_ast(i,6)=fpl2(i,5);

    % W
    for m=1:6
        % W(m,m)=1/(sqrt(abs(T_ast(i,m)))+1e-17);
        W(m,m)=abs(T_ast(i,m))^-0.5;
    end

    A=[W B;B' zeros(20,20)];
    b=[W*T_ast(i,:)' ;Qm(i,:)' ]';

    x=A\b;
    T(i,:)=x(1:23);

end

figure(13);clf;
subplot(2,1,1)

```

```

hold on;
plot(t,T_ast(:,1), 'r')
plot(t,T(:,1));
hold off
subplot(2,1,2)
hold on;
plot(t,T_ast(:,2), 'r')
plot(t,T(:,2));
hold off

% T(:, [16 20])=T(:, [16 20])/4;
% T(:, [17 21])=T(:, [17 21])/4;
% T(:, [18 22])=T(:, [18 22])/4;

% figure(9); clf;plot(T(:, [17 21]));xlabel('time (s)');ylabel('Momento
(N-m)');grid on;lighting gouraud
% set(gcf,'Renderer','OpenGL')
% legend('Rodilla Derecha','Rodilla Izquierda')

function [alfa,rg_g]=grafica_CG(qc,cg_l,l)
%grafica los CG, y devuelve el coeficiente de proximidad

alfa=zeros(1,18);
rg_g=zeros(length(qc),36);
ind = @(n) 2*n-1:2*n; pointer = @(j,m) qc(j,ind(m));
order = [1 2 1 3 3 4 4 5 3 6 6 7 7 8 3 9 9 10 10 11 2 12 12 13 13 14 14
15 2 16 16 17 17 18 18 19];

    for j=1:length(qc)
    for i=1:length(alfa)
        if i==4||i==7||i==10||i==14||i==18
            alfa(1,i)=1;%CG
        else
            alfa(1,i) = norm(cg_l(i,:))/l(i);
        end

        rg_g(j,ind(i))=pointer(j,order(2*i-
1))+alfa(i)*(pointer(j,order(2*i))-pointer(j,order(2*i-1)));
    end

    end

figure(8); clf;

order1 = [5 4 3 6 7 8 7 6 3 1 2 12 13 14 15];
order2 = [3 9 10 11];
order3 = [2 16 17 18 19];

P1 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b. '); hold on
P2 = plot(qc(1,1:2:end), qc(1,2:2:end), 'bo ');
P3 = plot(qc(1,1:2:end), qc(1,2:2:end), 'bo ');

v1 = plot(qc(1,1:2:end),qc(1,2:2:end), 'b- ');
v2 = plot(qc(1,1:2:end),qc(1,2:2:end), 'b-- ');
v3 = plot(qc(1,1:2:end),qc(1,2:2:end), 'b-- ');

P1P = plot(rg_g(1,1:2:end),rg_g(1,2:2:end), 'ro ');

```

```

hold off
title('Plano Sagital')
grid on
axis([-1.5 2 -0.5 2.5]); set(gca, 'DataAspectRatio', [1 1 1])
set(gcf, 'Renderer', 'OpenGL')
% legend('derecha', 'izquierda')

    x = qc(1,1:2:end);
    y = qc(1,2:2:end);

    set(P1, 'xdata', x(order1), 'ydata', y(order1))
    set(P2, 'xdata', x(order2), 'ydata', y(order2))
    set(P3, 'xdata', x(order3), 'ydata', y(order3))
    set(v1, 'xdata', x(order1), 'ydata', y(order1))
    set(v2, 'xdata', x(order2), 'ydata', y(order2))
    set(v3, 'xdata', x(order3), 'ydata', y(order3))

end

function grafica_fuerza_placas(qc, fp11, fp12)

figure(5); clf;
k=5e-4;

order1 = [5 4 3 6 7 8 7 6 3 1 2 12 13 14 15];
order2 = [3 9 10 11];
order3 = [2 16 17 18 19];

P1 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b. '); hold on
P2 = plot(qc(1,1:2:end), qc(1,2:2:end), 'bo');
P3 = plot(qc(1,1:2:end), qc(1,2:2:end), 'bo');

v1 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b-');
v2 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b--');
v3 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b--');

P1D = plot(fp11(1,7)+k*fp11(1,1), k*fp11(1,3), 'k^');
P2D = plot(fp11(1,7), 0, 'k. ');
v1D = plot([fp11(1,7) fp11(1,7)+k*fp11(1,1)], [0 k*fp11(1,3)], 'k-');

P1I = plot(fp12(1,7)+k*fp12(1,1), k*fp12(1,3), 'k^');
P2I = plot(fp12(1,7), 0, 'k. ');
v1I = plot([fp12(1,7) fp12(1,7)+k*fp12(1,1)], [0 k*fp12(1,3)], 'k-');

hold off
title('Plano Sagital')
grid on

```

```

axis([-1.5 2 -0.5 2.5]); set(gca, 'DataAspectRatio', [1 1 1])
xlabel('Axis X (m)');ylabel('Axis Z (m)');
set(gcf, 'Renderer', 'OpenGL')

for i = 1:length(qc)
    x = qc(i,1:2:end);
    y = qc(i,2:2:end);
    xpd = fp11(i,7)+k*fp11(i,1);
    ypd = k*fp11(i,3);
    xppd=fp11(i,7);
    yppd=0;
    xpi = fp12(i,7)+k*fp12(i,1);
    ypi = k*fp12(i,3);
    xppi=fp12(i,7);
    yppi=0;

    set(P1, 'xdata', x(order1), 'ydata', y(order1))
    set(P2, 'xdata', x(order2), 'ydata', y(order2))
    set(P3, 'xdata', x(order3), 'ydata', y(order3))
    set(v1, 'xdata',x(order1), 'ydata',y(order1))
    set(v2, 'xdata',x(order2), 'ydata',y(order2))
    set(v3, 'xdata',x(order3), 'ydata',y(order3))

    if fp11(i,3)>100

        set(P1D, 'xdata', xpd, 'ydata', ypd)
        set(P2D, 'xdata', xppd, 'ydata', yppd)
        set(v1D, 'xdata', [fp11(i,7) fp11(i,7)+k*fp11(i,1)], 'ydata', [0
k*fp11(i,3)]);
        end

        if fp12(i,3)>100
            set(P1I, 'xdata', xpi, 'ydata', ypi)
            set(P2I, 'xdata', xppi, 'ydata', yppi)
            set(v1I, 'xdata', [fp12(i,7) fp12(i,7)+k*fp12(i,1)], 'ydata', [0
k*fp12(i,3)]);
            end

        drawnow
        pause(0.001)
    end
end

function grafica_trayectorias(r,q_proj,qc)
%Funcion que grafica las trayectorias de los markerts, la proyeccion y la
%qcomplete

figure(2); clf
for i=1:19

```

```

plot(r(:,2*i-1),r(:,2*i),'b',q_proj(:,2*i-1),q_proj(:,2*i),'r',qc(:,2*i-1),qc(:,2*i),'g');hold on
end
pointer=[1 2 11 12 13 14 15 16 17 18 19];
for i=1:11
plot(r(:,2*i-1),r(:,2*i),'b',q_proj(:,2*i-1),q_proj(:,2*i),'r',qc(:,2*i-1),qc(:,2*i),'g');hold on
plot(qc(:,2*pointer(i-1)),qc(:,2*i),'b');hold on
end
grid on
lighting gouraud
set(gcf,'Renderer','OpenGL')
legend('q*','q proyect','q complete')

end

function I = Inercia(MI,cg_l,m)
%calcula el momento de inercia con respecto al primer punto del segmento,
%osea stainer

MI(:, [1 3 4])=[];
I = zeros(18,1);

%Momento de inercia con respecto a CG de cada segmento
I(1,:) = MI(1,:);
I(2,:) = MI(2,:);
I(3,:) = MI(3,:);
I(4,:) = MI(4,:);
I(5,:) = MI(5,:);
I(6,:) = MI(6,:);
I(7,:) = MI(7,:);
I(8,:) = MI(8,:);
I(9,:) = MI(9,:);
I(10,:) = MI(10,:);
I(11,:) = MI(11,:);
I(12,:) = MI(12,:);
I(13,:) = MI(13,:);
I(14,:) = MI(17,:);
I(15,:) = MI(14,:);
I(16,:) = MI(15,:);
I(17,:) = MI(16,:);
I(18,:) = MI(18,:);

%Aplicacion de Stainer
for i = 1:length(I)
    I(i,:)=I(i,:)+m(i)*norm(cg_l(i,:))^2;
end

end

function [l] = longitudes(r)
%funcion que calcula longitudes promedio de los segmentos

l = zeros(1,18);
ind = @(i) 2*i-1:2*i;

```

```

l(1) = promedio(r(:,ind(1)),r(:,ind(2)));
l(2) = promedio(r(:,ind(1)),r(:,ind(3)));
l(3) = promedio(r(:,ind(3)),r(:,ind(4)));
l(4) = promedio(r(:,ind(4)),r(:,ind(5)));
l(5) = promedio(r(:,ind(3)),r(:,ind(6)));
l(6) = promedio(r(:,ind(6)),r(:,ind(7)));
l(7) = promedio(r(:,ind(7)),r(:,ind(8)));
l(8) = promedio(r(:,ind(9)),r(:,ind(3)));
l(9) = promedio(r(:,ind(10)),r(:,ind(9)));
l(10) = promedio(r(:,ind(11)),r(:,ind(10)));
l(11) = promedio(r(:,ind(12)),r(:,ind(2)));
l(12) = promedio(r(:,ind(13)),r(:,ind(12)));
l(13) = promedio(r(:,ind(14)),r(:,ind(13)));
l(14) = promedio(r(:,ind(15)),r(:,ind(14)));
l(15) = promedio(r(:,ind(16)),r(:,ind(2)));
l(16) = promedio(r(:,ind(17)),r(:,ind(16)));
l(17) = promedio(r(:,ind(18)),r(:,ind(17)));
l(18) = promedio(r(:,ind(19)),r(:,ind(18)));

```

```

function lm = promedio(a,b)
    m = zeros(length(a),1);

    for t = 1:length(a)

        m(t,1) = norm(a(t,:)-b(t,:));

    end
    lm=mean(m);

end

```

```
end
```

```

function markers(r,q_proj)

figure(1); clf;

order1 = [5 4 3 6 7 8 7 6 3 1 2 12 13 14 15];
order2 = [3 9 10 11];
order3 = [2 16 17 18 19];

P1 = plot(r(1,1:2:end), r(1,2:2:end), 'b. '); hold on
P2 = plot(r(1,1:2:end), r(1,2:2:end), 'bo ');
P3 = plot(r(1,1:2:end), r(1,2:2:end), 'bo ');

v1 = plot(r(1,1:2:end),r(1,2:2:end), 'b- ');
v2 = plot(r(1,1:2:end),r(1,2:2:end), 'b-- ');
v3 = plot(r(1,1:2:end),r(1,2:2:end), 'b-- ');

P1P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'r. ');
P2P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'ro ');

```

```

P3P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'ro');
v1P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'r-');
v2P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'r--');
v3P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'r--');

P1P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'b.');
```

hold on

```

P2P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'bo');
P3P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'bo');
v1P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'b-');
v2P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'b--');
v3P = plot(q_proj(1,1:2:end),q_proj(1,2:2:end), 'b--');
```



```

title('Plano Sagital')
axis([-1.5 2 -0.5 2]); set(gca, 'DataAspectRatio', [1 1 1])
set(gcf, 'Renderer', 'OpenGL')
xlabel('Axis X (m)');ylabel('Axis Z (m)');
grid on
```



```

for i = 1:length(r)
    x = r(i,1:2:end);
    y = r(i,2:2:end);
    xp = q_proj(i,1:2:end);
    yp = q_proj(i,2:2:end);
    set(P1, 'xdata', x(order1), 'ydata', y(order1))
    set(P2, 'xdata', x(order2), 'ydata', y(order2))
    set(P3, 'xdata', x(order3), 'ydata', y(order3))
    set(v1, 'xdata', x(order1), 'ydata', y(order1))
    set(v2, 'xdata', x(order2), 'ydata', y(order2))
    set(v3, 'xdata', x(order3), 'ydata', y(order3))

    set(P1P, 'xdata', xp(order1), 'ydata', yp(order1))
    set(P2P, 'xdata', xp(order2), 'ydata', yp(order2))
    set(P3P, 'xdata', xp(order3), 'ydata', yp(order3))
    set(v1P, 'xdata', xp(order1), 'ydata', yp(order1))
    set(v2P, 'xdata', xp(order2), 'ydata', yp(order2))
    set(v3P, 'xdata', xp(order3), 'ydata', yp(order3))

    drawnow
    pause(0.001)
    % P=num2str(i);permite guardar imagenes
    % saveas(figure(1),P, 'png');
    %

end
```



```

function [m] = masas(MI)
%funcion que permite calcular las masas de los segmentos corporales
m = zeros(1,18);

%cadera
m(1)=MI(1,4);
```

```
%tronco
m(2)=MI(2,4);

%cuello
m(3)=MI(3,4);

%cabeza
m(4)=MI(4,4);

%Brazo derecho
m(5)=MI(5,4);

%Antebrazo derecho
m(6)=MI(6,4);

%Mano derecha
m(7)=MI(7,4);

%Brazo izquierdo
m(8)=MI(8,4);

%Antebrazo izquierdo
m(9)=MI(9,4);

%mano izquierda
m(10)=MI(10,4);

%pierna derecha
m(11)=MI(11,4);

%pantorrilla derecha
m(12)=MI(12,4);

%Pie derecho
m(13)=MI(13,4);

%punta pie derecho
m(14)=MI(17,4);

%pierna izquierda
m(15)=MI(14,4);

%pantorrilla izquierda
m(16)=MI(15,4);

%pie izquierdo
m(17)=MI(16,4);

%punta pie izquierdo
m(18)=MI(18,4);
end

function [Qc] = matriz_de_fuerzas(m,l,alfa)
%Funcion que calcula la matriz de fuerzas, solo el peso
%gravedad
```

```

g=9.81;
Q=zeros(38,1);
order = [1 2 1 3 3 4 4 5 3 6 6 7 7 8 3 9 9 10 10 11 2 12 12 13 13 14 14
15 2 16 16 17 17 18 18 19];
ind = @(n) 2*n-1:2*n; pointer = @(m) order(ind(m));

for i=1:length(l)

b=pointer(i);
x=b(1); y=b(2);
Q([ind(x),ind(y)])=Q([ind(x),ind(y)])-m(i)*g*[0 1-alfa(i) 0 alfa(i)]';

end

Qc=zeros(56,1);

%Peso de segmentos
Qc(1:38,1)=Q(:,1);

```

```
end
```

```

function [M] = Matriz_de_masa(ma,cg_l,I,l)
%Matriz de masas generalizada
order = [1 2 1 3 3 4 4 5 3 6 6 7 7 8 3 9 9 10 10 11 2 12 12 13 13 14 14
15 2 16 16 17 17 18 18 19];
M = zeros(56);
ind = @(n) 2*n-1:2*n; pointer = @(m) order(ind(m));

```

```

for i = 1:length(l)
x = mass(ma(i),cg_l(i,:),I(i),l(i));
b=pointer(i);
m=b(1);n=b(2);
M(ind(m),ind(m))=M(ind(m),ind(m))+x(ind(1),ind(1));
M(ind(n),ind(m))=M(ind(n),ind(m))+x(ind(1),ind(2));
M(ind(m),ind(n))=M(ind(m),ind(n))+x(ind(2),ind(1));
M(ind(n),ind(n))=M(ind(n),ind(n))+x(ind(2),ind(2));

```

```
end
```

```

function a = mass(m1,cg_l1,I1,l1)
a = [m1-(2*m1*cg_l1(1)/l1)+I1/(l1^2),0,(m1*cg_l1(1)/l1)-I1/(l1^2),
-m1*cg_l1(2)/(l1); 0,m1-(2*m1*cg_l1(1)/l1)+I1/(l1^2),m1*cg_l1(2)/(l1),
(m1*cg_l1(1)/l1)-I1/(l1^2); (m1*cg_l1(1)/l1)-I1/(l1^2),m1*cg_l1(2)/(l1),
I1/(l1^2),0; -m1*cg_l1(2)/(l1),(m1*cg_l1(1)/l1)-I1/(l1^2),0,I1/(l1^2)];
end

```

```
end
```

```
end
```

```

function [R] = Matriz_R(qc,l)
%Calculo de la matriz R

```

```
R=zeros(56,20);
```

```
inc=3:38;
```

```

[Phic,Phiqc] = restric_c(qc,1);
R(inc,:)=Phiqc(:,inc)\(-Phiqc(:, [1:2,39:56]));
R([1:2,39:56],:)=eye(20);

```

```
end
```

```

function [Qd] = metodo_directo(qc,M,qcs,Qc,t,1)
%Función que calcula Qd={T lambda}' con el paso del tiempo t

```

```

%Matriz B
B = zeros(11,5);
%Referencia a variables independientes x1,y1,theta1,theta2,theta3
B(1:2,1:2) = eye(2);
B(9:11,3:5) = eye(3);

```

```
Qd=zeros(length(t),11);
```

```

for i = 1:length(t)
[Phic,Phiqc] = restric_c(qc(i,:),1);
A=[B, -Phiqc'];
A=A\(M*qcs(i,.)'-Qc(i,.)');

```

```
Qd(i,:)=B*A(1:5);
```

```
end
```

```
end
```

```

function [r] = modificacion_r0(r0,AM,r1)
%Función que calcula los r0 para CDG de la cabeza, manos y pies, y las
%adiciona a la matriz de r0.

```

```

A=@(t,i) reshape(AM(t,9*i-8:9*i),[3 3]);%función anónima que forma
matrices
%de rotacion de 3x3

```

```

r1=zeros(length(r0),15);
ind = @(i) 3*i-1:3*i+1;%indice que apunta en la matriz r0 para coger las
%coordenadas

```

```

%columna 1: Cabeza, Columna 2: Mano derecha, Columna 3: Mano izquierda,
%Columna 4: Pie derecho, Columna 5: Pie izquierdo.

```

```

r0p=[7 9 11 16 17];
AMp=[4 7 10 17 18];
rlp=[8 11 14 21 22];

```

```
sub= @(n) 3*n-2:3*n;
```

```
%Formación de la matriz r de los puntos de la cabeza, manos y pies
```

```

for n = 1:length(r0p)
for t = 1:length(r0)
r1(t,sub(n))=(r0(t,ind(r0p(n))))'+A(t,AMp(n))*(r1(rlp(n),:))');
end
end

```

```
end
```

```

%%Formación de la matriz r general, union de r0 y r1

r = zeros(length(r0),57);%cambiar de 57 a 66

%%Comentar lo siguiente para activar grafica en 3D%%
%%

r(:,sub(1))=r0(:,ind(1));
for i=1:length(r0)
r(i,sub(2))=mean([r0(i,ind(2));r0(i,ind(3))]);
r(i,sub(3))=mean([r0(i,ind(4));r0(i,ind(5));r0(i,ind(6))]);
end
r(:,sub(4))=r0(:,ind(7));
r(:,sub(5))=r1(:,sub(1));
r(:,sub(6))=r0(:,ind(8));
r(:,sub(7))=r0(:,ind(9));
r(:,sub(8))=r1(:,sub(2));
r(:,sub(9))=r0(:,ind(10));
r(:,sub(10))=r0(:,ind(11));
r(:,sub(11))=r1(:,sub(3));
r(:,sub(12))=r0(:,ind(12));
r(:,sub(13))=r0(:,ind(13));
r(:,sub(14))=r0(:,ind(16));
r(:,sub(15))=r1(:,sub(4));
r(:,sub(16))=r0(:,ind(14));
r(:,sub(17))=r0(:,ind(15));
r(:,sub(18))=r0(:,ind(17));
r(:,sub(19))=r1(:,sub(5));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %%%%%%%%%%-----Grafica en 3D%-----
% %%%%%%%%%%

% r(:,sub(1))=r0(:,ind(1));
% r(:,sub(2))=r0(:,ind(2));
% r(:,sub(3))=r0(:,ind(3));
% r(:,sub(4))=r0(:,ind(4));
% r(:,sub(5))=r0(:,ind(5));
% r(:,sub(6))=r0(:,ind(6));
% r(:,sub(7))=r0(:,ind(7));
% r(:,sub(8))=r1(:,sub(1));
% r(:,sub(9))=r0(:,ind(8));
% r(:,sub(10))=r0(:,ind(9));
% r(:,sub(11))=r1(:,sub(2));
% r(:,sub(12))=r0(:,ind(10));
% r(:,sub(13))=r0(:,ind(11));
% r(:,sub(14))=r1(:,sub(3));
% r(:,sub(15))=r0(:,ind(12));
% r(:,sub(16))=r0(:,ind(13));
% r(:,sub(17))=r0(:,ind(16));
% r(:,sub(18))=r0(:,ind(14));
% r(:,sub(19))=r0(:,ind(15));
% r(:,sub(20))=r0(:,ind(17));

```

```

% r(:,sub(21))=r1(:,sub(4));
% r(:,sub(22))=r1(:,sub(5));
% figure(4);
% clf;
%
%
% order1 = [8 7 4 1 5 4 5 9 10 11 10 9 5 4 1 2 3 2 15 16 17 21];
% order2 = [4 1 6 4 6 12 13 14];
% order3 = [1 3 18 19 20 22];
%
%
% P1 = plot3(r(1,1:3:end), r(1,2:3:end), r(1,3:3:end),'b.');
```

hold on

```

% P2 = plot3(r(1,1:3:end), r(1,2:3:end), r(1,3:3:end),'bo');
```

P3 = plot3(r(1,1:3:end), r(1,2:3:end),r(1,3:3:end), 'bo');

```

%
% v1 = plot3(r(1,1:3:end),r(1,2:3:end),r(1,3:3:end),'b-');
```

v2 = plot3(r(1,1:3:end),r(1,2:3:end),r(1,3:3:end),'b--');

```

% v3 = plot3(r(1,1:3:end),r(1,2:3:end),r(1,3:3:end),'b--');
```

hold off

```

% axis([-1.5 2 -0.5 2.5 -0.5 2.5]); set(gca, 'DataAspectRatio', [1 1
1])
% set(gcf,'Renderer','OpenGL')
% xlabel('Axis X (m)');ylabel('Axis Z (m)');zlabel('Axis Y (m)');
% grid on
%
%
% for i = 1:length(r)
%     x = r(i,1:3:end);
%     y = r(i,2:3:end);
%     z = r(i,3:3:end);
%
%     set(P1, 'xdata', x(order1), 'ydata', y(order1), 'zdata',
z(order1))
%     set(P2, 'xdata', x(order2), 'ydata', y(order2), 'zdata',
z(order2))
%     set(P3, 'xdata', x(order3), 'ydata', y(order3), 'zdata',
z(order3))
%     set(v1,'xdata',x(order1),'ydata',y(order1), 'zdata', z(order1))
%     set(v2,'xdata',x(order2),'ydata',y(order2), 'zdata', z(order2))
%     set(v3,'xdata',x(order3),'ydata',y(order3), 'zdata', z(order3))
%
%
%
%     drawnow
%     pause(0.001)
% end
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%-----
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r(:,2:3:end)=[];%se elimina la columna Y, 3D a 2D

%longitudes pierna derecha

%longitud muslo
% long_muslo=zeros(length(r),1);
% for i=1:length(r)
%
%     long_muslo(i,1)= norm(r(i,3:4)-r(i,23:24));
%
%
```

```

% end

end

function [r] = modificacion_rl(r0,AM,rl)
A=@(t,i) reshape(AM(t,9*i-8:9*i),[3 3]);%función anónima que forma
matrices de rotacion de 3x3

r=zeros(length(r0),15);
ind = @(i) 3*i-1:3*i+1;

for t = 1:length(r0)
    r(t,1:3)=(r0(t,ind(7))'+A(t,4)*rl(8,:))';%CDG de la cabeza
    r(t,4:6)=(r0(t,ind(9))'+A(t,7)*rl(11,:))';%CDG mano derecha
    r(t,7:9)=(r0(t,ind(11))'+A(t,10)*rl(14,:))';%CDG mano izquierda
    r(t,10:12)=(r0(t,ind(16))'+A(t,17)*rl(21,:))';%CDG pie derecho
    r(t,13:15)=(r0(t,ind(17))'+A(t,18)*rl(22,:))';%CDG pie izquierdo

end

end

function [qc]=qcomplete(q_proj,l,t)

%Ángulos iniciales
theta = zeros(1,18);
theta(1)=pi/2; theta(2)=pi/2; theta(3)=pi/2; theta(4)=pi/4;
theta(5)=pi/6;
theta(6)=pi/2; theta(7)=pi/2; theta(8)=pi/6; theta(9)=pi/2;
theta(10)=pi/2;
theta(11)=pi/2; theta(12)=pi/3; theta(13)=pi/6; theta(14)=pi/6;
theta(15)=pi/2;
theta(16)=pi/3; theta(17)=pi/6; theta(18)=pi/6;

qc=zeros(length(q_proj),56);
grados=zeros(length(q_proj),18);
%=====
==
for i=1:length(q_proj)

    qc(i,:)=[q_proj(i,:),theta]';

    [Phic,Phiqc] = restric_c(qc(i,:),l);

    error=norm(Phic); tol=1e-10;

```

```

while error > tol

    inc = 39:56;
    A=Phiqc'*Phiqc;
    B=-Phiqc'*Phic;
    deltaq=A(inc, inc)\B(inc);
    qc(i,inc)=qc(i,inc)+deltaq';

    [Phic, Phiqc]=restric_c(qc(i,:),l);

    error=norm(Phic);

end

%   grados(i,:)=qc(i,39:56)*180/pi;%si se quiere los angulos en grados
%centesimales

=====
==
end

end

function [q_proj]= qproj(r,l)

% q proyectada, los puntos que no cambian mucho son los tobillos
% Constantes
q_proj = zeros(length(r), 38);

for i=1:length(r)
    tol=1e-12; alpha=1e9;

    % Estimación de la posición inicial
    q=r(i, :)';
    q0=q;

    % Proyección de la posición
    %peso en los puntos que no varían mucho
    a=ones(1,38);
    ind = @(n) 2*n-1:2*n;
    a(ind(13))=60;
    a(ind(17))=60;

    W=diag(a); lambda=zeros(18,1); ite=0;

    [Phi, Phiq]=restric(q,l);

    error=norm(Phi);

    while error > tol
        if ite>0
            lambda=lambda+alpha*Phi;

```

```

end
Tan=W+Phiq'*alpha*Phiq;
res=W*(q-q0)+Phiq'*(alpha*Phi+lambda);
deltaq=-Tan\res; q=q+deltaq;

[Phi, Phiq]=restric(q,l);

error=norm(Phi);

ite=ite+1;
end
q_proj(i,:)=q';

end

% longitud muslo
% long_muslo_1=zeros(length(q_proj),1);
% for i=1:length(q_proj)
%
%     long_muslo_1(i,1)= norm(q_proj(i,3:4)-q_proj(i,23:24));
%
% end

end

function [Phi, Phiq]=restric(q1,l)

%Funcion que devuelve la matriz Phi y Phiq

ind = @(i) 2*i-1:2*i; r = @(i) q1(ind(i));

Phi = [(r(1)-r(2))'*(r(1)-r(2))-l(1)^2;
        (r(1)-r(3))'*(r(1)-r(3))-l(2)^2;
        (r(3)-r(4))'*(r(3)-r(4))-l(3)^2;
        (r(4)-r(5))'*(r(4)-r(5))-l(4)^2;
        (r(3)-r(6))'*(r(3)-r(6))-l(5)^2;
        (r(6)-r(7))'*(r(6)-r(7))-l(6)^2;
        (r(7)-r(8))'*(r(7)-r(8))-l(7)^2;
        (r(3)-r(9))'*(r(3)-r(9))-l(8)^2;
        (r(9)-r(10))'*(r(9)-r(10))-l(9)^2;
        (r(10)-r(11))'*(r(10)-r(11))-l(10)^2;
        (r(2)-r(12))'*(r(2)-r(12))-l(11)^2;
        (r(12)-r(13))'*(r(12)-r(13))-l(12)^2;
        (r(13)-r(14))'*(r(13)-r(14))-l(13)^2;
        (r(14)-r(15))'*(r(14)-r(15))-l(14)^2;
        (r(2)-r(16))'*(r(2)-r(16))-l(15)^2;
        (r(16)-r(17))'*(r(16)-r(17))-l(16)^2;
        (r(17)-r(18))'*(r(17)-r(18))-l(17)^2;
        (r(18)-r(19))'*(r(18)-r(19))-l(18)^2];

if nargin == 2
    Phiq = zeros(length(l),length(q1));
    Phiq(1,[ind(1) ind(2)]) = [2*(r(1)-r(2))' -2*(r(1)-r(2))'];
    Phiq(2,[ind(1) ind(3)]) = [2*(r(1)-r(3))' -2*(r(1)-r(3))'];
    Phiq(3,[ind(3) ind(4)]) = [2*(r(3)-r(4))' -2*(r(3)-r(4))'];

```

```

    Phiq(4,[ind(4) ind(5)]) = [2*(r(4)-r(5))' -2*(r(4)-r(5))'];
    Phiq(5,[ind(3) ind(6)]) = [2*(r(3)-r(6))' -2*(r(3)-r(6))'];
    Phiq(6,[ind(6) ind(7)]) = [2*(r(6)-r(7))' -2*(r(6)-r(7))'];
    Phiq(7,[ind(7) ind(8)]) = [2*(r(7)-r(8))' -2*(r(7)-r(8))'];
    Phiq(8,[ind(3) ind(9)]) = [2*(r(3)-r(9))' -2*(r(3)-r(9))'];
    Phiq(9,[ind(9) ind(10)]) = [2*(r(9)-r(10))' -2*(r(9)-
r(10))'];
    Phiq(10,[ind(10) ind(11)]) = [2*(r(10)-r(11))' -2*(r(10)-
r(11))'];
    Phiq(11,[ind(2) ind(12)]) = [2*(r(2)-r(12))' -2*(r(2)-
r(12))'];
    Phiq(12,[ind(12) ind(13)]) = [2*(r(12)-r(13))' -2*(r(12)-
r(13))'];
    Phiq(13,[ind(13) ind(14)]) = [2*(r(13)-r(14))' -2*(r(13)-
r(14))'];
    Phiq(14,[ind(14) ind(15)]) = [2*(r(14)-r(15))' -2*(r(14)-
r(15))'];
    Phiq(15,[ind(2) ind(16)]) = [2*(r(2)-r(16))' -2*(r(2)-
r(16))'];
    Phiq(16,[ind(16) ind(17)]) = [2*(r(16)-r(17))' -2*(r(16)-
r(17))'];
    Phiq(17,[ind(17) ind(18)]) = [2*(r(17)-r(18))' -2*(r(17)-
r(18))'];
    Phiq(18,[ind(18) ind(19)]) = [2*(r(18)-r(19))' -2*(r(18)-
r(19))'];

    end
end

```

```
function [Phic,Phiqc] = restric_c(q,l)
```

```

q0=q(1:38);
theta=q(39:56);
[Phi, Phiq] = restric(q0',l);

```

```

x = @(n) q(2*n-1);
y = @(m) q(2*m);

```

```

    Phic = [Phi;
    x(1)-x(2)+cos(theta(1))*1(1);
    x(1)-x(3)+cos(theta(2))*1(2);
    x(3)-x(4)+cos(theta(3))*1(3);
    y(4)-y(5)+sin(theta(4))*1(4);
    y(3)-y(6)+sin(theta(5))*1(5);
    x(6)-x(7)+cos(theta(6))*1(6);
    x(7)-x(8)+cos(theta(7))*1(7);
    y(3)-y(9)+sin(theta(8))*1(8);
    x(9)-x(10)+cos(theta(9))*1(9);
    x(10)-x(11)+cos(theta(10))*1(10);
    x(2)-x(12)+cos(theta(11))*1(11);
    x(12)-x(13)+cos(theta(12))*1(12);
    y(13)-y(14)+sin(theta(13))*1(13);%ecuacion redundante
    x(13)-x(14)+cos(theta(13))*1(13);%ecuacion redundante
    y(14)-y(15)+sin(theta(14))*1(14);%ecuacion redundante
    x(14)-x(15)+cos(theta(14))*1(14);%ecuacion redundante
    x(2)-x(16)+cos(theta(15))*1(15);
    x(16)-x(17)+cos(theta(16))*1(16);
    y(17)-y(18)+sin(theta(17))*1(17);%ecuacion redundante

```

```

x(17)-x(18)+cos(theta(17))*l(17);%ecuacion redundante
y(18)-y(19)+sin(theta(18))*l(18);%ecuacion redundante
x(18)-x(19)+cos(theta(18))*l(18);];%ecuacion redundante

```

```

if nargout == 2
    Phiqc = zeros(40,56);
    Phiqc(1:18,1:38) = Phiq;
    Phiqc(19,1:3) = [1 0 -1];
    Phiqc(19,39) = -l(1)*sin(theta(1));
    Phiqc(20,1:5) = [1 0 0 0 -1];
    Phiqc(20,40) = -l(2)*sin(theta(2));
    Phiqc(21,5:7) = [1 0 -1];
    Phiqc(21,41) = -l(3)*sin(theta(3));
    Phiqc(22,8:10) = [1 0 -1];
    Phiqc(22,42) = l(4)*cos(theta(4));
    Phiqc(23,6:12) = [1 0 0 0 0 0 -1];
    Phiqc(23,43) = l(5)*cos(theta(5));
    Phiqc(24,11:13) = [1 0 -1];
    Phiqc(24,44) = -l(6)*sin(theta(6));
    Phiqc(25,13:15) = [1 0 -1];
    Phiqc(25,45) = -l(7)*sin(theta(7));
    Phiqc(26,6:18) = [1 0 0 0 0 0 0 0 0 0 0 0 -1];
    Phiqc(26,46) = l(8)*cos(theta(8));
    Phiqc(27,17:19) = [1 0 -1];
    Phiqc(27,47) = -l(9)*sin(theta(9));
    Phiqc(28,19:21) = [1 0 -1];
    Phiqc(28,48) = -l(10)*sin(theta(10));
    Phiqc(29,3) = 1; Phiqc(29,23)=-1;
    Phiqc(29,49) = -l(11)*sin(theta(11));
    Phiqc(30,23:25) = [1 0 -1];
    Phiqc(30,50) = -l(12)*sin(theta(12));
    Phiqc(31,26:28) = [1 0 -1];
    Phiqc(31,51) = l(13)*cos(theta(13));
    Phiqc(32,25:27) = [1 0 -1];
    Phiqc(32,51) = -l(13)*sin(theta(13));
    Phiqc(33,28:30) = [1 0 -1];
    Phiqc(33,52) = l(14)*cos(theta(14));
    Phiqc(34,27:29) = [1 0 -1];
    Phiqc(34,52) = -l(14)*sin(theta(14));
    Phiqc(35,3) = 1; Phiqc(35,31)=-1;
    Phiqc(35,53) = -l(15)*sin(theta(15));
    Phiqc(36,31:33) = [1 0 -1];
    Phiqc(36,54) = -l(16)*sin(theta(16));
    Phiqc(37,34:36) = [1 0 -1];
    Phiqc(37,55) = l(17)*cos(theta(17));
    Phiqc(38,33:35) = [1 0 -1];
    Phiqc(38,55) = -l(17)*sin(theta(17));
    Phiqc(39,36:38) = [1 0 -1];
    Phiqc(39,56) = l(18)*cos(theta(18));
    Phiqc(40,35:37) = [1 0 -1];
    Phiqc(40,56) = -l(18)*sin(theta(18));

```

```
end
```

```
end
```

```
function [L] = segmentos(r1)
```

```

%Longitudes de los segmentos
L=zeros(18,1);

%segmento del tronco y cabeza
L(1)=norm(rl(1,:)-rl(2,:));
L(2)=norm(rl(1,:)-rl(3,:));
L(3)=norm(rl(3,:)-rl(4,:));
L(4)=norm(rl(4,:)-rl(5,:));

%segmentos de los brazos
%Derecho
L(5)=norm(rl(6,:)-rl(3,:));
L(6)=norm(rl(7,:)-rl(6,:));
L(7)=norm(rl(8,:)-rl(7,:));
%Izquierdo
L(8)=norm(rl(9,:)-rl(3,:));
L(9)=norm(rl(10,:)-rl(9,:));
L(10)=norm(rl(11,:)-rl(10,:));

%segmentos de las piernas
%Derecha
L(11)=norm(rl(12,:)-rl(2,:));
L(12)=norm(rl(13,:)-rl(12,:));
L(13)=norm(rl(14,:)-rl(13,:));
L(17)=norm(rl(18,:)-rl(14,:));
%Izquierda
L(14)=norm(rl(15,:)-rl(2,:));
L(15)=norm(rl(16,:)-rl(15,:));
L(16)=norm(rl(17,:)-rl(16,:));
L(18)=norm(rl(19,:)-rl(17,:));

```

End

```
function [y, r, vr] = ssa(x1, L)
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% -----
%   Author: Francisco Javier Alonso Sanchez   e-mail:fjas@unex.es
%   Departament of Electronics and Electromecanical Engineering
%   Industrial Engineering School
%   University of Extremadura
%   Badajoz
%   Spain
% -----
%
% SSA generates a trajectory matrix X from the original series x1
% by sliding a window of length L. The trajectory matrix is approximated
% by using Singular Value Decomposition. The last step reconstructs
% the series from the approximated trayectory matrix. The SSA
% applications
% include smoothing, filtering, and trend extraction.
% The algorithm used is described in detail in: Golyandina, N.,
% Nekrutkin,
% V., Zhigljavsky, A., 2001. Analisis of Time Series Structure - SSA and
% Related Techniques. Chapman & Hall/CR.

```

```

% x1 Original time series (column vector form)
% L Window length
% y Reconstructed time series
% r Residual time series r = x1 - y
% vr Relative value of the norm of the approximated trajectory matrix
with
%   respect to the original trajectory matrix

% The program output is the Singular Spectrum of x1 (must be a column
% vector), using a window length L. You must choose the components used
% to reconstruct the series in the form [i1,i2:ik,...,iL], based on the
% Singular Spectrum appearance.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

% Step 1: Build trajectory matrix

N = length(x1);
if L > N/2
    L = N - L;
end
K = N - L + 1;
X = zeros(L, K);
for i = 1:K
    X(1:L, i) = x1(i:L+i-1);
end

% Step 2: SVD

U = X * X';
[U, autoval] = eig(U);
[d, i] = sort(-diag(autoval));
U = U(:, i);
V = X' * U;

% Step 3: Grouping

% I = input(['Component agrupation to reconstruct the series ' ...
%   'in the form I = [i1, i2:ik, ..., iL] ']);
I = 1:2;
rca = U(:, I) * V(:, I)';

% Step 4: Reconstruction

y = zeros(N, 1);
Lp = min(L, K);
Kp = max(L, K);

for k = 0:Lp-2
    for m = 1:k+1
        y(k+1) = y(k+1) + rca(m, k-m+2) / (k+1);
    end
end

for k = Lp-1:Kp-1

```

```

    for m = 1:Lp
        y(k+1) = y(k+1) + rca(m,k-m+2) / Lp;
    end
end

for k = Kp:N
    for m = k-Kp+2:N-Kp+1;
        y(k+1) = y(k+1) + rca(m,k-m+2) / (N-k);
    end
end

if nargout > 1
    d = -d;
    sev = sum(d);
    r = x1 - y;
    vr = sum(d(I)) / sev * 100;

    figure(1)

    semilogy(100*d./sev); hold on; semilogy(100*d./sev, 'rx'); hold off
    title('Singular Spectrum')
    xlabel('Eigenvalue Number')
    ylabel('Eigenvalue (% norm of trajectory matrix retained)')

    figure(2)

    subplot(2, 1, 1)
    plot(x1); hold on; plot(y, 'r'); hold off
    xlabel('Data point')
    ylabel('Original and reconstructed series')

    subplot(2, 1, 2)
    plot(r, 'g')
    xlabel('Data point')
    ylabel('Residual series')
end

function [z]=zeta(qc)

[m,n]=size(qc);

z=zeros(m,20);

for i=1:m

    z(i,:)=[qc(i,1:2) qc(i,39:56)];

end

```

MODELO CON MULETAS Y ÓRTESIS

```

%Import captures data
AM=importdata('AM.dat');

```

```

r1=importdata('r1.dat');
r0=importdata('r0.dat');
MI=importdata('MI.dat');
rg=importdata('rg.dat');
fp11=importdata('FPL1.dat');
fp12=importdata('FPL2.dat');
t=r0(:,1);

[r] = modificacion_r0(r0,AM,r1);

[l] = longitudes(r);

[q_proj] = qproj(r,l);

for i=1:1
markers(r,q_proj);
end

[qc] = qcomplete(q_proj,l,t);
qc_antes=qc;

[z] = zeta(qc);% vector de 20x1

%Filtrado de z
z_fil = z;
for i = 1:20
    for j = 1:3
        z_fil(:,i) = ssa(z_fil(:,i),9);
    end
end

%construccion de la nueva qc con valores filtrados
qc(:,1:2) = z_fil(:,1:2);
qc(:,39:56) = z_fil(:,3:20);

%consistencia cinemática con los valores filtrados de z de los valores
%independientes
for i = 1:length(t)
    [Phic,Phiqc] = restric_c(qc(i,:),l);
    error = norm(Phic); tol = 1e-14;
    while error > tol

        inc = 3:38;
        A = Phiqc'*Phiqc;
        B = -Phiqc'*Phic;
        deltaq = A(inc, inc)\B(inc);
        qc(i,inc) = qc(i,inc)+deltaq';

        [Phic, Phiqc] = restric_c(qc(i,:),l);

        error = norm(Phic);

    end

end

%Primera Derivada de z

```

```

z_fil_p = zeros(length(t),20);

%Segunda Derivada de z

z_fil_s = zeros(length(t),20);

for i = 1:20

    z_fil_p(:,i) = derivar(t,z_fil(:,i));
    z_fil_s(:,i) = derivar(t,z_fil_p(:,i));

end

%Calculo de qcp a partir de z_fil_p
qcp = zeros(length(t),56);
qcp(:,1:2) = z_fil_p(:,1:2);
qcp(:,39:56) = z_fil_p(:,3:20);

%Calculo de qcs a partir de z_fil_s
qcs = zeros(length(t),56);
qcs(:,1:2) = z_fil_s(:,1:2);
qcs(:,39:56) = z_fil_s(:,3:20);

%Cálculo phiqqqp
order = [1 2 1 3 3 4 4 5 3 6 6 7 7 8 3 9 9 10 10 11 2 12 12 13 13 14 14
15 2 16 16 17 17 18 18 19];
phiqqqp = zeros(36,1);
for i = 1:length(t)
    [Phic,Phiqc] = restric_c(qc(i,:),1);

    inc = 3:38;
    A = Phiqc;
    B = -Phiqc(:, [1:2,39:56])*z_fil_p(i,:)' ;
    C = A(:,inc)\B;
    qcp(i,inc)=C';

    p = @(k) 2*k-1:2*k; m = @(c,g) qcp(c,p(order(g)));

    %armado del vector phiqqqp(1:18,1)
    a=0;
    for j=1:2:36
        a=a+1;
        f = m(i,j)-m(i,j+1);
        phiqqqp(a,:) = 2*(f*f');
    end

    %armado del vector phiqqqp(19:36,1)

    phiqqqp(19,:) = -1(1)*qcp(i,39)^2*cos(qcp(i,39));
    phiqqqp(20,:) = -1(2)*qcp(i,40)^2*cos(qcp(i,40));
    phiqqqp(21,:) = -1(3)*qcp(i,41)^2*cos(qcp(i,41));
    phiqqqp(22,:) = 1(4)*qcp(i,42)^2*sin(qcp(i,42));
    phiqqqp(23,:) = 1(5)*qcp(i,43)^2*sin(qcp(i,43));
    phiqqqp(24,:) = -1(6)*qcp(i,44)^2*cos(qcp(i,44));

```

```

    phiqpqp(25,:) = -l(7)*qcp(i,45)^2*cos(qcp(i,45));
    phiqpqp(26,:) = l(8)*qcp(i,46)^2*sin(qcp(i,46));
    phiqpqp(27,:) = -l(9)*qcp(i,47)^2*cos(qcp(i,47));
    phiqpqp(28,:) = -l(10)*qcp(i,48)^2*cos(qcp(i,48));
    phiqpqp(29,:) = -l(11)*qcp(i,49)^2*cos(qcp(i,49));
    phiqpqp(30,:) = -l(12)*qcp(i,50)^2*cos(qcp(i,50));
    phiqpqp(31,:) = l(13)*qcp(i,51)^2*sin(qcp(i,51));
    phiqpqp(32,:) = -l(13)*qcp(i,51)^2*cos(qcp(i,51));
    phiqpqp(33,:) = l(14)*qcp(i,52)^2*sin(qcp(i,52));
    phiqpqp(34,:) = -l(14)*qcp(i,52)^2*cos(qcp(i,52));
    phiqpqp(35,:) = -l(15)*qcp(i,53)^2*cos(qcp(i,53));
    phiqpqp(36,:) = -l(16)*qcp(i,54)^2*cos(qcp(i,54));
    phiqpqp(37,:) = l(17)*qcp(i,55)^2*sin(qcp(i,55));
    phiqpqp(38,:) = -l(17)*qcp(i,55)^2*cos(qcp(i,55));
    phiqpqp(39,:) = l(18)*qcp(i,56)^2*sin(qcp(i,56));
    phiqpqp(40,:) = -l(18)*qcp(i,56)^2*cos(qcp(i,56));

    B = -phiqpqp-Phiqc(:, [1:2,39:56])*z_fil_s(i,:)' ;
    C = A(:,inc)\B;
    qcs(i,inc)=C';
end

%Modulo velocidad
modulo_velocidad=zeros(length(t),19);
for i=1:length(t)

    for j=1:19
        modulo_velocidad(i,j)=sqrt((qcp(i,2*j-1))^2+(qcp(i,2*j))^2);
    end

end

figure(4);clf
plot(t,modulo_velocidad(:,1))

figure(7);clf
hold on
plot(t,modulo_velocidad(:,12))
xlabel('Tiempo(s)');
ylabel('Velocidad(m/s)');
title('Velocidad Rodilla Derecha');

plot(t,modulo_velocidad(:,16),'r')
hold off

% End velocidad

%Modulo aceleracion
modulo_aceleracion=zeros(length(t),19);
for i=1:length(t)

    for j=1:19
        modulo_aceleracion(i,j)=sqrt((qcs(i,2*j-1))^2+(qcs(i,2*j))^2);
    end

end

end

```

```

figure(10);clf
plot(t,modulo_aceleracion(:,1))

figure(11);clf
hold on
plot(t,modulo_aceleracion(:,12))
xlabel('Tiempo(s)');
ylabel('Aceleración(m/s^2)');
title('Aceleración Rodilla');

plot(t,modulo_aceleracion(:,16),'r')
hold off

%End Aceleracion

grafica_trayectorias(r,q_proj,qc);

figure(3);clf
subplot(7,2,1)
plot(t,qc(:,23),'b',t,qcp(:,23),'r',t,qcs(:,23),'g');
grid on
title('x12 vs t')
lighting gouraud
set(gcf,'Renderer','OpenGL')

subplot(7,2,2)
plot(t,qc(:,24),'b',t,qcp(:,24),'r',t,qcs(:,24),'g');
grid on
title('y12 vs t')
lighting gouraud

subplot(7,2,3)
plot(t,qc(:,31),'b',t,qcp(:,31),'r',t,qcs(:,31),'g');
grid on
title('x16 vs t')
lighting gouraud

subplot(7,2,4)
plot(t,qc(:,32),'b',t,qcp(:,32),'r',t,qcs(:,32),'g');
grid on
title('y16 vs t')
lighting gouraud

subplot(7,2,5)
plot(t,qc(:,25),'b',t,qcp(:,25),'r',t,qcs(:,25),'g');
grid on
title('x13 vs t')
lighting gouraud

subplot(7,2,6)
plot(t,qc(:,26),'b',t,qcp(:,26),'r',t,qcs(:,26),'g');
grid on
title('y13 vs t')
lighting gouraud

subplot(7,2,7)
plot(t,qc(:,33),'b',t,qcp(:,33),'r',t,qcs(:,33),'g');
grid on

```

```

title('x17 vs t')
lighting gouraud

subplot(7,2,8)
plot(t,qc(:,34),'b',t,qcp(:,34),'r',t,qcs(:,34),'g');
grid on
title('y17 vs t')
lighting gouraud

subplot(7,2,9:10)
plot(t,qc(:,50),'b',t,qcp(:,50),'r',t,qcs(:,50),'g');
grid on
title('theta 1')
lighting gouraud

subplot(7,2,11:12)
plot(t,qc(:,51),'b',t,qcp(:,51),'r',t,qcs(:,51),'g');
grid on
title('theta 2')
lighting gouraud

subplot(7,2,13:14)
plot(t,qc(:,52),'b',t,qcp(:,52),'r',t,qcs(:,52),'g');
grid on
title('theta 3')
lighting gouraud
legend('posicion','velocidad','aceleracion')

%Cálculo de matriz de masa
%masas de segmentos
m=masas(MI);
%CG locales de segmentos
cg_l=CG(rg);
[alfa,rg_g]=grafica_CG(qc,cg_l,l);

%Primera Derivada de rg_g

rg_gp = zeros(length(t),36);

for i = 1:36

    rg_gp(:,i) = derivar(t,rg_g);

end

%Momento de inercia con respecto a un extremo
I=Inercia(MI,cg_l,m);

[M]=Matriz_de_masa(m,cg_l,I,l);

%Matriz de fuerzas
[Qc]=matriz_de_fuerzas(m,l,alfa);

%Dinámica Inversa
tic
[T]= Dinamica_Inversa(t,qc,qcs,z_fil_p,fp11,fp12,l,M,Qc);

```

```
toc
```

```
T(:,16)=ssa(T(:,16),3);
T(:,20)=ssa(T(:,20),3);
T(:,17)=ssa(T(:,17),3);
T(:,21)=ssa(T(:,21),3);
T(:,18)=ssa(T(:,18),3);
T(:,22)=ssa(T(:,22),3);
```

```
figure(12);
subplot(3,1,1)
plot(t,T(:, [16 20])/2);ylabel('Caderas (N.m)');ylim([-80 80]);xlim([0
1.56]);
subplot(3,1,2)
plot(t,T(:, [17 21])/2);ylabel('Rodillas (N.m)');ylim([-80 80]);xlim([0
1.56]);
subplot(3,1,3)
plot(t,T(:, [18 22])/2);xlabel('Tiempo (s)');ylabel('Tobillos
(N.m)');ylim([-80 80]);xlim([0 1.56]);
```

```
for i=1:3
grafica_fuerza_placas(r,fp11,fp12);
end
```

```
%Energía
```

```
ener_cine=zeros(length(t),1); ener_pot_grav=zeros(length(t),1);g=9.81;
ener_tot=zeros(length(t),1);
for i=1:length(t)
ener_cine(i,1)=0.5*qcp(i,:)*M*qcp(i,:);
for j=1:length(m)
ener_pot_grav(i,1)=ener_pot_grav(i,1)+m(j)*g*(rg_g(i,2*j));
end
ener_tot(i,1)=ener_cine(i,1)+ener_pot_grav(i,1);
end
```

```
figure(6);clf
subplot(3,1,1)
plot(t,ener_cine);title('Energía Cinética');xlabel('time
(s)');ylabel('Energy (J)');
```

```
subplot(3,1,2)
plot(t,ener_pot_grav);title('Energía Potencial
Gravitatoria');xlabel('time (s)');ylabel('Energy (J)');
```

```
subplot(3,1,3)
plot(t,ener_tot);title('Energía Total');xlabel('time
(s)');ylabel('Energy (J)');
```

```
function cg_1 = CG(rg)
```

```
%Funcion que permite determinar los centros de gravedad locales
```

```
cg_1= zeros(20,2);
```

```
rg(:,2)=[];%eliminando la columna Y, paso de 2D a 3D;
```

```

cg_l(1,:) = rg(1,:);
cg_l(2,:) = rg(2,:);
cg_l(3,:) = rg(3,:);
cg_l(4,:) = rg(4,:);
cg_l(5,:) = rg(5,:);
cg_l(6,:) = rg(6,:);
cg_l(7,:) = rg(7,:);
cg_l(8,:) = rg(8,:);
cg_l(9,:) = rg(9,:);
cg_l(10,:) = rg(10,:);
cg_l(11,:) = rg(11,:);
cg_l(12,:) = rg(12,:);
cg_l(13,:) = rg(13,:);
cg_l(14,:) = rg(17,:);
cg_l(15,:) = rg(14,:);
cg_l(16,:) = rg(15,:);
cg_l(17,:) = rg(16,:);
cg_l(18,:) = rg(18,:);
cg_l(19,:) = rg(19,:);
cg_l(20,:) = rg(20,:);

```

```
end
```

```
function [df]=derivar(X,Y)
```

```
%Funcion que permite derivar dy/dx
```

```
N=numel(X);
```

```
df(1)=(Y(2)-Y(1))/(X(2)-X(1));
```

```
df(N)=(Y(N)-Y(N-1))/(X(N)-X(N-1));
```

```
for n=2:N-1
```

```
df(n)=(Y(n+1)-Y(n-1))/(X(n+1)-X(n-1));
```

```
end
```

```
% plot(X,df)
```

```
end
```

```
function [T,T_ast]= Dinamica_Inversa(t,qc,qs,zp,fpl1,fpl2,l,M,Qc,rt)
```

```
T=zeros(length(qc),23);%compuesto por reacciones, momentos y angulos relativos Ti
```

```
Qm = zeros(length(qc),20);%Q conocida, en este caso segmentos
```

```
xy = @(n) 2*n-1:2*n;
```

```
pv = @(w,v) w*[-v(2) v(1)];
```

```
ang = @(n) 42+n;
```

```
yp=zeros(length(qc),23);
```

```
order = [2 1 3 2 4 3 5 2 6 5 7 6 8 2 9 8 10 9 11 1 12 11 13 12 14 13 15 1 16 15 17 16 18 17];%Angulos relativos
```

```

B = zeros(23,20);
T_ast=zeros(length(qc),23);
W = zeros(23);
for i=1:length(qc)

    R = Matriz_R(qc(i,:),l,rt);
    Qm(i,:) = R'(M*qs(i,：)'-Qc(i,：)');

    %Cálculo de la matriz B
    for j=1:20
        B(1:2,j)=R(xy(13),j)'+pv(R(ang(13),j),[fp11(i,7) fp11(i,8)]-
qc(i,xy(13)));
        B(3,j)=R(ang(13),j);
        B(4:5,j)=R(xy(17),j)'+pv(R(ang(17),j),[fp12(i,7) fp12(i,8)]-
qc(i,xy(17)));
        B(6,j)=R(ang(17),j);
        for m=1:17

            B(6+m,j)=R(ang(order(2*m-1)),j)-R(ang(order(2*m)),j);

        end

    end

yp(i,：)=B*zp(i,：)';

%Problema de minimización, con restriccion [W B;B' 0]*[T lambda]=[WT*;Qm]
T_ast(i,1)=fp11(i,1);
T_ast(i,2)=fp11(i,3);
T_ast(i,3)=fp11(i,5);%momento aplicado en el COP, las fuerzas son
aplicadas en el COP
T_ast(i,4)=fp12(i,1);
T_ast(i,5)=fp12(i,3);
T_ast(i,6)=fp12(i,5);%momento aplicado en el COP, las fuerzas son
aplicadas en el COP

    % W
    for m=1:6

        %
        W(m,m)=1/(sqrt(abs(T_ast(i,m)))+1e-7);
        W(m,m)=abs(T_ast(i,m))^-0.5;
    end

A=[W B;B' zeros(20,20)];
b=[W*T_ast(i,：)';Qm(i,：)'];

x=A\b;
T(i,：)=x(1:23);

end
function [alfa,rg_g]=grafica_CG(qc,cg_l,1)
%grafica los CG, y devuelve el coeficiente de proximidad

alfa=zeros(1,20);
rg_g=zeros(1,40);
ind = @(n) 2*n-1:2*n; pointer = @(j,m) qc(j,ind(m));

```

```
order = [1 2 1 3 3 4 4 5 3 6 6 7 7 8 3 9 9 10 10 11 2 12 12 13 13 14 14
15 2 16 16 17 17 18 18 19 8 20 11 21];
```

```
for i=1:length(alfa)
    if i==4||i==14||i==18
        alfa(1,i)=1;%CG
    else
        alfa(1,i) = norm(cg_1(i,:))/l(i);
    end
end
for j=1:length(qc)
    for i=1:length(alfa)
        rg_g(j,ind(i))=pointer(j,order(2*i-
1))+alfa(i)*(pointer(j,order(2*i))-pointer(j,order(2*i-1)));
    end
end
```

```
figure(8); clf;
```

```
order1 = [5 4 3 6 7 8 20 8 7 6 3 1 2 12 13 14 15];
order2 = [3 9 10 11 21];
order3 = [2 16 17 18 19];
```

```
P1 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b.');
```

```
hold on
```

```
P2 = plot(qc(1,1:2:end), qc(1,2:2:end), 'bo');
```

```
P3 = plot(qc(1,1:2:end), qc(1,2:2:end), 'bo');
```

```
v1 = plot(qc(1,1:2:end),qc(1,2:2:end),'b-');
```

```
v2 = plot(qc(1,1:2:end),qc(1,2:2:end),'b--');
```

```
v3 = plot(qc(1,1:2:end),qc(1,2:2:end),'b--');
```

```
P1P = plot(rg_g(1,1:2:end),rg_g(1,2:2:end), 'ro');hold off
```

```
title('Plano Sagital')
grid on
axis([-1.5 2 -0.5 2.5]); set(gca, 'DataAspectRatio', [1 1 1])
set(gcf, 'Renderer', 'OpenGL')
% legend('derecha','izquierda')
```

```
x = qc(1,1:2:end);
y = qc(1,2:2:end);
```

```
set(P1, 'xdata', x(order1), 'ydata', y(order1))
set(P2, 'xdata', x(order2), 'ydata', y(order2))
set(P3, 'xdata', x(order3), 'ydata', y(order3))
```

```

        set(v1, 'xdata', x(order1), 'ydata', y(order1))
        set(v2, 'xdata', x(order2), 'ydata', y(order2))
        set(v3, 'xdata', x(order3), 'ydata', y(order3))

    end

function grafica_fuerza_placas(qc, fp11, fp12)

figure(5); clf;
k=5e-4;

order1 = [5 4 3 6 7 8 7 6 3 1 2 12 13 14 15];
order2 = [3 9 10 11];
order3 = [2 16 17 18 19];

P1 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b. '); hold on
P2 = plot(qc(1,1:2:end), qc(1,2:2:end), 'bo ');
P3 = plot(qc(1,1:2:end), qc(1,2:2:end), 'bo ');

v1 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b- ');
v2 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b-- ');
v3 = plot(qc(1,1:2:end), qc(1,2:2:end), 'b-- ');

P1D = plot(fp11(1,7)+k*fp11(1,1), k*fp11(1,3), 'k^ ');
P2D = plot(fp11(1,7), 0, 'k. ');
v1D = plot([fp11(1,7) fp11(1,7)+k*fp11(1,1)], [0 k*fp11(1,3)], 'k- ');

P1I = plot(fp12(1,7)+k*fp12(1,1), k*fp12(1,3), 'k^ ');
P2I = plot(fp12(1,7), 0, 'k. ');
v1I = plot([fp12(1,7) fp12(1,7)+k*fp12(1,1)], [0 k*fp12(1,3)], 'k- ');

hold off
title('Plano Sagital')
grid on
axis([-1.5 2 -0.5 2.5]); set(gca, 'DataAspectRatio', [1 1 1])
set(gcf, 'Renderer', 'OpenGL')

for i = 1:length(qc)
    x = qc(i,1:2:end);
    y = qc(i,2:2:end);
    xpd = fp11(i,7)+k*fp11(i,1);
    ypd = k*fp11(i,3);
    xppd=fp11(i,7);
    yppd=0;
    xpi = fp12(i,7)+k*fp12(i,1);
    ypi = k*fp12(i,3);
    xppi=fp12(i,7);
    yppi=0;

    set(P1, 'xdata', x(order1), 'ydata', y(order1))
    set(P2, 'xdata', x(order2), 'ydata', y(order2))
    set(P3, 'xdata', x(order3), 'ydata', y(order3))
    set(v1, 'xdata', x(order1), 'ydata', y(order1))
    set(v2, 'xdata', x(order2), 'ydata', y(order2))
    set(v3, 'xdata', x(order3), 'ydata', y(order3))
end

```

```

    if fp11(i,3)>100

        set(P1D, 'xdata', xpd, 'ydata', ypd)
        set(P2D, 'xdata', xppd, 'ydata', yppd)
        set(v1D, 'xdata', [fp11(i,7) fp11(i,7)+k*fp11(i,1)], 'ydata', [0
k*fp11(i,3)]);
        end

        if fp12(i,3)>100
        set(P1I, 'xdata', xpi, 'ydata', ypi)
        set(P2I, 'xdata', xppi, 'ydata', yppi)
        set(v1I, 'xdata', [fp12(i,7) fp12(i,7)+k*fp12(i,1)], 'ydata', [0
k*fp12(i,3)]);
        end

        drawnow
        pause(0.001)
    end
end

function grafica_trayectorias(r,q_proj,qc)
%Funcion que grafica las trayectorias de los markerts, la proyeccion y la
%qcomplete

figure(2); clf
for i=1:21
plot(r(:,2*i-1),r(:,2*i),'b',q_proj(:,2*i-1),q_proj(:,2*i),'r',qc(:,2*i-
1),qc(:,2*i),'g');hold on

end
grid on
lighting gouraud
set(gcf, 'Renderer', 'OpenGL')
legend('q*', 'q proyect', 'q complete')

end

function I = Inercia(MI, cg_l, m)
%calcula el momento de inercia con respecto al primer punto del segmento,
%osea stainer

MI(:, [1 3 4])=[];%se elimina MI con respecto al eje X, Z y la columna de
masa
I = zeros(20,1);

%Momento de inercia con respecto a CG de cada segmento
I(1,:) = MI(1,:);
I(2,:) = MI(2,:);
I(3,:) = MI(3,:);
I(4,:) = MI(4,:);
I(5,:) = MI(5,:);
I(6,:) = MI(6,:);

```

```

I(7,:) = MI(7,:);%MI de la mano derecha
I(8,:) = MI(8,:);
I(9,:) = MI(9,:);
I(10,:) = MI(10,:);%MI de la mano izquierda
I(11,:) = MI(11,:);
I(12,:) = MI(12,:);
I(13,:) = MI(13,:);
I(14,:) = MI(17,:);
I(15,:) = MI(14,:);
I(16,:) = MI(15,:);
I(17,:) = MI(16,:);
I(18,:) = MI(18,:);
I(19,:) = MI(19,:);
I(20,:) = MI(20,:);

%Aplicacion de Stainer
for i = 1:length(I)
    I(i,:) = I(i,:) + m(i) * norm(cg_1(i,:))^2;
end

end

function [l] = longitudes(r)
%funcion que calcula longitudes promedio de los segmentos

n=20;%número de segmentos en el sistema
l = zeros(1,n);
ind = @(i) 2*i-1:2*i;
long_muleta=0.889;%longitud de la muleta

l(1) = promedio(r(:,ind(1)),r(:,ind(2)));
l(2) = promedio(r(:,ind(1)),r(:,ind(3)));
l(3) = promedio(r(:,ind(3)),r(:,ind(4)));
l(4) = promedio(r(:,ind(4)),r(:,ind(5)));
l(5) = promedio(r(:,ind(3)),r(:,ind(6)));
l(6) = promedio(r(:,ind(6)),r(:,ind(7)));
l(7) = promedio(r(:,ind(7)),r(:,ind(8)));
l(8) = promedio(r(:,ind(9)),r(:,ind(3)));
l(9) = promedio(r(:,ind(10)),r(:,ind(9)));
l(10) = promedio(r(:,ind(11)),r(:,ind(10)));
l(11) = promedio(r(:,ind(12)),r(:,ind(2)));
l(12) = promedio(r(:,ind(13)),r(:,ind(12)));
l(13) = promedio(r(:,ind(14)),r(:,ind(13)));
l(14) = promedio(r(:,ind(15)),r(:,ind(14)));
l(15) = promedio(r(:,ind(16)),r(:,ind(2)));
l(16) = promedio(r(:,ind(17)),r(:,ind(16)));
l(17) = promedio(r(:,ind(18)),r(:,ind(17)));
l(18) = promedio(r(:,ind(19)),r(:,ind(18)));
l(19) = promedio(r(:,ind(8)),r(:,ind(20)));
l(20) = promedio(r(:,ind(11)),r(:,ind(21)));

function lm = promedio(a,b)
    m = zeros(size(a,1),1);

    for t = 1:size(a,1)

        m(t) = norm(a(t,:)-b(t,:));

```

```

end
lm=mean(m);

function markers(r,q_proj)

figure(1); clf;

order1 = [5 4 3 6 7 8 20 8 7 6 3 1 2 12 13 14 15];
order2 = [3 9 10 11 21];
order3 = [2 16 17 18 19];

P1 = plot(r(1,1:2:end), r(1,2:2:end), 'b.');
```

hold on

```

P2 = plot(r(1,1:2:end), r(1,2:2:end), 'bo');
```

P3 = plot(r(1,1:2:end), r(1,2:2:end), 'bo');

```

v1 = plot(r(1,1:2:end), r(1,2:2:end), 'b-');
v2 = plot(r(1,1:2:end), r(1,2:2:end), 'b--');
v3 = plot(r(1,1:2:end), r(1,2:2:end), 'b--');
```

```

P1P = plot(q_proj(1,1:2:end), q_proj(1,2:2:end), 'r.');
```

P2P = plot(q_proj(1,1:2:end), q_proj(1,2:2:end), 'ro');

P3P = plot(q_proj(1,1:2:end), q_proj(1,2:2:end), 'ro');

```

v1P = plot(q_proj(1,1:2:end), q_proj(1,2:2:end), 'r-');
v2P = plot(q_proj(1,1:2:end), q_proj(1,2:2:end), 'r--');
v3P = plot(q_proj(1,1:2:end), q_proj(1,2:2:end), 'r--');
```

```

hold off
title('Plano Sagital')
axis([-1.5 2 -0.5 2]); set(gca, 'DataAspectRatio', [1 1 1])
set(gcf, 'Renderer', 'OpenGL')
grid on

for i = 1:length(r)
    x = r(i,1:2:end);
    y = r(i,2:2:end);
    xp = q_proj(i,1:2:end);
    yp = q_proj(i,2:2:end);
    set(P1, 'xdata', x(order1), 'ydata', y(order1))
    set(P2, 'xdata', x(order2), 'ydata', y(order2))
    set(P3, 'xdata', x(order3), 'ydata', y(order3))
    set(v1, 'xdata', x(order1), 'ydata', y(order1))
    set(v2, 'xdata', x(order2), 'ydata', y(order2))
    set(v3, 'xdata', x(order3), 'ydata', y(order3))

    set(P1P, 'xdata', xp(order1), 'ydata', yp(order1))
    set(P2P, 'xdata', xp(order2), 'ydata', yp(order2))
    set(P3P, 'xdata', xp(order3), 'ydata', yp(order3))
    set(v1P, 'xdata', xp(order1), 'ydata', yp(order1))
    set(v2P, 'xdata', xp(order2), 'ydata', yp(order2))
    set(v3P, 'xdata', xp(order3), 'ydata', yp(order3))

    drawnow
    pause(0.001)
end

```

```
        %grabar printscreen de figure1
    %     P=num2str(i);
    %     saveas (figure(1),P,'png');
end

function [m] = masas(MI)
%funcion que permite calcular las masas de los segmentos corporales
m = zeros(1,20);

%cadera
m(1)=MI(1,4);

%tronco
m(2)=MI(2,4);

%cuello
m(3)=MI(3,4);

%cabeza
m(4)=MI(4,4);

%Brazo derecho
m(5)=MI(5,4);

%Antebrazo derecho
m(6)=MI(6,4);

%Mano derecha
m(7)=MI(7,4);

%Brazo izquierdo
m(8)=MI(8,4);

%Antebrazo izquierdo
m(9)=MI(9,4);

%mano izquierda
m(10)=MI(10,4);

%pierna derecha
m(11)=MI(11,4);

%pantorrilla derecha
m(12)=MI(12,4);

%Pie derecho
m(13)=MI(13,4);

%punta pie derecho
m(14)=MI(17,4);

%pierna izquierda
m(15)=MI(14,4);

%pantorrilla izquierda
```

```

m(16)=MI(15,4);

%pie izquierdo
m(17)=MI(16,4);

%punta pie izquierdo
m(18)=MI(18,4);

%muleta derecha
m(19)=MI(19,4);

%muleta izquierda
m(20)=MI(20,4);

end

function [Qc] = matriz_de_fuerzas(m,l,alfa,ref1,ref11,ref2,t,qc)
%Funcion que calcula la matriz de fuerzas, solo el peso
%gravedad
g=9.81;
Q=zeros(42,1);
order = [1 2 1 3 3 4 4 5 3 6 6 7 7 8 3 9 9 10 10 11 2 12 12 13 13 14 14
15 2 16 16 17 17 18 18 19 8 20 11 21];
ind = @(n) 2*n-1:2*n; pointer = @(m) order(ind(m));

for i=1:length(l)

b=pointer(i);
x=b(1); y_izq=b(2);
Q([ind(x),ind(y_izq)])=Q([ind(x),ind(y_izq)])-m(i)*g*[0 1-alfa(i) 0
alfa(i)]';

end

Qc=zeros(length(t),60);

%Peso de segmentos
for i=1:length(t)
Qc(i,1:42)=Q(:,1)';
end

%Torque en las rodillas
%-----
--
% Time-step (data is resampled at 100 Hz)
h = 0.01;

KP = 1e5; % Proportional gain (was 566)
KI = 1e2; % Integral gain (was 1412)
KD = 1e5; % Derivative gain (was 1208)

% Motor torque constant (Nm/A)
Km = 156*25.1e-3;

% Acceleration feedforward gain (1e6*J/Km)
m = 5.6771; l = 0.5924;

```

```

Ka = 1e6*m*1^2/Km;

% Conversion from EPOS2 to SI units (A, rad, s)
KP = 10e-3*KP; KI = 78e-3*KI;
KD = 80e-6*KD; Ka = 1e-6*Ka;

% Control parameters in PID form
K0 = 16*KP*h+KD;
K1 = KD/K0;
K2 = (16*KP^2+KD*KI)*h/K0;
K3 = 17*KP*KD*h/K0;
K4 = 1e-12;
% K4 = 16*KP*KI*h/K0;

%Ángulo deseado
r=ref1(:,1)*pi/180+pi/2;
r1=ref11(:,1)*pi/180+pi/2;
r2=ref2(:,1)*pi/180+pi/2;

%duplicamos la curva, posicion del angulo
ang_izq=zeros(size(t))+pi/2;
ang_izq(1:142)=r(10:151);
ang_izq(390:540)=r1;

ang_der=zeros(size(t))+pi/2;
ang_der(190:340) = r2;

%Aceleración del ángulo
ace_izq = zeros(size(t));
ace_der = zeros(size(t));
ace_izq(1:142)=ref1(10:151,3)*pi/180;
ace_izq(390:540)=ref11(:,3)*pi/180;
ace_der(190:340)=ref2(:,3)*pi/180;

torque = zeros(length(t),2);%Momentos de la rodilla izquierda y derecha

I = zeros(length(t),2);
se = 0; e0 = 0;

for i=2:609 %Par pierna izquierda

    % PID to calculate input torque
    e0=ang_izq(i) - (qc(i,58));
    e_izq = ang_izq(i) - (qc(i,58));
    de = (e_izq-e0)/h;
    se = se + h*e_izq;
    I(i,1) = [K1 K2 K3 K4 Ka]*[I(i-1,1) e_izq de se ace_izq(i)]';
    torque(i,1) = Km*[K1 K2 K3 K4 Ka]*[I(i-1,1) e_izq de se
ace_izq(i)]';

end
se = 0; e0 = 0;

```

```

for i=2:609 %Par pierna derecha

    % PID to calculate input torque
    e0=ang_der(i) - (qc(i,54));
    e_der = ang_der(i) - (qc(i,54));
    de = (e_der-e0)/h;
    se = se + h*e_der;
    I(i,2) = [K1 K2 K3 K4 Ka]*[I(i-1,2) e_der de se ace_der(i)]';
    torque(i,2) = Km*[K1 K2 K3 K4 Ka]*[I(i-1,2) e_der de se
ace_der(i)]';

end

Qc(:,54)=Qc(:,54)/9.81;
Qc(:,58)=Qc(:,58)/9.81;

end

function [M] = Matriz_de_masa(ma,cg_l,I,l)
%Matriz de masas generalizada
order = [1 2 1 3 3 4 4 5 3 6 6 7 7 8 3 9 9 10 10 11 2 12 12 13 13 14 14
15 2 16 16 17 17 18 18 19 8 20 11 21];
M = zeros(60);
ind = @(n) 2*n-1:2*n; pointer = @(m) order(ind(m));

for i = 1:length(l)
    x = mass(ma(i),cg_l(i,:),I(i),l(i));
    b=pointer(i);
    m=b(1);n=b(2);
    M(ind(m),ind(m))=M(ind(m),ind(m))+x(ind(1),ind(1));
    M(ind(n),ind(m))=M(ind(n),ind(m))+x(ind(1),ind(2));
    M(ind(m),ind(n))=M(ind(m),ind(n))+x(ind(2),ind(1));
    M(ind(n),ind(n))=M(ind(n),ind(n))+x(ind(2),ind(2));

end

function a = mass(m1,cg_l1,I1,l1)
    a = [m1-(2*m1*cg_l1(1)/l1)+I1/(l1^2),0,(m1*cg_l1(1)/l1)-
I1/(l1^2), -m1*cg_l1(2)/(l1); 0,m1-
(2*m1*cg_l1(1)/l1)+I1/(l1^2),m1*cg_l1(2)/(l1), (m1*cg_l1(1)/l1)-
I1/(l1^2); (m1*cg_l1(1)/l1)-I1/(l1^2),m1*cg_l1(2)/(l1),
I1/(l1^2),0; -m1*cg_l1(2)/(l1), (m1*cg_l1(1)/l1)-I1/(l1^2),0,I1/(l1^2)];
end
end

function [R] = Matriz_R(qc,l,rt)
%Calculo de la matriz R

R=zeros(60,20);

inc=3:42;

```

```

[Phic,Phiqc] = restric_c(qc,l,rt);
R(inc,:)=Phiqc(:,inc)\-Phiqc(:, [1:2,43:60]);
R([1:2,43:60],:)=eye(20);

```

```
end
```

```

function [r] = modificacion_r0(r0,AM,r1)
%Función que calcula los r0 para CDG de la cabeza, manos y pies, y las
%adiciona a la matriz de r0.

A=@(t,i) reshape(AM(t,9*i-8:9*i),[3 3]);%función anónima que forma
matrices
%de rotacion de 3x3

r1=zeros(length(r0),9);
ind = @(i) 3*i-1:3*i+1;%indice que apunta en la matriz r0 para coger las
%coordenadas

%columna 1: Cabeza, Columna 2: Pie derecho, Columna 3: Pie izquierdo.
r0p=[7 16 17];
AMp=[4 17 18];
rlp=[8 21 22];

sub = @(n) 3*n-2:3*n;

%Formación de la matriz r de los puntos de la cabeza, manos y pies
for n = 1:length(r0p)
    for t = 1:size(r0,1)

r1(t,sub(n))=((r0(t,ind(r0p(n))))'+A(t,AMp(n))*(rl(rlp(n),:))')')';
    end
end

%Formación de la matriz r general, union de r0 y r1

r = zeros(length(r0),19*3);

r(:,sub(1))=r0(:,ind(1));
for i=1:length(r0)
    r(i,sub(2))=mean([r0(i,ind(2));r0(i,ind(3))]);
    r(i,sub(3))=mean([r0(i,ind(4));r0(i,ind(5));r0(i,ind(6))]);
end
r(:,sub(4))=r0(:,ind(7));
r(:,sub(5))=r1(:,sub(1));
r(:,sub(6))=r0(:,ind(8));
r(:,sub(7))=r0(:,ind(9));
r(:,sub(8))=r0(:,ind(18));
r(:,sub(9))=r0(:,ind(10));
r(:,sub(10))=r0(:,ind(11));
r(:,sub(11))=r0(:,ind(19));
r(:,sub(12))=r0(:,ind(12));
r(:,sub(13))=r0(:,ind(13));
r(:,sub(14))=r0(:,ind(16));
r(:,sub(15))=r1(:,sub(2));

```

```

r(:,sub(16))=r0(:,ind(14));
r(:,sub(17))=r0(:,ind(15));
r(:,sub(18))=r0(:,ind(17));
r(:,sub(19))=r1(:,sub(3));

%Calculo de los puntos extremos de las muletas
lmuleta=0.889;%metros
rmuleta=zeros(length(r0),6);

r0p=[8 11];
AMp=[19 20];
    %termino de la crutch
    w=@(v) v(:,3);
    for n=1:2
        for i=1:length(r0)
            rmuleta(i,sub(n))=(r(i,sub(r0p(n)))-(w(A(i,AMp(n)))*lmuleta)');
        end
    end
end

r(:,sub(20))=rmuleta(:,sub(1));
r(:,sub(21))=rmuleta(:,sub(2));

r(:,2:3:end)=[];%se elimina la columna Y, 3D a 2D

end

function [r] = modificacion_rl(r0,AM,rl)
A=@(t,i) reshape(AM(t,9*i-8:9*i),[3 3]);%función anónima que forma
matrices de rotacion de 3x3

r=zeros(length(r0),15);
ind = @(i) 3*i-1:3*i+1;

    for t = 1:length(r0)
        r(t,1:3)=(r0(t,ind(7))'+A(t,4)*rl(8,:))';%CDG de la cabeza
        r(t,4:6)=(r0(t,ind(9))'+A(t,7)*rl(11,:))';%CDG mano derecha
        r(t,7:9)=(r0(t,ind(11))'+A(t,10)*rl(14,:))';%CDG mano izquierda
        r(t,10:12)=(r0(t,ind(16))'+A(t,17)*rl(21,:))';%CDG pie derecho
        r(t,13:15)=(r0(t,ind(17))'+A(t,18)*rl(22,:))';%CDG pie izquierdo
    end

end

function [qc]=qcomplete(q_proj,l,rt)

%Ángulos iniciales
theta = zeros(1,18);

```

```

theta(1)=pi/2; theta(2)=pi/2; theta(3)=pi/2; theta(4)=pi/4;
theta(5)=pi/6;
theta(6)=pi/2; theta(7)=pi/2; theta(8)=pi/6; theta(9)=pi/2;
theta(10)=pi/2;
theta(11)=pi/2; theta(12)=pi/3; theta(13)=pi/6; theta(14)=pi/6;
theta(15)=pi/2;
theta(16)=pi/3; theta(17)=pi/6; theta(18)=pi/6;

qc=zeros(length(q_proj),60);
grados=zeros(length(q_proj),18);
%=====
==
for i=1:length(q_proj)

    qc(i,:)=[q_proj(i,:),theta]';

    [Phic,Phiqc] = restric_c(qc(i,:),l,rt);

    error=norm(Phic); tol=1e-10;

    while error > tol

        inc = 43:60;
        A=Phiqc'*Phiqc;
        B=-Phiqc'*Phic;
        deltaq=A(inc, inc)\B(inc);
        qc(i,inc)=qc(i,inc)+deltaq';

        [Phic, Phiqc]=restric_c(qc(i,:),l,rt);

        error=norm(Phic);

    end

%    grados(i,:)=qc(i,39:56)*180/pi;%si se quiere los angulos en grados
%    centesimales

%=====
==
end
function [q_proj]= qproj(r,l,rt)

% q proyectada, los puntos que no cambian mucho son los tobillos
num=21;%número de puntos
seg=22;%número de restricciones
m=num*2;

q_proj = zeros(length(r), m);

for i=1:length(r)
    tol=1e-12; alpha=1e9;

    % Estimación de la posición inicial

```

```

q=r(i, :)' ;
q0=q;

% Proyección de la posición
% peso en los puntos que no varían mucho
a=ones(1,m);
ind = @(n) 2*n-1:2*n;
a(ind(13))=100;
a(ind(17))=100;
a(ind( 8))=100;
a(ind(11))=100;

W=diag(a); lambda=zeros(seg,1); ite=0;

[Phi, Phiq]=restric(q,l,rt);

error=norm(Phi);

while error > tol
    if ite>0
        lambda=lambda+alpha*Phi;
    end
    Tan=W+Phiq'*alpha*Phiq;
    res=W*(q-q0)+Phiq'*(alpha*Phi+lambda);
    deltaq=-Tan\res; q=q+deltaq;

    [Phi, Phiq]=restric(q,l,rt);

    error=norm(Phi);

    ite=ite+1;
end
q_proj(i,:)=q';
end

end

function rt = relativo_muleta(r)
%funcion que retorna la posición promedio de las posiciones
%del origen de la muleta con respecto a la mano.

rt = zeros(1,4);
pos_rt_der = zeros(length(r),2);
pos_rt_izq = zeros(length(r),2);
mano_ini_p = [7 10];
crutch_ini_p = [8 11];
crutch_fin_p = [20 21];
ind = @(i,n) r(i,2*n-1:2*n);
A = @(a,b) (1/norm(a-b))*[[a(1)-b(1);a(2)-b(2)], [b(2)-a(2);a(1)-b(1)]];

```

```

for i = 1:length(r)

    pos_rt_der(i,:) =
A(ind(i,mano_ini_p(1)),ind(i,crutch_ini_p(1)))'*((ind(i,crutch_fin_p(1))-
ind(i,crutch_ini_p(1)))');
    pos_rt_izq(i,:) =
A(ind(i,mano_ini_p(2)),ind(i,crutch_ini_p(2)))'*((ind(i,crutch_fin_p(2))-
ind(i,crutch_ini_p(2)))');

end

rt(1:2) = mean(pos_rt_der);
rt(3:4) = mean(pos_rt_izq);

function [Phi, Phiq]=restric(q1,l,rt)

%Funcion que devuelve la matriz Phi y Phiq

ind = @(i) 2*i-1:2*i;
r = @(i) q1(ind(i));
x = @(i) q1(2*i-1);
y = @(i) q1(2*i);
A1 = @(i,j) [x(i)-x(j) y(i)-y(j)];
A2 = @(i,j) [y(j)-y(i) x(i)-x(j)];

    Phi = [(r(1)-r(2))'*(r(1)-r(2))-l(1)^2;
            (r(1)-r(3))'*(r(1)-r(3))-l(2)^2;
            (r(3)-r(4))'*(r(3)-r(4))-l(3)^2;
            (r(4)-r(5))'*(r(4)-r(5))-l(4)^2;
            (r(3)-r(6))'*(r(3)-r(6))-l(5)^2;
            (r(6)-r(7))'*(r(6)-r(7))-l(6)^2;
%            (r(7)-r(8))'*(r(7)-r(8))-l(7)^2;
            (r(3)-r(9))'*(r(3)-r(9))-l(8)^2;
            (r(9)-r(10))'*(r(9)-r(10))-l(9)^2;
%            (r(10)-r(11))'*(r(10)-r(11))-l(10)^2;
            (r(2)-r(12))'*(r(2)-r(12))-l(11)^2;
            (r(12)-r(13))'*(r(12)-r(13))-l(12)^2;
            (r(13)-r(14))'*(r(13)-r(14))-l(13)^2;
            (r(14)-r(15))'*(r(14)-r(15))-l(14)^2;
            (r(2)-r(16))'*(r(2)-r(16))-l(15)^2;
            (r(16)-r(17))'*(r(16)-r(17))-l(16)^2;
            (r(17)-r(18))'*(r(17)-r(18))-l(17)^2;
            (r(18)-r(19))'*(r(18)-r(19))-l(18)^2;
            (r(8)-r(20))'*(r(8)-r(20))-l(19)^2; %muleta derecha
            (r(11)-r(21))'*(r(11)-r(21))-l(20)^2;%muleta izquierda
            A1(7,8)*(r(20)-r(8))-l(7)*rt(1); %posicion xt de muleta
DERECHA relativa a su origen
            A2(7,8)*(r(20)-r(8))-l(7)*rt(2); %posicion yt de muleta
DERECHA relativa a su origen
            A1(10,11)*(r(21)-r(11))-l(10)*rt(3); %posicion xt de muleta
IZQUIERDA relativa a su origen
            A2(10,11)*(r(21)-r(11))-l(10)*rt(4)];%posicion yt de muleta
IZQUIERDA relativa a su origen

```

```

if nargout == 2
    Phi = zeros(length(Phi),length(q1));
    Phiq(1,[ind(1) ind(2)]) = [2*(r(1)-r(2))' -2*(r(1)-r(2))'];
    Phiq(2,[ind(1) ind(3)]) = [2*(r(1)-r(3))' -2*(r(1)-r(3))'];
    Phiq(3,[ind(3) ind(4)]) = [2*(r(3)-r(4))' -2*(r(3)-r(4))'];
    Phiq(4,[ind(4) ind(5)]) = [2*(r(4)-r(5))' -2*(r(4)-r(5))'];
    Phiq(5,[ind(3) ind(6)]) = [2*(r(3)-r(6))' -2*(r(3)-r(6))'];
    Phiq(6,[ind(6) ind(7)]) = [2*(r(6)-r(7))' -2*(r(6)-r(7))'];
%    Phiq(7,[ind(7) ind(8)]) = [2*(r(7)-r(8))' -2*(r(7)-r(8))'];
    Phiq(7,[ind(3) ind(9)]) = [2*(r(3)-r(9))' -2*(r(3)-r(9))'];
    Phiq(8,[ind(9) ind(10)]) = [2*(r(9)-r(10))' -2*(r(9)-
r(10))'];
%    Phiq(10,[ind(10) ind(11)]) = [2*(r(10)-r(11))' -2*(r(10)-
r(11))'];
    Phiq(9,[ind(2) ind(12)]) = [2*(r(2)-r(12))' -2*(r(2)-
r(12))'];
    Phiq(10,[ind(12) ind(13)]) = [2*(r(12)-r(13))' -2*(r(12)-
r(13))'];
    Phiq(11,[ind(13) ind(14)]) = [2*(r(13)-r(14))' -2*(r(13)-
r(14))'];
    Phiq(12,[ind(14) ind(15)]) = [2*(r(14)-r(15))' -2*(r(14)-
r(15))'];
    Phiq(13,[ind(2) ind(16)]) = [2*(r(2)-r(16))' -2*(r(2)-
r(16))'];
    Phiq(14,[ind(16) ind(17)]) = [2*(r(16)-r(17))' -2*(r(16)-
r(17))'];
    Phiq(15,[ind(17) ind(18)]) = [2*(r(17)-r(18))' -2*(r(17)-
r(18))'];
    Phiq(16,[ind(18) ind(19)]) = [2*(r(18)-r(19))' -2*(r(18)-
r(19))'];
    Phiq(17,[ind(8) ind(20)]) = [2*(r(8)-r(20))' -2*(r(8)-
r(20))'];
    Phiq(18,[ind(11) ind(21)]) = [2*(r(11)-r(21))' -2*(r(11)-
r(21))'];
    Phiq(19,[ind(7) ind(8) ind(20)]) = [(r(20)-r(8))' (2*r(8)-
r(7)-r(20))' (r(7)-r(8))'];
    Phiq(20,[ind(7) ind(8) ind(20)]) = [y(20)-y(8) x(8)-x(20)
y(7)-y(20) x(20)-x(7) y(8)-y(7) x(7)-x(8)];
    Phiq(21,[ind(10) ind(11) ind(21)]) = [(r(21)-r(11))'
(2*r(11)-r(10)-r(21))' (r(10)-r(11))'];
    Phiq(22,[ind(10) ind(11) ind(21)]) = [y(21)-y(11) x(11)-x(21)
y(10)-y(21) x(21)-x(10) y(11)-y(10) x(10)-x(11)];
end

```

```
function [Phic,Phiqc] = restric_c(q,l,rt)
```

```

q0=q(1:42);
theta=q(43:60);
[Phi, Phiq] = restric(q0',l,rt);

```

```

x = @(n) q(2*n-1);
y = @(m) q(2*m);

```

```

Phic = [Phi;
x(1)-x(2)+cos(theta(1))*l(1);
x(1)-x(3)+cos(theta(2))*l(2);
x(3)-x(4)+cos(theta(3))*l(3);

```

```

y(4)-y(5)+sin(theta(4))*1(4);
y(3)-y(6)+sin(theta(5))*1(5);
x(6)-x(7)+cos(theta(6))*1(6);
x(8)-x(20)+cos(theta(7))*1(19);
y(3)-y(9)+sin(theta(8))*1(8);
x(9)-x(10)+cos(theta(9))*1(9);
x(11)-x(21)+cos(theta(10))*1(20);
x(2)-x(12)+cos(theta(11))*1(11);
x(12)-x(13)+cos(theta(12))*1(12);
y(13)-y(14)+sin(theta(13))*1(13);%ecuacion redundante
x(13)-x(14)+cos(theta(13))*1(13);%ecuacion redundante
y(14)-y(15)+sin(theta(14))*1(14);%ecuacion redundante
x(14)-x(15)+cos(theta(14))*1(14);%ecuacion redundante
x(2)-x(16)+cos(theta(15))*1(15);
x(16)-x(17)+cos(theta(16))*1(16);
y(17)-y(18)+sin(theta(17))*1(17);%ecuacion redundante
x(17)-x(18)+cos(theta(17))*1(17);%ecuacion redundante
y(18)-y(19)+sin(theta(18))*1(18);%ecuacion redundante
x(18)-x(19)+cos(theta(18))*1(18)];%ecuacion redundante

```

```

if nargin == 2
    Phiqc = zeros(44,60);
    Phiqc(1:22,1:42) = Phiq;
    Phiqc(23,1:3) = [1 0 -1];
    Phiqc(23,43) = -1(1)*sin(theta(1));
    Phiqc(24,1:5) = [1 0 0 0 -1];
    Phiqc(24,44) = -1(2)*sin(theta(2));
    Phiqc(25,5:7) = [1 0 -1];
    Phiqc(25,45) = -1(3)*sin(theta(3));
    Phiqc(26,8:10) = [1 0 -1];
    Phiqc(26,46) = 1(4)*cos(theta(4));
    Phiqc(27,6:12) = [1 0 0 0 0 0 -1];
    Phiqc(27,47) = 1(5)*cos(theta(5));
    Phiqc(28,11:13) = [1 0 -1];
    Phiqc(28,48) = -1(6)*sin(theta(6));
    Phiqc(29,15) = 1; Phiqc(29,39)=-1;
    Phiqc(29,49) = -1(19)*sin(theta(7));
    Phiqc(30,6:18) = [1 0 0 0 0 0 0 0 0 0 0 0 -1];
    Phiqc(30,50) = 1(8)*cos(theta(8));
    Phiqc(31,17:19) = [1 0 -1];
    Phiqc(31,51) = -1(9)*sin(theta(9));
    Phiqc(32,21) = 1; Phiqc(32,41)=-1;
    Phiqc(32,52) = -1(20)*sin(theta(10));
    Phiqc(33,3) = 1; Phiqc(33,23)=-1;
    Phiqc(33,53) = -1(11)*sin(theta(11));
    Phiqc(34,23:25) = [1 0 -1];
    Phiqc(34,54) = -1(12)*sin(theta(12));
    Phiqc(35,26:28) = [1 0 -1];
    Phiqc(35,55) = 1(13)*cos(theta(13));
    Phiqc(36,25:27) = [1 0 -1];
    Phiqc(36,55) = -1(13)*sin(theta(13));
    Phiqc(37,28:30) = [1 0 -1];
    Phiqc(37,56) = 1(14)*cos(theta(14));
    Phiqc(38,27:29) = [1 0 -1];
    Phiqc(38,56) = -1(14)*sin(theta(14));
    Phiqc(39,3) = 1; Phiqc(39,31)=-1;
    Phiqc(39,57) = -1(15)*sin(theta(15));

```

```

    Phiqc(40,31:33) = [1 0 -1];
    Phiqc(40,58) = -l(16)*sin(theta(16));
    Phiqc(41,34:36) = [1 0 -1];
    Phiqc(41,59) = l(17)*cos(theta(17));
    Phiqc(42,33:35) = [1 0 -1];
    Phiqc(42,59) = -l(17)*sin(theta(17));
    Phiqc(43,36:38) = [1 0 -1];
    Phiqc(43,60) = l(18)*cos(theta(18));
    Phiqc(44,35:37) = [1 0 -1];
    Phiqc(44,60) = -l(18)*sin(theta(18));

end

end

function [L] = segmentos(rl)
%Longitudes de los segmentos
L=zeros(18,1);

%segmento del tronco y cabeza
L(1)=norm(rl(1,:)-rl(2,:));
L(2)=norm(rl(1,:)-rl(3,:));
L(3)=norm(rl(3,:)-rl(4,:));
L(4)=norm(rl(4,:)-rl(5,:));

%segmentos de los brazos
%Derecho
L(5)=norm(rl(6,:)-rl(3,:));
L(6)=norm(rl(7,:)-rl(6,:));
L(7)=norm(rl(8,:)-rl(7,:));
%Izquierdo
L(8)=norm(rl(9,:)-rl(3,:));
L(9)=norm(rl(10,:)-rl(9,:));
L(10)=norm(rl(11,:)-rl(10,:));

function [y, r, vr] = ssa(xl, L)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% -----
%   Author: Francisco Javier Alonso Sanchez   e-mail:fjas@unex.es
%   Departament of Electronics and Electromecanical Engineering
%   Industrial Engineering School
%   University of Extremadura
%   Badajoz
%   Spain
% -----
%
% SSA generates a trajectory matrix X from the original series xl
% by sliding a window of length L. The trajectory matrix is approximated
% by using Singular Value Decomposition. The last step reconstructs
% the series from the approximated trayectory matrix. The SSA
applications
% include smoothing, filtering, and trend extraction.
% The algorithm used is described in detail in: Golyandina, N.,
Nekrutkin,
% V., Zhigljavsky, A., 2001. Analisis of Time Series Structure - SSA and

```

```

% Related Techniques. Chapman & Hall/CR.

% x1 Original time series (column vector form)
% L Window length
% y Reconstructed time series
% r Residual time series r = x1 - y
% vr Relative value of the norm of the approximated trajectory matrix
with
% respect to the original trajectory matrix

% The program output is the Singular Spectrum of x1 (must be a column
% vector), using a window length L. You must choose the components used
% to reconstruct the series in the form [i1,i2:ik,...,iL], based on the
% Singular Spectrum appearance.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

% Step 1: Build trajectory matrix

N = length(x1);
if L > N/2
    L = N - L;
end
K = N - L + 1;
X = zeros(L, K);
for i = 1:K
    X(1:L, i) = x1(i:L+i-1);
end

% Step 2: SVD

U = X * X';
[U, autoval] = eig(U);
[d, i] = sort(-diag(autoval));
U = U(:, i);
V = X' * U;

% Step 3: Grouping

% I = input(['Component agrupation to reconstruct the series ' ...
% 'in the form I = [i1, i2:ik, ..., iL] ']);
I = 1:2;
rca = U(:, I) * V(:, I)';

% Step 4: Reconstruction

y = zeros(N, 1);
Lp = min(L, K);
Kp = max(L, K);

for k = 0:Lp-2
    for m = 1:k+1
        y(k+1) = y(k+1) + rca(m, k-m+2) / (k+1);
    end
end
end

```

```

for k = Lp-1:Kp-1
    for m = 1:Lp
        y(k+1) = y(k+1) + rca(m,k-m+2) / Lp;
    end
end

for k = Kp:N
    for m = k-Kp+2:N-Kp+1;
        y(k+1) = y(k+1) + rca(m,k-m+2) / (N-k);
    end
end

if nargout > 1
    d = -d;
    sev = sum(d);
    r = x1 - y;
    vr = sum(d(I)) / sev * 100;

    figure(1)

    semilogy(100*d./sev); hold on; semilogy(100*d./sev, 'rx'); hold off
    title('Singular Spectrum')
    xlabel('Eigenvalue Number')
    ylabel('Eigenvalue (% norm of trajectory matrix retained)')

    figure(2)

    subplot(2, 1, 1)
    plot(x1); hold on; plot(y, 'r'); hold off
    xlabel('Data point')
    ylabel('Original and reconstructed series')

    subplot(2, 1, 2)
    plot(r, 'g')
    xlabel('Data point')
    ylabel('Residual series')
end
function [z]=zeta(qc)

[m,n]=size(qc);

z=zeros(m,20);

for i=1:m

    z(i,:)=[qc(i,1:2) qc(i,43:60)];

end

```