

Proyecto Fin de Grado

Grado en Ingeniería de Tecnologías Industriales

Optimización de parámetros de un modelo de contacto pie-suelo mediante el uso de un algoritmo evolutivo

Autor: Eduardo Maza Ortega

Tutor: Joaquín Ojeda Granja

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

Sevilla, 2016



Proyecto Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Optimización de parámetros de un modelo de contacto pie-suelo mediante el uso de un algoritmo evolutivo

Autor:

Eduardo Maza Ortega

Tutor:

Joaquín Ojeda Granja

Profesor ayudante doctor

Dep. Ingeniería Mecánica y Fabricación

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016

Proyecto Fin de Grado: Optimización de parámetros de un modelo de contacto pie-suelo mediante el uso de un algoritmo evolutivo

Autor: Eduardo Maza Ortega

Tutor: Joaquín Ojeda Granja

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal

A mi familia

A mis amigos

Resumen

El objetivo del presente trabajo es obtener un modelo del pie que reproduzca de forma correcta las fuerzas de reacción normales del suelo durante la pisada. Se utiliza para ello un modelo cinemático multisegmento del pie y un modelo de esferas con características viscoelásticas para el cálculo de las fuerzas. La fuerza vertical y los momentos asociados calculados se comparan con los valores experimentales obtenidos en el Laboratorio de Ingeniería Mecánica de la Universidad de Sevilla. El error es una suma de los errores cometidos al calcular la fuerza vertical, el momento en X y el momento en Y. La optimización de los parámetros del modelo tiene como objetivo minimizar dicho error y se hace en MATLAB usando el algoritmo evolutivo CMA-ES. El proceso de optimización ha consistido básicamente en ir modificando el estimado inicial y los límites de búsqueda del algoritmo para ir mejorando los resultados. El número de esferas ha ido aumentando: primero 3, luego 6 y por último 9, produciéndose mejoras notables con el aumento. Se presenta el mejor resultado obtenido, que corresponde al caso de 9 esferas. Este resultado se compara con el obtenido en un TFG previo que calcula las fuerzas de manera similar aunque en un modelo monosegmento, poniendo de manifiesto las bondades del modelo multisegmento utilizado aquí. Por último se analiza el mejor resultado obtenido y se sacan conclusiones para una mejora futura de los resultados.

Índice

Resumen	ix
Índice	xi
Índice de Tablas	xiii
Índice de Figuras	xv
1 Introducción	1
1.1 <i>Motivación</i>	1
1.2 <i>Revisión Bibliográfica</i>	1
1.3 <i>Objetivo</i>	2
2 Ciclo de Marcha y Obtención de Datos	3
2.1 <i>El ciclo de marcha humana</i>	3
2.2 <i>Obtención de datos</i>	4
2.3 <i>Ejes Coordinados</i>	7
2.3.1 <i>Ejes anatómicos</i>	7
2.3.2 <i>Ejes globales</i>	7
2.3.3 <i>Ejes locales</i>	7
3 Modelo del pie	9
3.1 <i>Modelo cinemático</i>	9
3.2 <i>Cálculo de las fuerzas de contacto</i>	10
4 El Algoritmo Evolutivo CMA-ES	13
4.1 <i>Actualización del centro de la distribución</i>	14
4.2 <i>Actualización de la matriz de covarianza</i>	14
4.3 <i>Actualización del tamaño de paso</i>	15
5 Proceso de Optimización	17
6 Resultado y Conclusiones	25
Referencias	35
Anexo A	37
<i>Rutina busqueda_minimo</i>	37
<i>Función optimizacion_ESFERAS_Oxford_normal_y_momentos</i>	39
<i>Función GRF_oxford_normal_esferas_mod</i>	40
<i>Rutina postprocesado</i>	43

<i>Función plot_fuerzas_y_esferas</i>	43
<i>Función dibuja_esferas</i>	46
<i>Función plot_estatico</i>	55
<i>Función analisis</i>	58
Anexo B	61

ÍNDICE DE TABLAS

Tabla 1. Mejor resultado con 3 esferas	19
Tabla 2. Errores del modelo de 3 esferas	19
Tabla 3. Vector inicial del modelo de 6 esferas	20
Tabla 4. Límites iniciales del modelo de 6 esferas	20
Tabla 5. Mejor resultado con 6 esferas	20
Tabla 6. Comparación entre los modelos de 3 y 6 esferas	21
Tabla 7. Vector inicial del modelo de 9 esferas	22
Tabla 8. Nuevos límites para k y c	22
Tabla 9. Mejor resultado con 9 esferas	25
Tabla 10. Límites utilizados para llegar al mejor resultado con 9 esferas	26
Tabla 11. Comparación entre los modelos de 3, 6 y 9 esferas	32
Tabla 12. Comparación entre los modelos mono y multisegmento	33

ÍNDICE DE FIGURAS

Figura 1. Ciclo de marcha	4
Figura 2. Marcadores retrorreflectores	5
Figura 3. Marcadores del modelo	6
Figura 4. Ejes anatómicos	7
Figura 5. Ejes globales	7
Figura 6. Ejes locales	8
Figura 7. Huesos del pie	9
Figura 8. Pie de Oxford	10
Figura 9. Marcadores del pie	10
Figura 10. Cálculo de la penetración	11
Figura 11. Algoritmo CMA-ES: generación de candidatos, estimación y nueva generación	13
Figura 12. Fuerzas modelo 3 esferas	19
Figura 13. Fuerzas modelo 6 esferas	21
Figura 14. Fuerzas modelo 9 esferas	26
Figura 15. Contribución de cada esfera	27
Figura 16. Fuerza elástica de cada esfera	28
Figura 17. Fuerza de amortiguamiento de cada esfera	28
Figura 18. Pseudopenetración de cada esfera	29
Figura 19. Vista de perfil	29
Figura 20. Vista del ensayo estático	30
Figura 21. Representación en locales de las esferas del HF	31
Figura 22. Representación en locales de las esferas del FF	31
Figura 23. Representación en locales de las esferas del HX	32

1 INTRODUCCIÓN

1.1 Motivación

Caminar es una acción que realizamos de forma cotidiana y que consideramos básica. Sin embargo, existen numerosas enfermedades como la parálisis cerebral, esclerosis lateral amiotrófica, enfermedad de Charcot-Marie-Tooth... que impiden andar de forma correcta, empeorando de forma notable la calidad de vida de los enfermos. Una de las enfermedades más estudiadas en este ámbito es la parálisis cerebral, en la que los músculos que intervienen en la marcha están en buen estado, sin embargo, las órdenes cerebrales que controlan el movimiento no son las adecuadas. Andar de forma correcta supone la contracción sucesiva de los músculos que intervienen; en el caso de pacientes con parálisis cerebral, sin embargo, existen músculos que están contraídos durante todo el ciclo de marcha o que se contraen en instantes que no les corresponden impidiendo el desarrollo normal del ciclo de marcha.

El tratamiento para estos pacientes puede requerir de una intervención quirúrgica, siendo una práctica común acortar o alargar determinados músculos de la pierna. El principal atractivo del presente trabajo reside en que pretende generar un modelo directo del pie de fácil aplicación de manera que se establezca una relación directa entre causas y efectos del movimiento. De esta manera se podría predecir qué efecto puede producir un posible cambio en la configuración del pie realizado en una intervención de este tipo.

1.2 Revisión Bibliográfica

En el desarrollo de la marcha humana la parte del cuerpo encargada de interactuar con el entorno es el pie. Este debe ser capaz de absorber los impactos que cíclicamente se producen contra el suelo y además comportarse tanto de forma elástica para adaptarse a superficies irregulares como de forma rígida para actuar como palanca que propulse al cuerpo hacia delante. En determinados momentos de la marcha todo el peso del cuerpo recae en zonas concretas del pie pivotando todo el cuerpo en torno a él.

Esta importancia explica por qué modelar el pie ha sido uno de los objetivos principales en numerosos trabajos científicos destinados a entender el ciclo de marcha humana. A continuación se hace una breve revisión de los más importantes.

Uno de los primeros estudios que se realizan con el objeto de modelar el pie es el que realiza Valiant en 1990 (1). Este estudio se centra en caracterizar el comportamiento viscoelástico del talón en la pisada. El hecho de centrarse en el comportamiento de esta parte del pie en concreto se debe a su importancia en la pisada, ya que se considera que disipa en torno al 80-90% de la energía del impacto que se produce al apoyar el pie. En este estudio, por tanto, se obtienen curvas tensión deformación obtenidas de una experiencia consistente en golpear el talón de una serie de sujetos con un péndulo de 2 kg oscilando a varias velocidades.

Posteriormente Meglan (2) utiliza las curvas tensión-deformación obtenidas por Valiant para modelar una serie de conjuntos muelle-amortiguador de comportamiento no lineal. Aunque el experimento de Valiant se centraba en el comportamiento del talón, Meglan aplica los resultados para modelar toda la planta del pie.

El modelo de pie de Meglan es uno de los primeros en los que se modela el pie en 3 dimensiones y no se le aplican restricciones cinemáticas al mismo. Se prueban dos configuraciones geométricas distintas: la primera de ellas posee 3 esferas deformables que representan el primer y el quinto metatarso y el talón. Esta primera

configuración no considera los dedos. La segunda emplea 7 esferas deformables y representa los 5 dedos como esferas no deformables unidas al resto del pie por articulaciones metatarsófalgicas. Cada una de estas uniones se modela como un muelle helicoidal no-lineal pasivo y un amortiguador en paralelo de características similares. En cuanto a las esferas deformables, ya sean 3 o 7, se consideran unidas a un mismo cuadro rígido.

A partir de un ensayo estático se obtienen las coordenadas x-y de las posiciones de las esferas. La coordenada z se obtiene mediante medida directa del grosor en las localizaciones de las esferas. Para calcular las fuerzas normales al suelo en función de la penetración de las esferas se utiliza una fórmula similar a las que se utilizan en el presente trabajo.

En cuanto a los resultados obtenidos no se aprecian diferencias notables entre las configuraciones de 3 y 7 esferas deformables. Las fuerzas de reacción calculadas no se aproximan a las medidas experimentales. La gráfica de la fuerza normal tiene la forma característica de M pero su magnitud difiere mucho de los valores medidos. Meglan atribuye estos malos resultados a la incertidumbre en la medida del sistema de movimiento usado para obtener la trayectoria del pie.

En la misma línea, Gilchrist (3) crea un modelo formado por 9 conjuntos muelle-amortiguador lineales situados a lo largo de la línea media del pie, 3 de ellos bajo los dedos. Las características pasivas de los muelles son obtenidas cuando sólo contacta uno de ellos y se igualan para el resto de muelles. Los coeficientes de amortiguación se modelan con un polinomio de tercer grado que aproxima a una función escalón de Heaviside.

En la simulación que se realiza se usan muelles para evitar la rotación del pie alrededor del eje anteroposterior. Los resultados que se obtienen son excelentes para estimar la fuerza normal al suelo y bastante buenos para estimar en cada momento la posición del centro de fuerzas.

Cabe destacar que en estos trabajos la marcha humana se trata desde el enfoque de la dinámica directa. Existe la corriente opuesta de usar la dinámica inversa para resolver el problema. En estos trabajos son conocidos tanto el movimiento del cuerpo como las fuerzas de reacción del suelo, siendo el objetivo calcular las fuerzas de reacción de las articulaciones y los momentos netos creados por los músculos. Al ser conocidas las fuerzas de reacción del suelo, no es necesario modelar el contacto entre este y el pie. Por el contrario el análisis dinámico directo ofrece la ventaja de ser predictivo ya que obtiene como resultado el movimiento del cuerpo teniendo como datos las fuerzas generadas por el. Es en el ámbito de este tipo de análisis donde el modelado del contacto pie-suelo se vuelve fundamental.

1.3 Objetivo

El objetivo del presente trabajo es modelar de forma correcta las fuerzas normales de reacción del suelo mediante esferas con características viscoelásticas. Se utiliza para ello el modelo multisegmento propuesto por Carson en 'Kinematic analysis of a multi-segment foot model for research and clinical applications: a repeatability analysis' (4) y para calcular las fuerzas se utiliza el modelo de esferas propuesto por Silva en 'A Biomechanical Multibody Foot Model for Forward Dynamic Analysis' (5) y las fórmulas propuestas por Wojtyra en 'Multibody Simulation Model of Human Walking' (6).

Se obtienen la fuerza normal y los momentos asociados y se comparan con los valores obtenidos experimentalmente. Se obtiene un error ponderado que se introduce como función objetivo en el algoritmo CMA-ES. Usando MATLAB se realizan optimizaciones de los parámetros del modelo que van reduciendo el valor de la función objetivo y por tanto, minimizando el error.

2 CICLO DE MARCHA Y OBTENCIÓN DE DATOS

2.1 El ciclo de marcha humana

La marcha humana se divide en dos fases principales:

- Fase de apoyo, en la que el pie de referencia está en contacto con el suelo. Ocupa aproximadamente el 60% del tiempo de un ciclo.
- Fase de balanceo, en la que el pie de referencia está suspendido en el aire. Le corresponde el 40% del tiempo de un ciclo.

Por tanto, un ciclo de marcha va desde el apoyo en el suelo de un pie hasta el siguiente apoyo del mismo pie. Durante este ciclo el cuerpo da dos pasos. La longitud que se recorre entre que apoya el pie de referencia (inicio del ciclo) y apoya el contrario hace un paso y hasta que vuelve a apoyar el pie de referencia hace otro, completando un ciclo.

En este trabajo el estudio se centra en la fase de apoyo puesto que es en la que se produce el contacto del pie de referencia, en este caso el derecho.

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

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

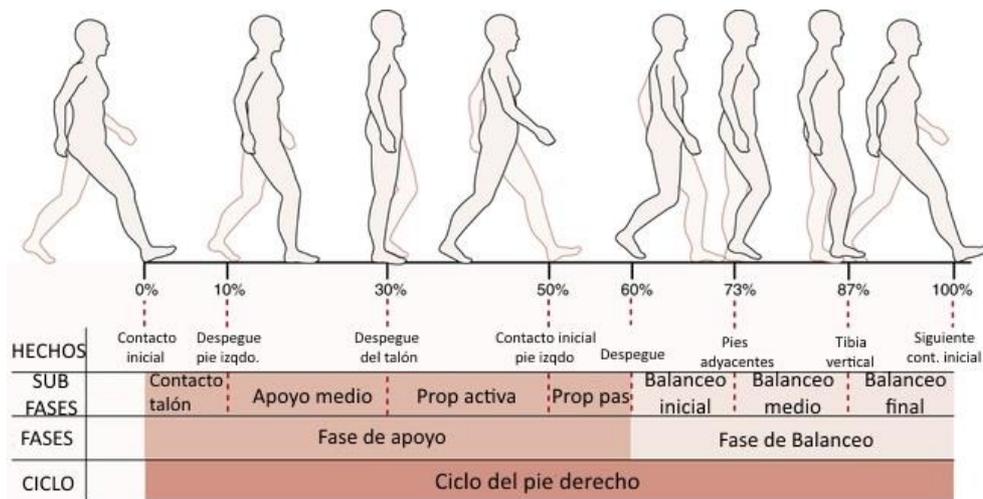


Figura 1. Ciclo de marcha

Si nos fijamos en la figura 1 comprenderemos que durante las fases de apoyo medio y de propulsión activa todo el peso del cuerpo recae solo sobre el pie de referencia, encontrándose el opuesto en la fase de balanceo.

Se intenta que los resultados de este trabajo se adecúen cualitativamente a lo que ocurre durante estas fases. Así, en el contacto de talón participarán mayoritariamente esferas situadas en esta zona, representada en el modelo en el segmento del antepié, produciéndose el despegue de estas esferas más o menos a la mitad de la fase de apoyo coincidiendo con el despegue del talón que marca el inicio de la fase de propulsión activa. Por su parte, las esferas localizadas en la parte delantera del pie comenzarán a penetrar en momentos anteriores al despegue del talón adecuándose a lo que ocurre durante el apoyo medio y serán las únicas que generen fuerza en los últimos instantes del contacto.

2.2 Obtención de datos

La obtención de datos experimentales se ha realizado en el Laboratorio de Ingeniería Mecánica de la US. El equipo de medida de la marcha humana consta de una plataforma de fuerza por la que camina el sujeto y un sistema de captura de imagen VICON compuesto por 6 cámaras.

La plataforma de fuerza registra la fuerza de reacción vertical y sus momentos asociados a lo largo del tiempo. El sistema de captura de imagen es del tipo óptico-pasivo: usa marcadores retrorreflectores que se adhieren a la piel del sujeto en determinados puntos importantes y que son reconocidos por cámaras infrarrojas. De esta manera se captura el movimiento que realiza el cuerpo durante la marcha. Este método es el más común y flexible que se usa en la industria.

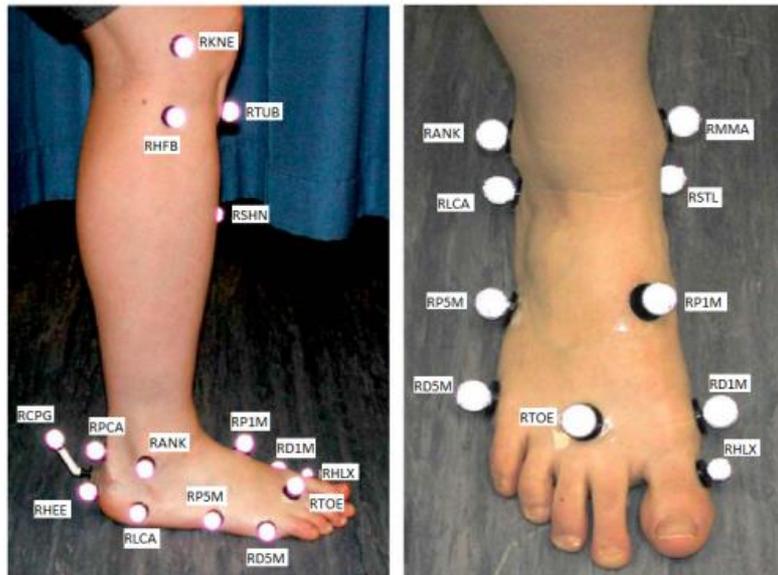


Figura 2. Marcadores retrorreflectores

Los datos obtenidos tanto por la plataforma de fuerza como por el sistema de captura de imagen se procesan en un ordenador mediante un software de la empresa VICON.

El modelo utilizado para el análisis cinemático del pie es el conocido como ‘pie de Oxford’, modelo integrado en el software de VICON que se utiliza en el laboratorio. Este modelo ha sido utilizado por el Nuffield Orthopaedic Centre durante varios años y ha sido validado por numerosos estudios clínicos, entre ellos el utilizado en este trabajo (4).

Este modelo utiliza 43 marcadores en total, localizados en puntos anatómicos específicos del miembro inferior del cuerpo humano.

Siguiendo el tutorial de la empresa VICON, después de colocar los marcadores se introducen algunas medidas del sujeto en el ordenador tales como su altura o peso. A continuación se realiza el ensayo estático, en el que el sujeto se coloca sobre la plataforma de fuerza y se queda quieto. Posteriormente se retiran los marcadores MMA, PCA y DIM y se realiza el ensayo dinámico, en el que el sujeto camina sobre la plataforma.

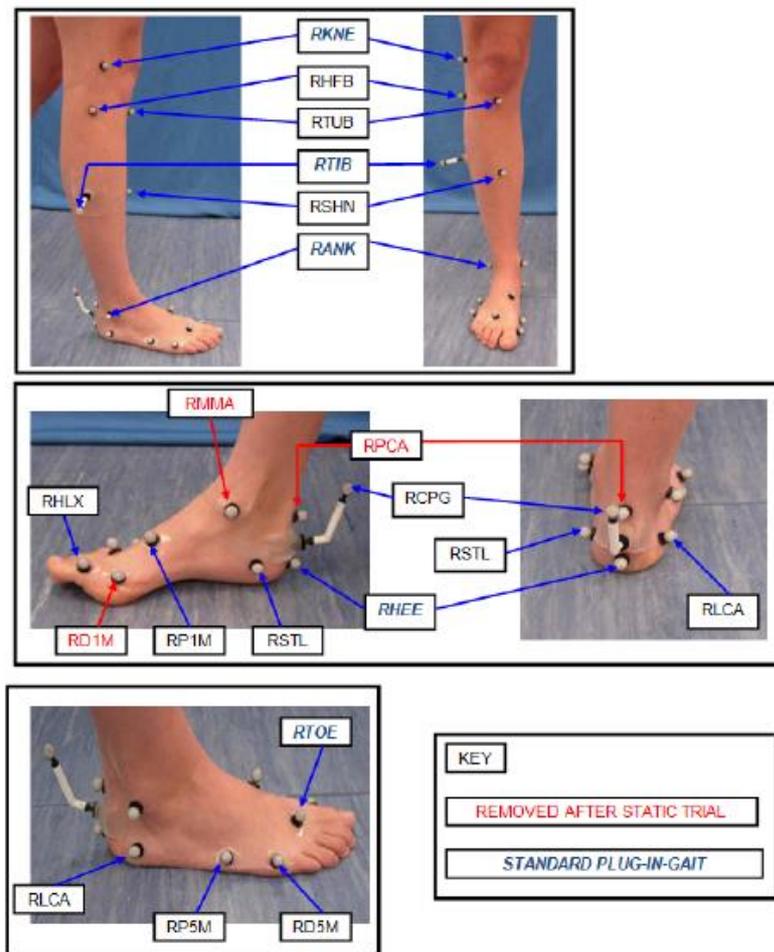


Figura 3. Marcadores del modelo

Se conoce de este modelo que su o repetibilidad es debida principalmente a la variabilidad a la hora de colocar los marcadores. Otra de las mayores dificultades del estudio en vivo reside en el hallux, donde los marcadores se mueven de su posición respecto al pie ya que están adheridos en zonas cercanas a tendones del pie.

Esta fase de obtención de datos no ha sido tratada en el presente trabajo; por el contrario, se ha tratado la información de la posición de los marcadores y los valores de las fuerzas de reacción registrados por la plataforma como datos del problema. La resolución de los equipos es de 100 Hz, por lo que los vectores obtenidos representan valores tomados entre instantes de tiempo de 0.01 segundos. En el procesado de estos datos en MATLAB se han tenido en cuenta solo los instantes de tiempo pertenecientes a la fase de apoyo: de la componente 7 a la 83.

2.3 Ejes Coordinados

Antes de explicar el modelo de pie utilizado y con el objetivo de entender términos y gráficas que se utilizarán posteriormente es muy importante conocer los ejes que utilizaremos.

2.3.1 Ejes anatómicos

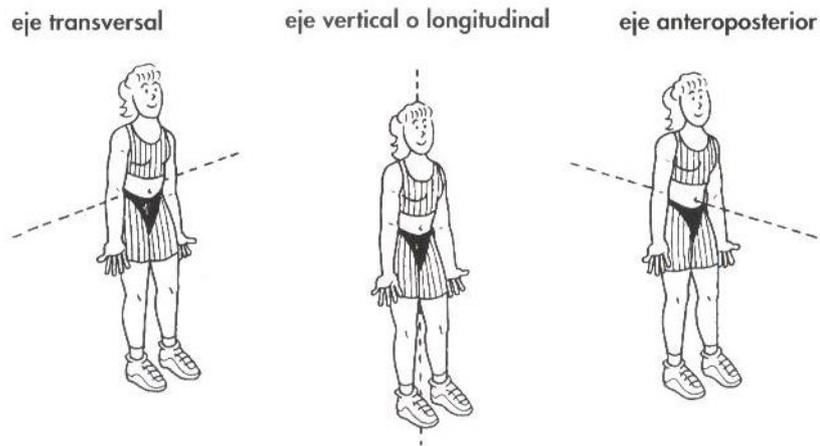


Figura 4. Ejes anatómicos

2.3.2 Ejes globales

En estos ejes se definirán las fuerzas de reacción del suelo: la fuerza normal es positiva en la dirección positiva del eje Z global.

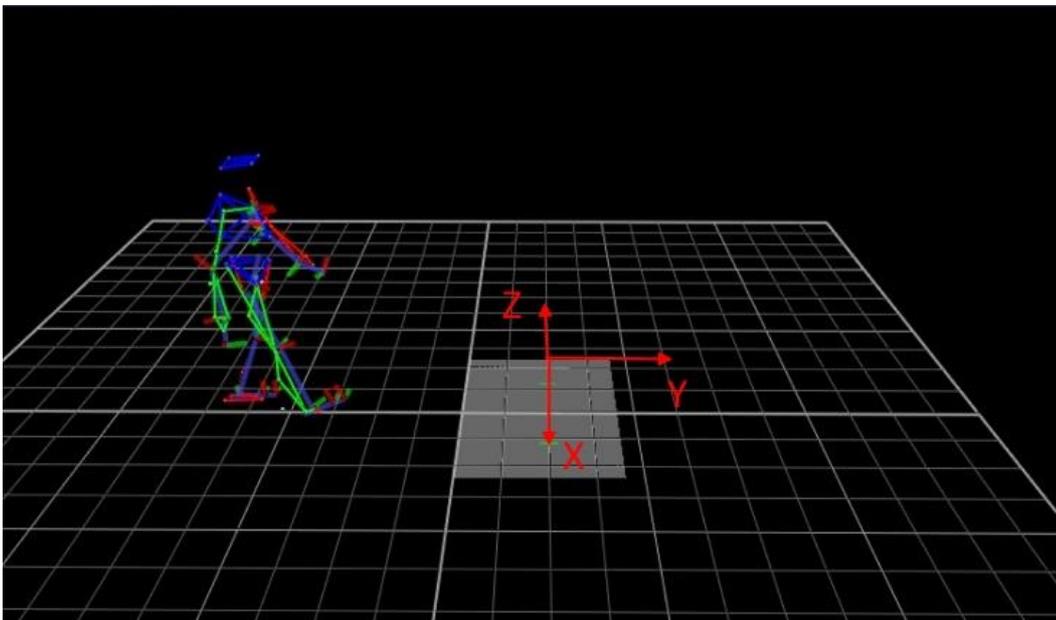


Figura 5. Ejes globales

2.3.3 Ejes locales

Los ejes locales son diferentes para cada segmento, pero todos coinciden aproximadamente con los que se representan a continuación. Al ser diferentes para cada segmento del pie, existirá por tanto un origen diferente para cada segmento y el representado en la figura a continuación no coincide con ninguno de ellos, sirviendo por tanto la figura como una mera aclaración.

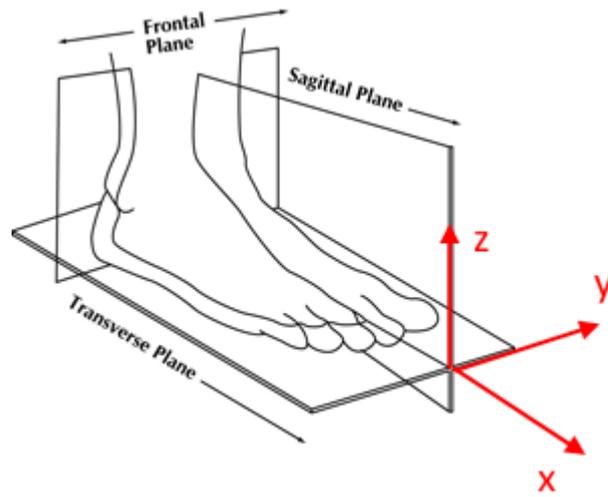


Figura 6. Ejes locales

Respecto al plano sagital representado en la figura anterior se definen los términos lateral y medial según la distancia a este plano sea mayor o menor, respectivamente. Al referirnos a la parte más lateral del pie nos referimos por tanto al borde exterior del mismo.

3 MODELO DEL PIE

3.1 Modelo cinemático

El modelo cinemático utilizado es el propuesto por Carson en 'Kinematic analysis of a multi-segment foot model for research and clinical applications: a repeatability analysis' (4). En este trabajo se simplifica la compleja estructura anatómica del pie consistente en 26 huesos (28 si contamos la tibia y el peroné incluidos en este modelo) usando un modelo que divide al pie en tres segmentos (retropié, antepié y dedo gordo) y además considera un segmento tibial con características de sólido rígido. El interés en la parte media del pie se centra tan solo en su papel como transmisora del movimiento entre retropié y antepié. El movimiento relativo se describe sin restricciones entre segmentos, es decir, con seis posibles grados de libertad entre cualquier par de segmentos.

Los 26 huesos del pie anteriormente citados se pueden agrupar en 3 grupos:

- Tarso: formado por 7 huesos, de atrás adelante el calcáneo, el astrágalo, el escafoides o navicular, el cuboides y tres cuñas (primera o medial, segunda o intermedia y tercera o lateral). La tibia y el peroné se conectan con la parte superior del astrágalo para formar el tobillo.
- Metatarso: con 5 huesos largos, los metatarsianos.
- Falanges: con 14 huesos. El dedo gordo posee 2, mientras que el resto poseen 3 cada uno: primera o proximal, segunda o media y tercera, distal o ungueal.

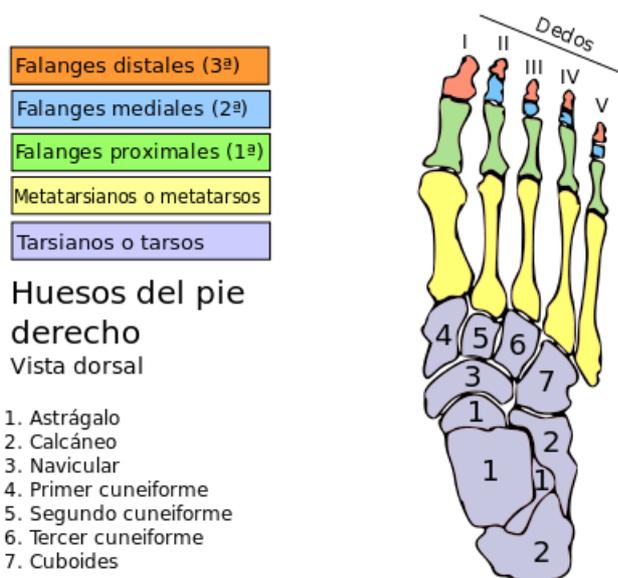


Figura 7. Huesos del pie

En este modelo, el segmento tibial (TB) lo forman la tibia y el peroné.

El calcáneo y el astrágalo forman el retropié, denominado en el trabajo de Carson Hindfoot (HF).

Por su parte, el cuboides, el navicular y las tres cuñas forman la parte media del pie, que no se tiene ningún

segmento asociado, restringiéndose su acción a transmitir el movimiento entre los segmentos del retropié y el antepié.

Finalmente, el antepié lo formarían los cinco metatarsianos y las falanges, sin embargo, en el trabajo de Carson este segmento, denominado Forefoot (FF) solo incluye los cinco metatarsianos dejando la falange proximal del dedo gordo como única componente del último segmento, el Hallux (HX), despreciando el resto de falanges.

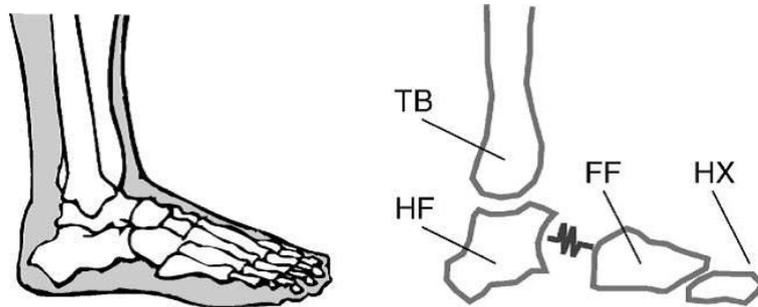


Figura 8. Pie de Oxford

En el siguiente gráfico se muestra la situación de los marcadores RHEE, RP1M y RHLX que corresponden a los orígenes de coordenadas locales del Hindfoot, Forefoot y Hallux, respectivamente.

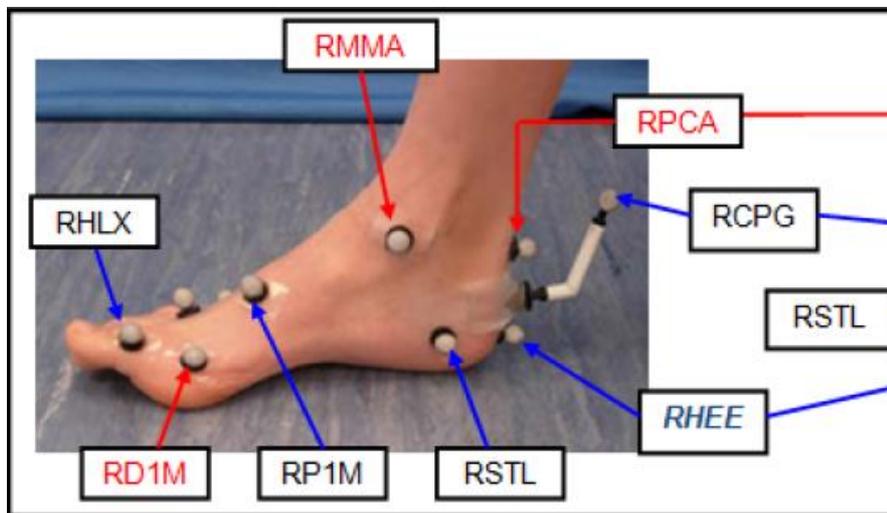


Figura 9. Marcadores del pie

3.2 Cálculo de las fuerzas de contacto

Para el cálculo de las fuerzas de contacto se situarán unas esferas en cada segmento y según su pseudo-penetración en el suelo se calcularán las fuerzas siguiendo la ecuación propuesta por Wojtyra. Para calcular la condición de penetración de cada esfera con el suelo se usan las fórmulas propuestas por Silva (5):

$$d = r_j^P - r_i^C \quad (3-1)$$

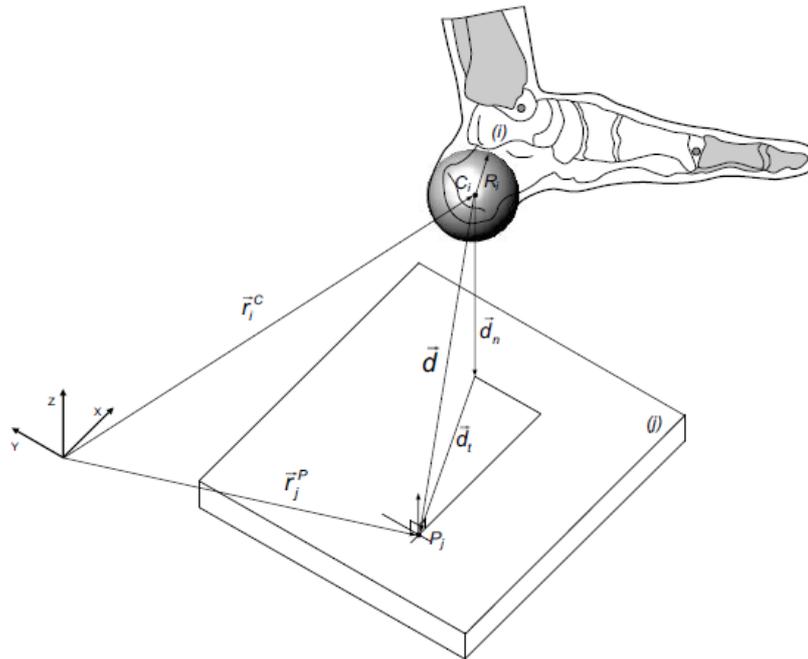


Figura 10. Cálculo de la penetración

Siendo P un punto cualquiera situado en el suelo y C el centro de la esfera considerada. Los vectores de la ecuación anterior están expresados en coordenadas globales y en el sistema de referencia inercial. La componente vertical del vector d se calcula:

$$d_n = d^T n \quad (3-2)$$

Siendo n un vector perpendicular al suelo orientado hacia arriba y de módulo unidad. Finalmente, la penetración se define como:

$$\delta = R_i - d_n \quad (3-3)$$

Wojtyra (6) calcula la pseudo-penetración como:

$$p_z = \text{MAX}(0, \delta) \quad (3-4)$$

Y las fuerzas calculadas en función de la penetración quedan:

$$F_z = \begin{cases} \text{MAX}(0, k\delta + c\dot{\delta}) & \delta > 0 \\ 0 & \delta \leq 0 \end{cases} \quad (3-5)$$

Siendo k el coeficiente de rigidez de la esfera, c el coeficiente de amortiguamiento. Estos dos coeficientes, distintos para cada esfera, definen el comportamiento viscoelástico de las mismas.

Normalmente, las esferas penetran en el suelo con velocidad distinta de cero, por tanto, para evitar discontinuidades en la fuerza de amortiguamiento el coeficiente de amortiguamiento c debe ser nulo al comienzo del contacto. Por ello Wojtyra lo define como una función lineal de la pseudo-penetración:

$$c = \begin{cases} c_{max} \left(\frac{3}{h^2} p_z^2 - \frac{2}{h^3} p_z^3 \right) & p_z < h \\ c_{max} & p_z \geq h \end{cases} \quad (3-6)$$

Donde h y c_{max} son valores constantes:

$$h = 0.01 \text{ m} \quad (3-7)$$

$$c_{max} = 1500 \frac{Ns}{m} \quad (3-8)$$

Estos valores se obtienen del trabajo de Wojtyra y en ningún caso se han modificado. En él se explica que se realizan varias simulaciones para encontrar los valores adecuados de estas constantes. La pseudopenetración corresponde a la deflexión del pie y el zapato durante la marcha. El valor de c_{max} se elige para prevenir que el pie rebote tras contactar con el suelo. Sin embargo, la elección de estos valores no es crucial, explica, ya que tras un cambio del 20% de cada valor el comportamiento del modelo permanece casi inalterado.

4 EL ALGORITMO EVOLUTIVO CMA-ES

Este algoritmo es un método estocástico para optimización de parámetros reales (de dominio continuo) de funciones objetivo no lineales y no convexas.

El objetivo consiste en minimizar una función objetivo de la que no hay mas información disponible que los valores de los puntos que se evalúan durante la búsqueda. Estos algoritmos se conocen como ‘de caja negra’, en inglés ‘black box’. No se dispone, por tanto, del gradiente de la función, de gran utilidad para la búsqueda del mínimo. Por tanto, este algoritmo se limita a elegir un número λ de puntos independientes a partir de una distribución, evaluarlos, y a partir de esa información, actualizar los parámetros de la distribución siguiente para mejorar la búsqueda de forma iterativa hasta llegar a algunos de los criterios de parada.

En el caso del CMA-ES, la distribución que se utiliza es la distribución normal, bien conocida en su versión unidimensional pero que aquí se amplía a múltiples variables. Se escoge esta distribución por ser la de mayor entropía y por no favorecer en ningún caso una dirección coordinada del espacio en particular.

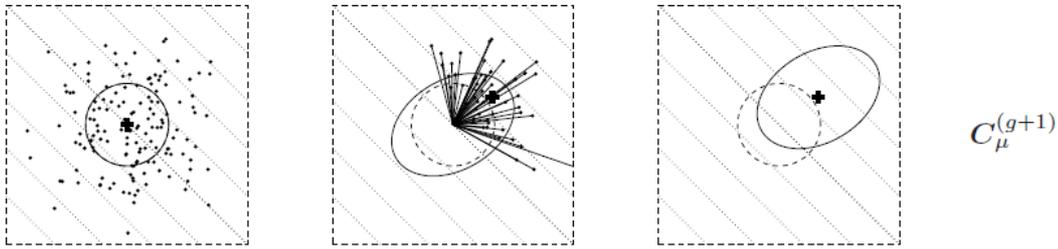


Figura 11. Algoritmo CMA-ES: generación de candidatos, estimación y nueva generación

Siguiendo una distribución normal, la ecuación básica para generar los λ candidatos anteriormente mencionados para una determinada generación es:

$$x_k^{g+1} \sim m^g + \sigma^g N(0, C^g) \text{ para } k = 1, \dots, \lambda \quad (4-1)$$

Donde

\sim denota misma distribución en lado izquierdo y derecho

$N(0, C^g)$ es una distribución normal de n variables con media cero y matriz de covarianza C^g . Se tiene que:

$$m^g + \sigma^g N(0, C^g) \sim N(m^g, (\sigma^g)^2 C^g) \quad (4-2)$$

$x_k^{g+1} \in \mathbb{R}^n$, k -ésimo descendiente, individuo o punto de búsqueda de la generación $g+1$

$m^g \in \mathbb{R}^n$, valor medio de la distribución de búsqueda de la generación g

$\sigma^g \in \mathbb{R}_{>0}$, desviación estándar o tamaño de paso de la generación g

$C^g \in \mathbb{R}^{n \times n}$, matriz de covarianza de la generación g .

$\lambda \geq 2$, tamaño de población, número de descendientes

Al conjunto de estos candidatos para una determinada generación se le conoce como población.

El siguiente paso después de generar estos candidatos es evaluarlos y ordenarlos según su valor de la función objetivo. Posteriormente se seleccionan los μ mejores individuos para en base a ellos optimizar los parámetros de búsqueda, obteniendo m^{g+1} , σ^{g+1} , C^{g+1} . A continuación, en cada uno de los siguientes apartados se detallan las fórmulas para actualizar cada uno de estos parámetros.

4.1 Actualización del centro de la distribución

Se realiza según la siguiente formula:

$$m^{g+1} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{g+1} \quad (4-3)$$

$$\sum_{i=1}^{\mu} w_i = 1, \quad w_1 \geq w_2 \geq \dots \geq w_{\mu} > 0 \quad (4-4)$$

Donde

$\mu \leq \lambda$, es el tamaño de población de progenitores

$w_{i=1\dots\mu} \in \mathbb{R}_{>0}$, pesos de recombinación positivos

$x_{i:\lambda}^{g+1}$, i -ésimo mejor individuo de $x_1^{g+1}, \dots, x_{\lambda}^{g+1}$. El índice $i:\lambda$ denota el índice del i -ésimo mejor individuo y $f(x_{1:\lambda}^{g+1}) \leq f(x_{2:\lambda}^{g+1}) \leq \dots \leq f(x_{\lambda:\lambda}^{g+1})$

Escribiendo la ecuación como una actualización de m :

$$m^{g+1} = m^g + c_m \sum_{i=1}^{\mu} w_i (x_{i:\lambda}^{g+1} - m^g) \quad (4-5)$$

Donde

$c_m \leq 1$ es un coeficiente de aprendizaje, normalmente 1.

4.2 Actualización de la matriz de covarianza

En un principio se supone que la población contiene la información suficiente para estimar una matriz de covarianza:

$$C_{\mu}^{g+1} = \sum_{i=1}^{\mu} w_i (x_{i:\lambda}^{g+1} - m^g)(x_{i:\lambda}^{g+1} - m^g)^T \quad (4-6)$$

Sin embargo, si la población no es lo suficientemente grande entonces no es posible estimar la matriz de covarianza usando la ecuación anterior, por lo que se usa una media de las matrices de covarianza de las generaciones anteriores:

$$C^{g+1} = \frac{1}{g+1} \sum_{i=1}^{\mu} \frac{1}{(\sigma^i)^2} C_{\mu}^{i+1} \quad (4-7)$$

También es necesario guardar información del signo de los pasos que se han ido tomando, por lo que se construye un sendero de evolución:

$$p_c^{g+1} = (1 - c_c) p_c^g + \sqrt{c_c(2 - c_c) \mu_{eff}} \frac{m^{g+1} - m^g}{\sigma^g} \quad (4-8)$$

Donde

$$\begin{aligned} p_c^g &\in \mathbb{R}^n \\ c_c &\leq 1 \\ \mu_{eff} &= \left(\frac{\|w\|_1}{\|w\|_2} \right)^2 = \frac{\|w\|_1^2}{\|w\|_2^2} = \frac{(\sum_{i=1}^{\mu} |w_i|)^2}{\sum_{i=1}^{\mu} w_i^2} = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \end{aligned} \quad (4-9)$$

Teniendo en cuenta estas dos últimas consideraciones, finalmente la matriz de covarianza se actualiza siguiendo esta fórmula:

$$C^{g+1} = \left(1 - c_1 - c_{\mu} \sum w_j\right) C^g + c_1 p_c^{g+1} p_c^{(g+1)T} + c_{\mu} \sum_{i=1}^{\lambda} w_i y_{i,\lambda}^{g+1} (y_{i,\lambda}^{g+1})^T \quad (4-10)$$

Donde

$$\begin{aligned} c_1 &\approx 2/n^2 \\ c_{\mu} &\approx \min\left(\frac{\mu_{eff}}{n^2}, 1 - c_1\right) \\ y_{i,\lambda}^{g+1} &= (x_{i,\lambda}^{g+1} - m^g) / \sigma^g \\ \sum w_j &= \sum_{i=1}^{\lambda} w_i \approx -c_1 / c_{\mu} \end{aligned}$$

4.3 Actualización del tamaño de paso

Por último, el tamaño de paso se actualiza de esta manera:

$$\sigma^{g+1} = \sigma^g \exp\left(\frac{c_{\sigma}}{d_{\sigma}} \left(\frac{\|p_{\sigma}^{g+1}\|}{E \|N(0, I)\|} - 1\right)\right) \quad (4-11)$$

El uso del algoritmo CMA-ES para este trabajo está justificado por tratarse de un problema no lineal, complejo y con un número de variables a optimizar alto, por lo que generaría un costo operativo prohibitivo usar un método tradicional de búsqueda.

5 PROCESO DE OPTIMIZACIÓN

El trabajo en Matlab ha consistido básicamente en programar las ecuaciones de Silva para el modelo multisegmento y elaborar la rutina postprocesado para evaluar los resultados de cada optimización.

Se ha partido de un código proporcionado por el Departamento que implementa el modelo de esferas para un modelo monosegmento del pie. Este código se modifica para implementar el modelo de esferas (ecuaciones de Silva) para el modelo multisegmento y da lugar a la rutina GRF_Oxford_normal_esferas_mod.

Esta rutina se mejoró posteriormente reduciendo el numero de bucles y prelocalizando todas las variables para mejorar su velocidad. Basta pensar en que en cada evaluación de la función f que realiza la subrutina cmaes se llama a GRF_Oxford_normal_mod y el numero de evaluaciones puede llegar a superar las 100000.

Desde la página web de Matlab también se recomienda el no usar variables globales, sin embargo esta mejora no ha podido ser llevada a cabo ya que al existir muchas funciones anidadas que usan las mismas variables habría que pasar muchos argumentos a cada una de esas funciones.

El proceso de optimización se ha llevado a cabo para encontrar los mejores valores de los siguientes parámetros del modelo:

- Posicion de las esferas (en coordenadas x,y,z locales)
- Radios de las esferas
- Valores de c y k para cada esfera

Por tanto el numero de variables a optimizar es $6n$, siendo n el numero de esferas.

Esta optimización es realizada por la subrutina cmaes (versión 3.61.beta, último cambio en Abril del 2012) descargada de la página oficial del algoritmo CMA-ES. Esta subrutina recibe como argumentos de entrada:

- Vector inicial x_0 de parámetros a optimizar
- Limites superior e inferior de los valores de los parámetros

Como argumento de salida se tiene el valor optimo de la función objetivo f que se calcula como:

$$f = \sum_{i=1}^3 NRMSE_i \quad (5-1)$$

Donde

$$NRMSE_i = \frac{RMSE_i}{F_i^{MAX} - F_i^{MIN}} \quad (5-2)$$

Donde

$$RMSE_i = \sqrt{\frac{1}{N} \sum_{k=1}^N (F_i(k)^{EXP} - F_i(k)^{CALC})^2} \quad (5-3)$$

Donde

f es la función objetivo

N es la longitud de los vectores que representan la fuerza normal, el momento en x, y el momento en y para los distintos instantes de tiempo

$F_i(k)^{EXP}$ son los vectores que representan la fuerza normal, el momento en x y el momento en y, para $i = 1, i = 2, i = 3$, respectivamente en los distintos instantes de tiempo k y que se han obtenido experimentalmente.

$F_i(k)^{CALC}$ son los vectores que representan la fuerza normal, el momento en x y el momento en y, para $i = 1, i = 2, i = 3$, respectivamente en los distintos instantes de tiempo k y que se han calculado mediante la subrutina GRF_Oxford_normal_mod.m

F_i^{MAX} y F_i^{MIN} son los valores máximo y mínimo, respectivamente de los vectores $F_i(k)^{CALC}$

$\sum_{i=1}^3 NRMSE_i$ es la suma de los $NRMSE$ correspondientes a la fuerza normal, el momento en x y el momento en y

La llamada a la subrutina cmaes se realiza desde el código búsqueda_minimo. En dicho código se especifican los argumentos de entrada a esta subrutina.

El proceso de optimización comienza utilizando 3 esferas, una para cada segmento del pie. El vector inicial se generó colocando de manera aproximada las esferas en la línea media del pie. Las coordenadas x e y de los centros de las esferas se obtuvieron calculando de manera aproximada dibujando estas encima de una plantilla del pie y estimando las distancias respecto a los orígenes de cada segmento.

La coordenada z (recordemos que trabajamos ahora en coordenadas locales) y el radio se obtuvieron de manera similar utilizando las medidas de un pie de tamaño normal y procurando que las esferas aproximasen la forma de dicho pie.

Los valores de las características viscoelásticas se mantuvieron respecto al código proporcionado en un primer momento por el departamento.

Los límites inicialmente se establecieron alrededor de los valores estimados anteriormente.

Una vez conseguidos el vector inicial y los límites se puede comenzar a utilizar el código búsqueda_minimo.

El proceso de optimización consiste en que cada vez que se ejecuta búsqueda_minimo van cambiando los valores de los mencionados argumentos de entrada:

- El valor de x_0 se va actualizando con el resultado de la anterior optimización en el caso de que se haya producido una mejora en el valor de $f(x_0)$
- Los valores para los límites superior e inferior se van cambiando también de forma manual tras ejecutar la rutina postprocesado, que analiza el resultado de cada optimización. Dicha rutina genera una serie de figuras y una tabla con los parámetros de cada esfera.

Las figuras más importantes que se utilizan para actualizar, como se ha mencionado, los valores de los límites superior e inferior son las correspondientes a la representación de la fuerza que genera cada esfera por un lado, y por otro a una animación 3D donde se representan las esferas durante toda la fase de apoyo.

Gracias a estas dos representaciones se ajustan los valores de los límites superior e inferior (de forma manual). Se busca que la localización de las esferas tenga sentido físico, es decir, que se encuentren dentro de las dimensiones normales de un pie. Se busca asimismo que no haya ninguna esfera que tenga un tamaño muy superior a las de su correspondiente segmento. Esto se hace para que la contribución de cada esfera sea parecida y se produzca de forma gradual, buscando que las esferas penetren sucesivamente desde el talón hasta los dedos.

En otros casos algunas esferas tienen un tamaño demasiado pequeño y no llegan a penetrar. En la actualización

de los valores de los límites se tiene en cuenta este hecho modificando los valores para intentar que la determinada esfera penetre en el suelo.

Procediendo de esta manera se realizaron varias simulaciones usando 3 esferas, llegando a la conclusión de que los resultados obtenidos no eran aceptables y que era necesario añadir más esferas. A continuación se presenta el mejor resultado obtenido para el modelo de 3 esferas.

	X (cm)	Y (cm)	Z (cm)	Radio (cm)	K ($\frac{N}{m^*} \cdot 10^{-4}$)	C ($\frac{Ns}{m^*} \cdot 10^{-3}$)
Esfera 1	2.72	-1.48	-2.00	1.86	19.38	2.00
Esfera 2	2.77	-0.48	-5.00	1.00	13.43	2.00
Esfera 3	-1.27	-1.00	-1.50	1.10	6.61	2.00

Tabla 1. Mejor resultado con 3 esferas

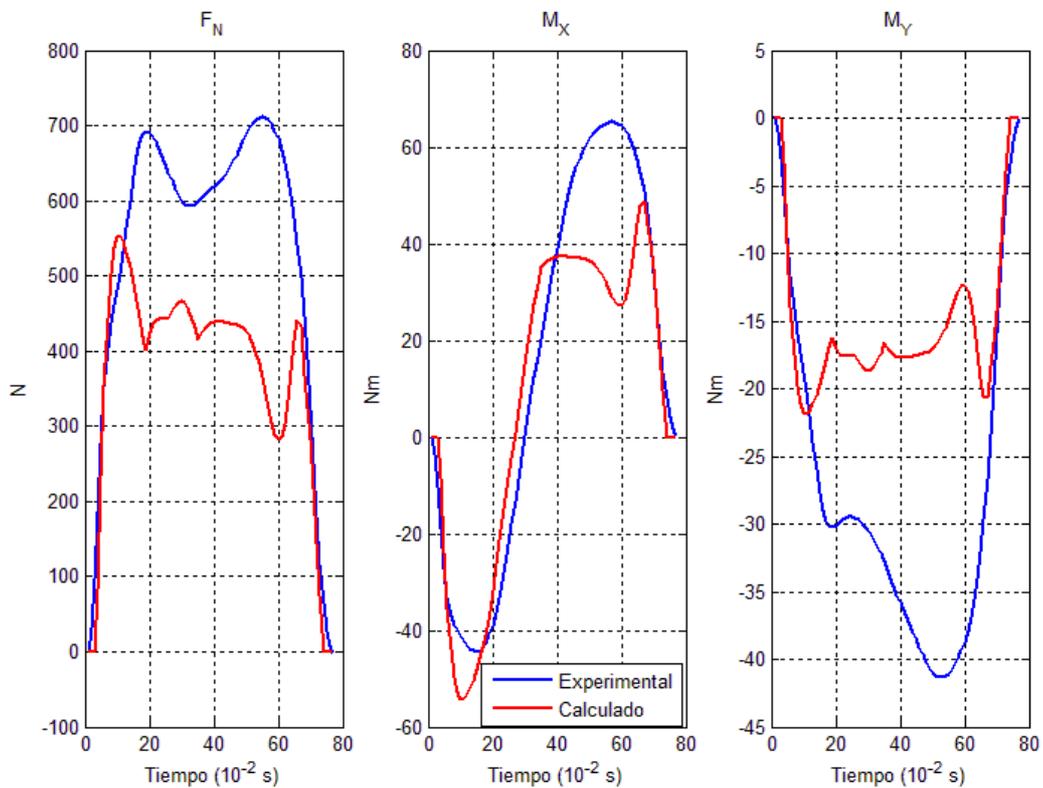


Figura 12. Fuerzas modelo 3 esferas

	NRMSE fuerza normal	NRSME momento X	NRSME momento Y	FUNCIÓN OBJ
3 ESFERAS	2.52	1.36	3.27	7.16

Tabla 2. Errores del modelo de 3 esferas

Pasamos ahora al modelo de 6 esferas. En este caso los límites iniciales son proporcionados por el departamento y el vector inicial se genera a dentro de esos límites.

	X (cm)	Y (cm)	Z (cm)	Radio (cm)	K ($\frac{N}{m} * 10^{-4}$)	C ($\frac{Ns}{m} * 10^{-3}$)
Esfera 1	3	0	-3	3	100	100
Esfera 2	5	-1	-2	4	100	100
Esfera 3	0	-2	-4	4	50	50
Esfera 4	5	0	-3	5	50	50
Esfera 5	-1	-5	-2	1	100	100
Esfera 6	2	-4	-1	2	100	100

Tabla 3. Vector inicial del modelo de 6 esferas

	X (cm)	Y (cm)	Z (cm)	Radio (cm)	K ($\frac{N}{m} * 10^{-4}$)	C ($\frac{Ns}{m} * 10^{-3}$)
Esfera 1	0÷7	-1÷1	-4÷-1	1.5÷5	2÷200	2÷200
Esfera 2	1÷7	-3÷2	-4÷-1	1.5÷5	2÷200	2÷200
Esfera 3	-5÷8	-6÷1	-6÷-2	3÷6	2÷100	2÷100
Esfera 4	-5÷8	-6÷1	-6÷-2	3÷6	2÷100	2÷100
Esfera 5	-3÷2	-6÷-2	-3÷-1.5	1÷2.5	2÷200	2÷200
Esfera 6	-3÷3	-7÷-2	-3÷-1.5	1÷2.5	2÷200	2÷200

Tabla 4. Límites iniciales del modelo de 6 esferas

Se procede de la misma manera que en el caso de 3 esferas, cambiando los valores de los argumentos de entrada, obteniéndose tras muchas simulaciones la mejor solución para este caso:

	X (cm)	Y (cm)	Z (cm)	Radio (cm)	K ($\frac{N}{m} * 10^{-4}$)	C ($\frac{Ns}{m} * 10^{-3}$)
Esfera 1	2.65	-0.65	-2.53	1.50	12.81	2.01
Esfera 2	4.78	-2.31	-2.20	1.51	17.80	2.01
Esfera 3	3.21	1.00	-2.70	3.00	60.49	14.51
Esfera 4	2.50	-0.93	-2.86	3.00	21.98	2.00
Esfera 5	-3.00	-2.85	-2.41	1.00	2.08	2.00
Esfera 6	2.52	-4.87	-1.50	1.30	2.00	45.32

Tabla 5. Mejor resultado con 6 esferas

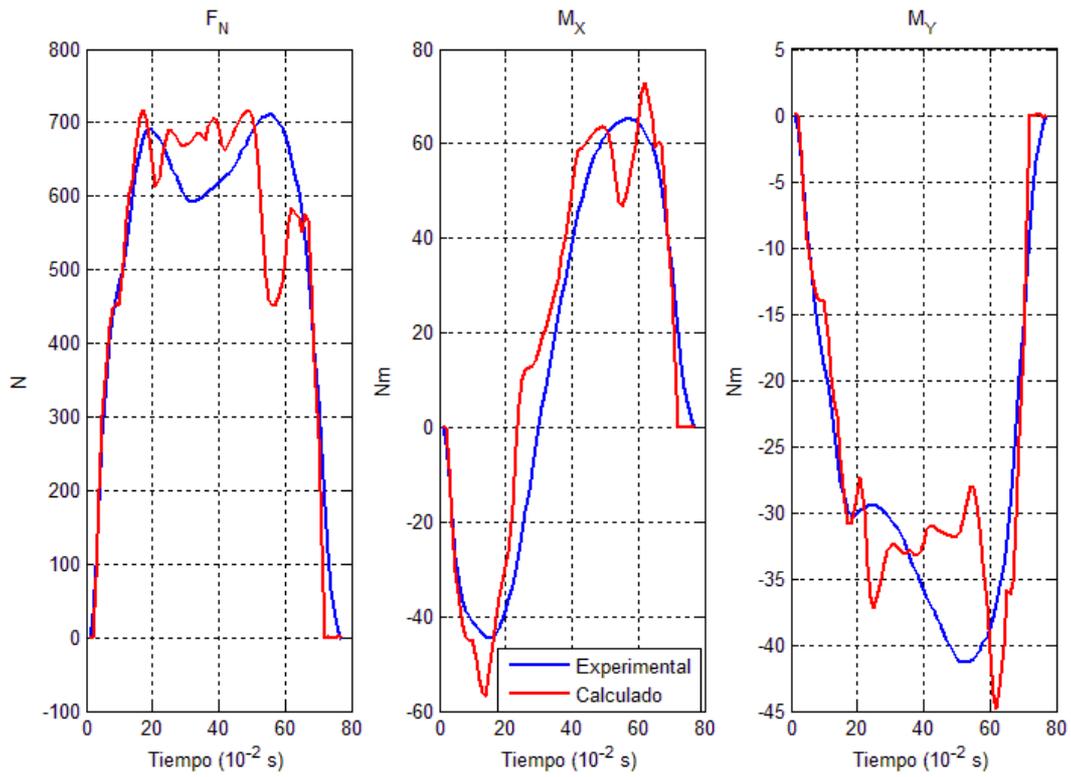


Figura 13. Fuerzas modelo 6 esferas

	NRMSE fuerza normal	NRSME momento X	NRSME momento Y	FUNCIÓN OBJ
3 ESFERAS	2.52	1.36	3.27	7.16
6 ESFERAS	1.10	0.95	1.18	3.24

Tabla 6. Comparación entre los modelos de 3 y 6 esferas

Se observa que el NRMSE se ha mejorado en todos los casos al aumentar el número de esferas.

Pasamos por último al modelo de 9 esferas. El vector inicial se genera a partir del mejor resultado del caso de 6 esferas, añadiendo una más para cada segmento. Se utilizan en principio los límites con los que se obtuvo el mejor resultado en el caso de 6 esferas.

	X (cm)	Y (cm)	Z (cm)	Radio (cm)	K ($\frac{N}{m} * 10^{-4}$)	C ($\frac{Ns}{m} * 10^{-3}$)
Esfera 1	2.65	-0.65	-2.53	1.50	12.81	2.01
Esfera 2	4.78	-2.31	-2.20	1.51	17.80	2.01
Esfera 3	8.00	-1.00	-2.20	1.50	15.00	2.00
Esfera 4	3.21	1.00	-2.70	3.00	60.49	14.51
Esfera 5	2.50	-0.93	-2.86	3.00	21.98	2.00
Esfera 6	-4.00	-0.80	-2.70	2.00	30.00	2.00
Esfera 7	-3.00	-2.85	-2.41	1.00	2.08	2.00
Esfera 8	2.52	-4.87	-1.50	1.30	2.00	45.32
Esfera 9	0.00	-3.00	-1.50	1.00	2.00	2.00

Tabla 7. Vector inicial del modelo de 9 esferas

Es en este modelo donde se invirtió más tiempo y se introdujeron nuevas estrategias de optimización además de la ya mencionada tras realizar numerosas simulaciones.

La primera de ellas consistió en cambiar los valores de los límites de los parámetros viscoelásticos, más ajustados a la literatura y se muestran a continuación.

	K ($\frac{N}{m} * 10^{-4}$)	C ($\frac{Ns}{m} * 10^{-3}$)
Esfera 1	1÷20	0.1÷5
Esfera 2	1÷20	0.1÷5
Esfera 3	1÷20	0.1÷5
Esfera 4	1÷10	0.1÷5
Esfera 5	1÷10	0.1÷5
Esfera 6	1÷10	0.1÷5
Esfera 7	1÷20	0.1÷5
Esfera 8	1÷20	0.1÷5
Esfera 9	1÷20	0.1÷5

Tabla 8. Nuevos límites para k y c

La segunda consistió en cambiar ciertos parámetros del algoritmo CMA-ES: el número de descendientes y el de progenitores. El código cmaes establece por defecto unos valores:

$$\lambda = \text{abs}(4 + 3 \log(N)) \quad (5-4)$$

Donde

abs redondea el número al entero más próximo menor o igual que el (comando floor de MATLAB)

log denota logaritmo neperiano (notación de MATLAB)

N es el número de parámetros a optimizar

$$\mu = \text{abs}\left(\frac{\lambda}{2}\right) \quad (5-5)$$

Para el caso de 9 esferas en el que nos encontramos, $N = 54$, $\lambda = 15$, $\mu = 7$. Los creadores del algoritmo recomiendan realizar optimizaciones aumentando el tamaño de población para realizar una búsqueda más robusta. En este sentido, se realizaron pruebas con un tamaño de población igual a 50, proporcionando mejores resultados en todos los casos respecto a las mismas búsquedas con $\lambda = 15$. Con este valor predeterminado de λ el algoritmo convergía prematuramente a valores de la función objetivo superiores (y por tanto peores) a los que posteriormente se obtenían con $\lambda = 50$. En cuanto al número de progenitores, los creadores del algoritmo recomiendan aumentarlo de forma proporcional al tamaño de población, sin embargo, se mantuvo $\mu = 7$ después de leer el trabajo (7), en el que los mejores resultados se obtienen usando $\lambda = 100$, $\mu = 7$, valores no típicos de este algoritmo. Se decidió también no aumentar λ a un número superior a 50 para no aumentar un tiempo de optimización ya largo de por sí.

La tercera estrategia implicó un mayor esfuerzo a la hora de comprender cuales son las zonas de la planta del pie con mas importancia a la hora de generar las fuerzas normales de la pisada. Por ello, tras leer el trabajo (8), se empezó a situar una sola esfera correspondiente al HF centrada en el plano sagital para modelar el talón y las otras dos esferas correspondientes a este segmento en una posición más lateral. Las tres esferas del MF se situaron en la zona del pie justo antes de los dedos, en la que se registran presiones altas en el trabajo citado anteriormente. Por último, las 3 esferas correspondientes al HX se situaron en una posición más medial, correspondiente a la situación del dedo gordo del pie.

Combinando todas estas estrategias se obtiene el mejor resultado conseguido, el cual se analiza en el siguiente capítulo.

En cuanto a los criterios de parada, se establecieron dos: numero máximo de evaluaciones de la función objetivo 1000000 y tolerancia de la función 0.01. Sin embargo, en la mayoría de ocasiones la parada ha sido manual a través del fichero signals.par. En ningún caso se ha llegado al numero máximo de evaluaciones 1000000 pues se ha parado manualmente antes.

6 RESULTADO Y CONCLUSIONES

Se presenta a continuación el mejor resultado obtenido, correspondiente a un modelo de 9 esferas, de las que solo penetran 7 de ellas. El valor de la función objetivo correspondiente a este resultado es $f = 2.52$

	X (cm)	Y (cm)	Z (cm)	Radio (cm)	$K \left(\frac{N}{m} * 10^{-4} \right)$	$C \left(\frac{Ns}{m} * 10^{-3} \right)$
Esfera 1	2.46	-1.00	-2.94	1.00	160.95	100.00
Esfera 2	6.00	-3.34	-2.65	1.00	53.75	100.00
Esfera 3	15.00	-4.76	-2.70	1.02	47.20	100.00
Esfera 4	4.97	-0.55	-5.20	1.12	34.30	343.30
Esfera 5	2.73	-3.00	-5.00	1.10	10.00	100.00
Esfera 6	3.00	-7.00	-4.90	1.56	10.00	1985.70
Esfera 7	1.21	-1.29	-1.64	0.90	111.65	1751.00
Esfera 8	-1.05	-4.30	2.00	0.99	95.00	100.00
Esfera 9	-1.44	-7.53	0.73	0.70	90.137	1907.60

Tabla 9. Mejor resultado con 9 esferas

Las esferas 8 y 9 no penetran y por tanto no generan contribución alguna al cálculo de fuerzas. El motivo por el que se tienen en cuenta es por una cuestión de simplicidad al programar en MATLAB. Desde el principio se decidió usar el mismo número de esferas para cada uno de los segmentos: primero 3, luego 6 y finalmente 9.

Los límites usados para llegar a esta solución son:

	X (cm)	Y (cm)	Z (cm)	Radio (cm)	K ($\frac{N}{m^*}$ 10^{-4})	C ($\frac{Ns}{m^*}$ 10^{-3})
Esfera 1	2 ÷ 4	-1 ÷ 1	-4 ÷ -2.5	1 ÷ 3	1 ÷ 20	0.1 ÷ 5
Esfera 2	6 ÷ 10	-4 ÷	-4 ÷ -2.5	1 ÷ 3	1 ÷ 20	0.1 ÷ 5
Esfera 3	10 ÷ 15	-6 ÷ -2	-3.4 ÷ -2.7	1 ÷ 1.3	1 ÷ 20	0.1 ÷ 5
Esfera 4	3 ÷ 5	-3 ÷ 0	-5.2 ÷ -4.5	1 ÷ 1.6	1 ÷ 10	0.1 ÷ 5
Esfera 5	2 ÷ 4	-5 ÷ -3	-5 ÷ -4.5	1 ÷ 1.7	1 ÷ 10	0.1 ÷ 5
Esfera 6	1.5 ÷ 3	-7 ÷ -5	-4.9 ÷ -4.5	1 ÷ 3	1 ÷ 10	0.1 ÷ 5
Esfera 7	-3 ÷ 2	-3 ÷ 2	-1.8 ÷ -1.6	0.5 ÷ 1	1 ÷ 20	0.1 ÷ 5
Esfera 8	-4 ÷ 0	-6 ÷ -1	0 ÷ 2	0.5 ÷ 1	1 ÷ 20	0.1 ÷ 5
Esfera 9	-3 ÷ 3	-8 ÷ -5	0 ÷ 3	0.5 ÷ 1	1 ÷ 20	0.1 ÷ 5

Tabla 10. Límites utilizados para llegar al mejor resultado con 9 esferas

La gráfica comparativa entre los valores calculados según esta solución y los medidos experimentalmente de la fuerza normal y sus momentos asociados:

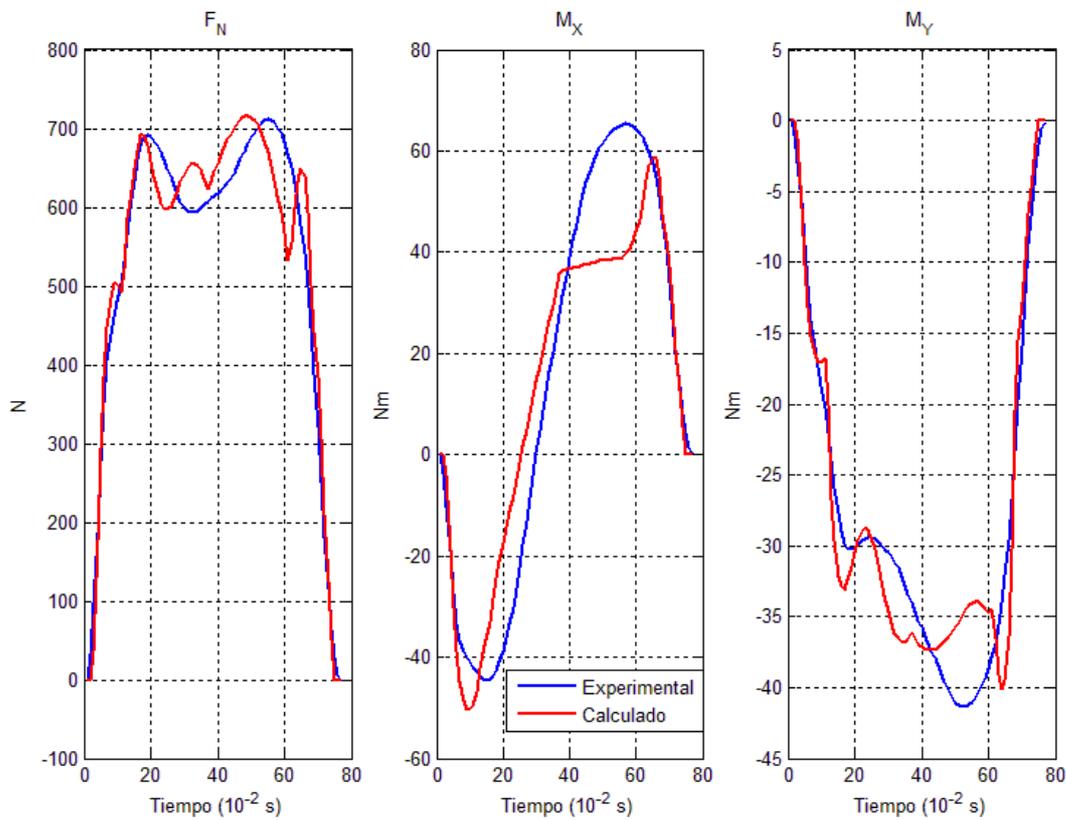


Figura 14. Fuerzas modelo 9 esferas

En la siguiente gráfica se observa la contribución de cada esfera al cálculo de la fuerza normal:

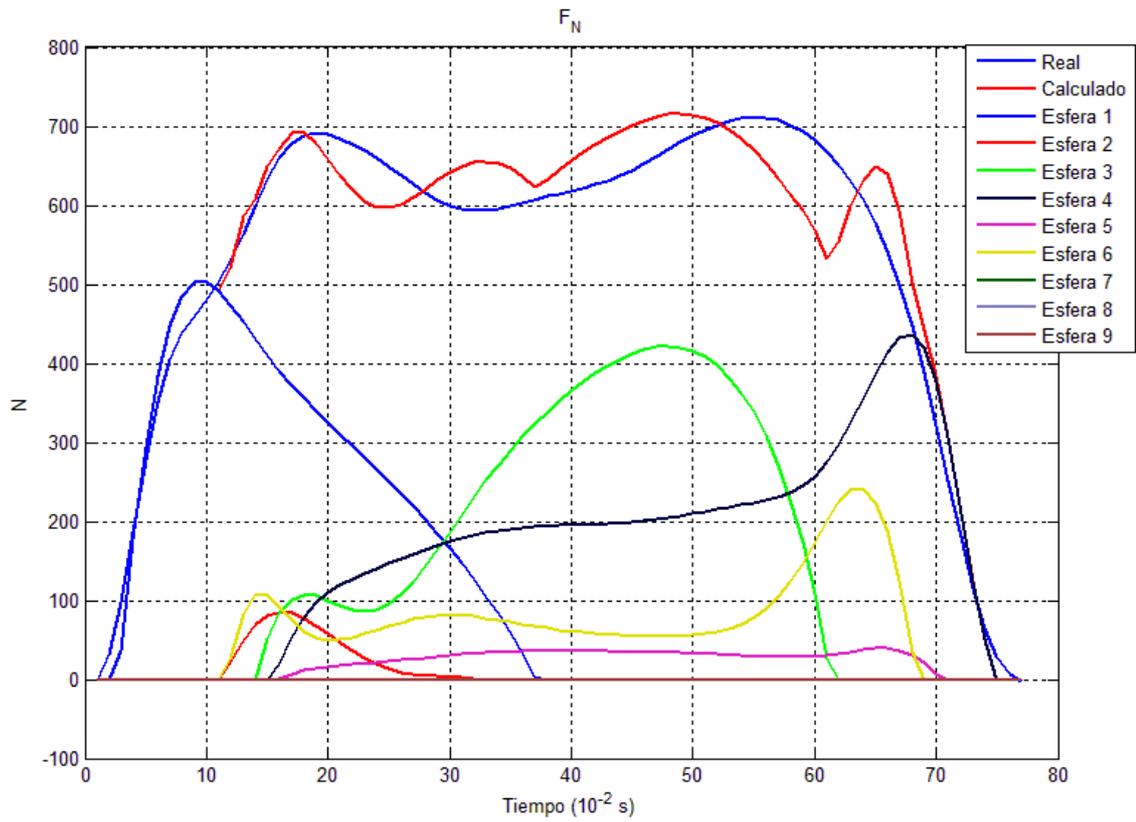


Figura 15. Contribución de cada esfera

En las siguientes gráficas se muestran la fuerza elástica y la de amortiguamiento para cada esfera:

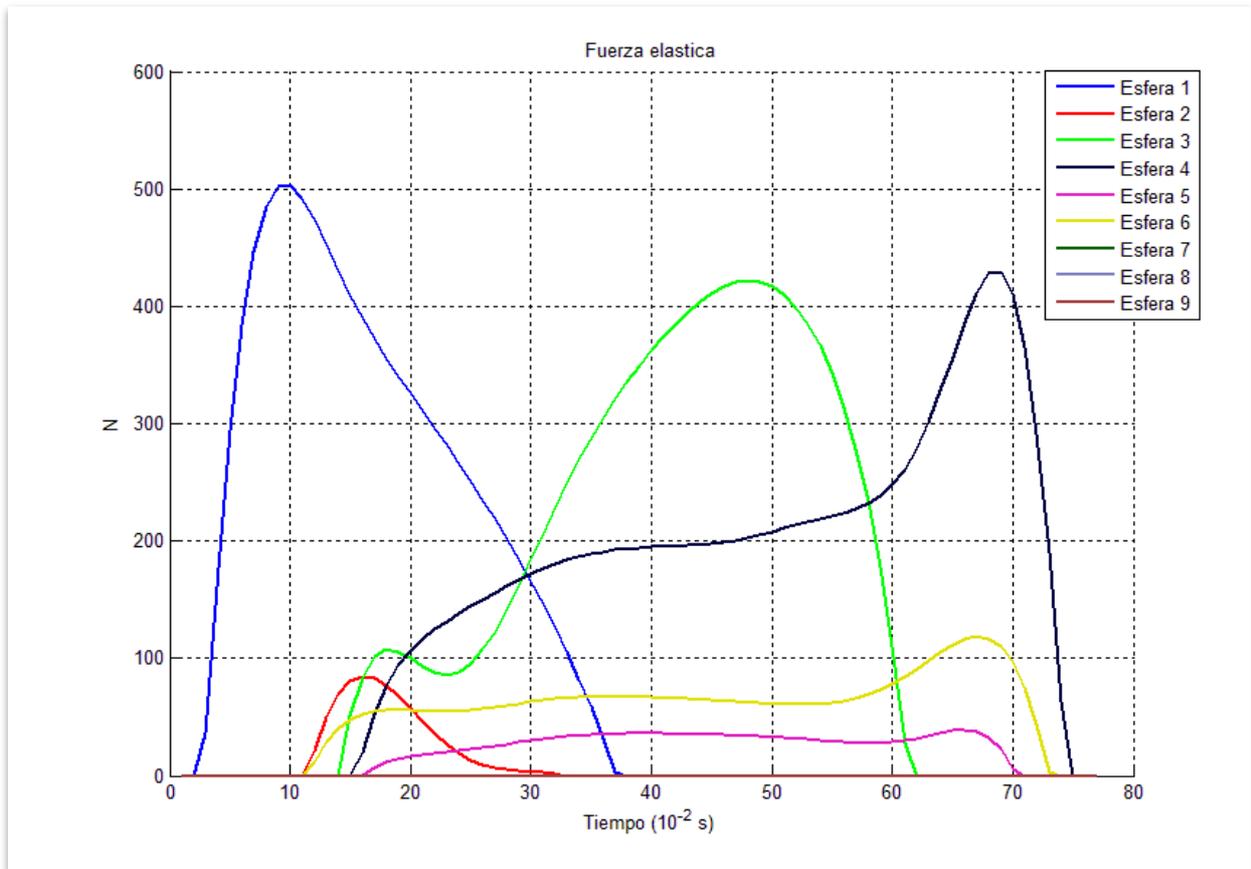


Figura 16. Fuerza elástica de cada esfera

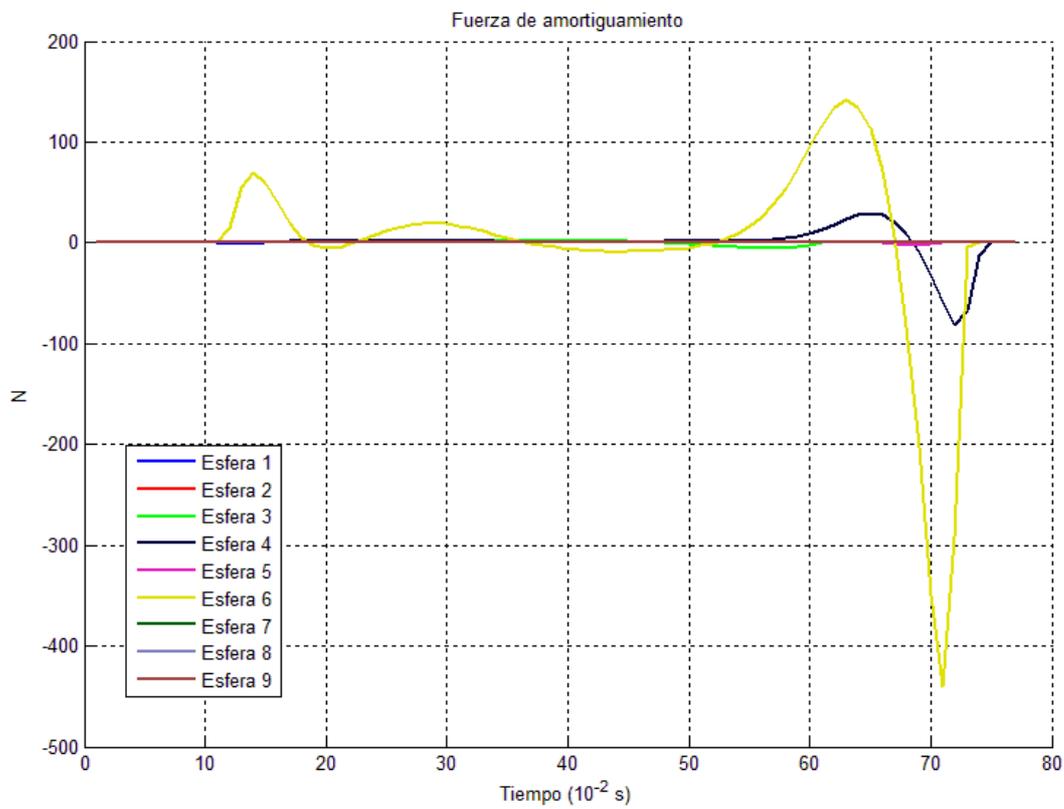


Figura 17. Fuerza de amortiguamiento de cada esfera

En esta gráfica se aprecia la pseudopetración de cada una de las esferas:

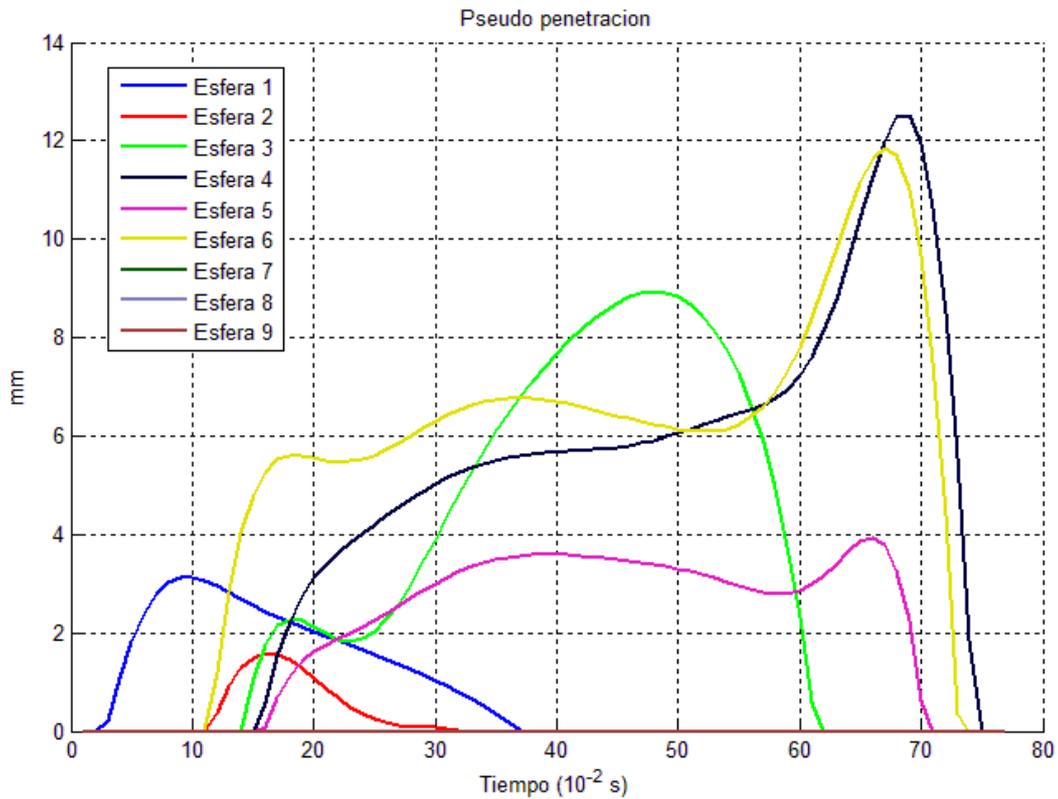


Figura 18. Pseudopetración de cada esfera

Se observa que el máximo corresponde a la esfera 4 y tiene un valor de 12.5 mm, lo cual concuerda bastante bien con la literatura existente.

En este gráfico se representan las 9 esferas junto con los orígenes de los segmentos en un instante intermedio de la fase de apoyo:

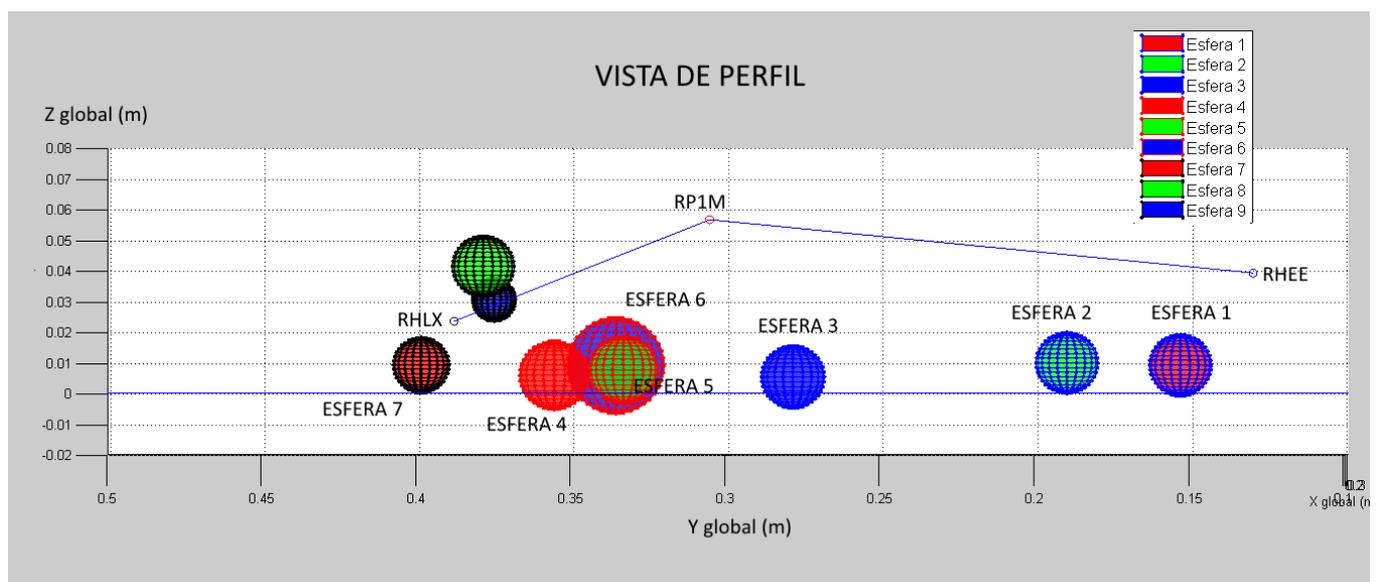


Figura 19. Vista de perfil

Las siguientes gráficas presentan la situación de las esferas y los orígenes de los segmentos en el llamado ensayo estático. En él, se pide al sujeto que apoye el pie en la plataforma de fuerza y se quede quieto. En estas

representaciones se ha girado la gráfica de manera que el eje anteroposterior coincida de manera aproximada con el eje Y global, de forma que el pie aparezca orientado hacia delante. Con este giro se facilita la comprensión del posicionamiento de las esferas.

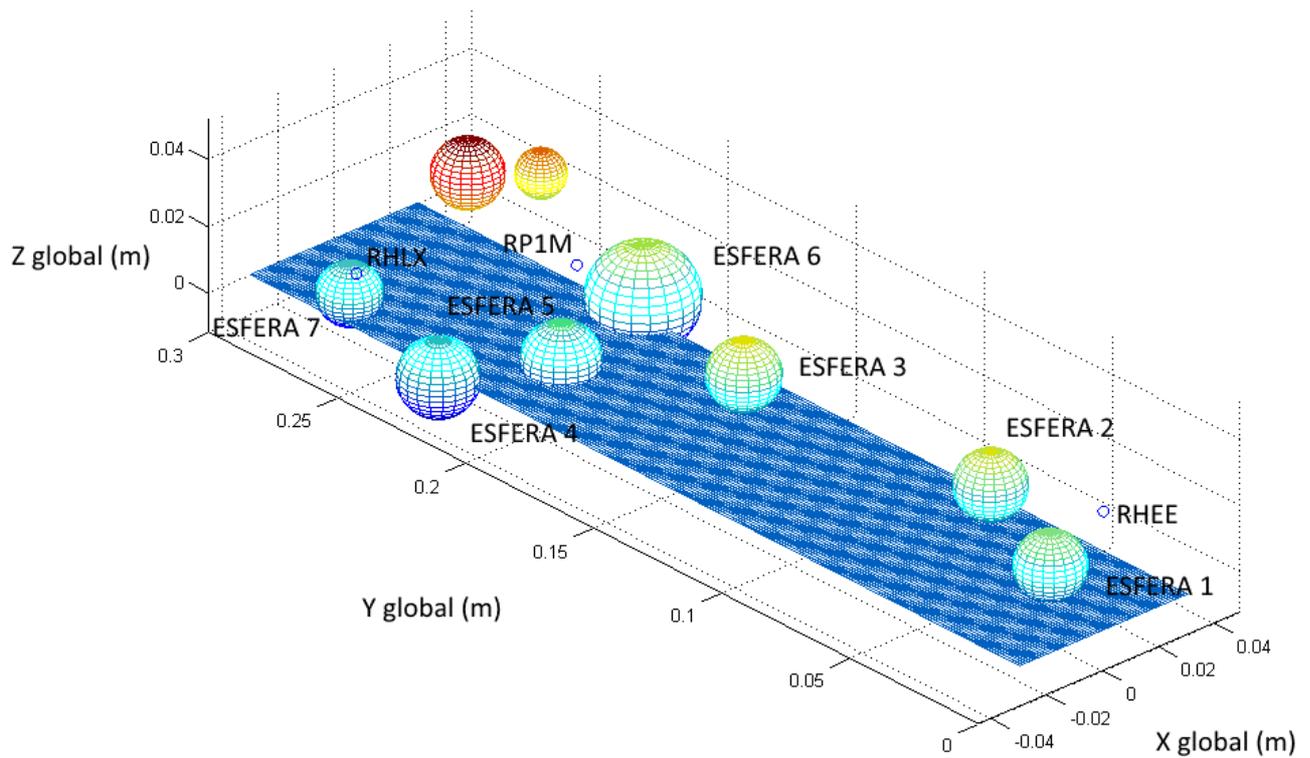


Figura 20. Vista del ensayo estático

En las siguientes gráficas se presenta el posicionamiento de las esferas de cada segmento en coordenadas locales. Para cada segmento, la gráfica superior representa una vista de perfil del pie, coincidiendo el eje horizontal de la gráfica con el anteroposterior aproximadamente y el vertical de la gráfica con el vertical anatómico aproximadamente; la gráfica inferior representa una vista en planta del pie, coincidiendo el eje horizontal de la gráfica con el eje anteroposterior aproximadamente y el vertical con el eje transversal:

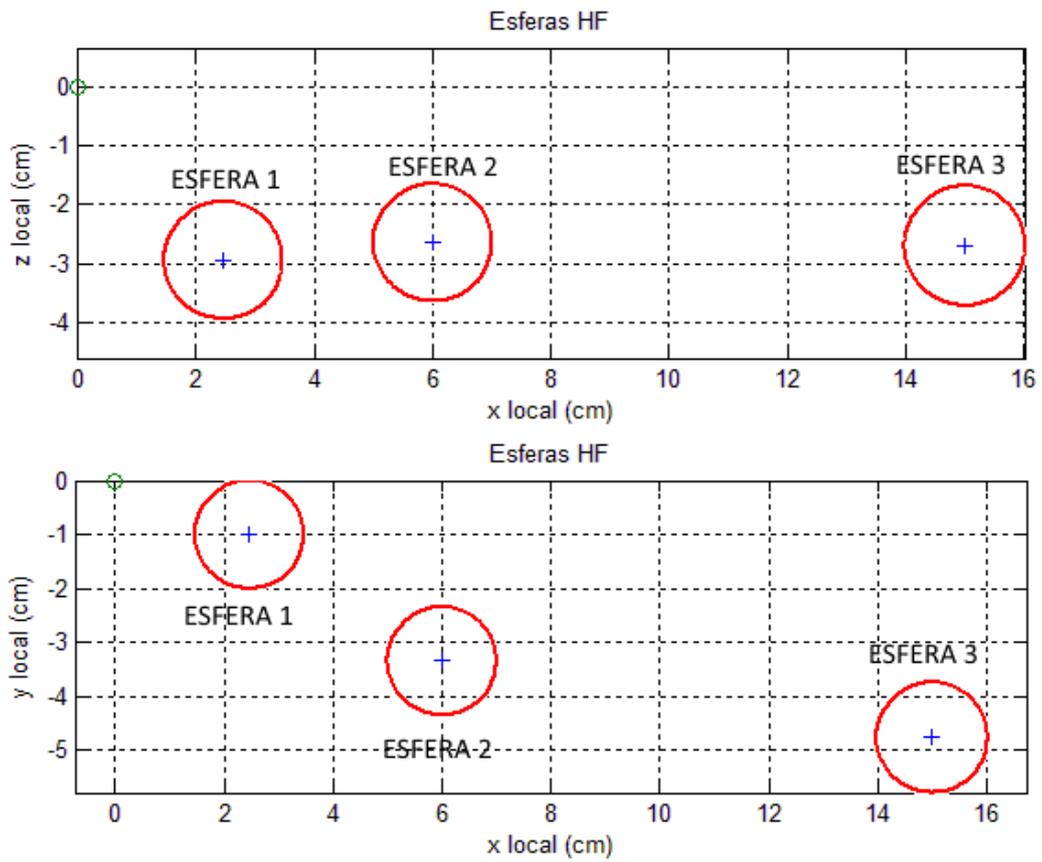


Figura 21. Representación en locales de las esferas del HF

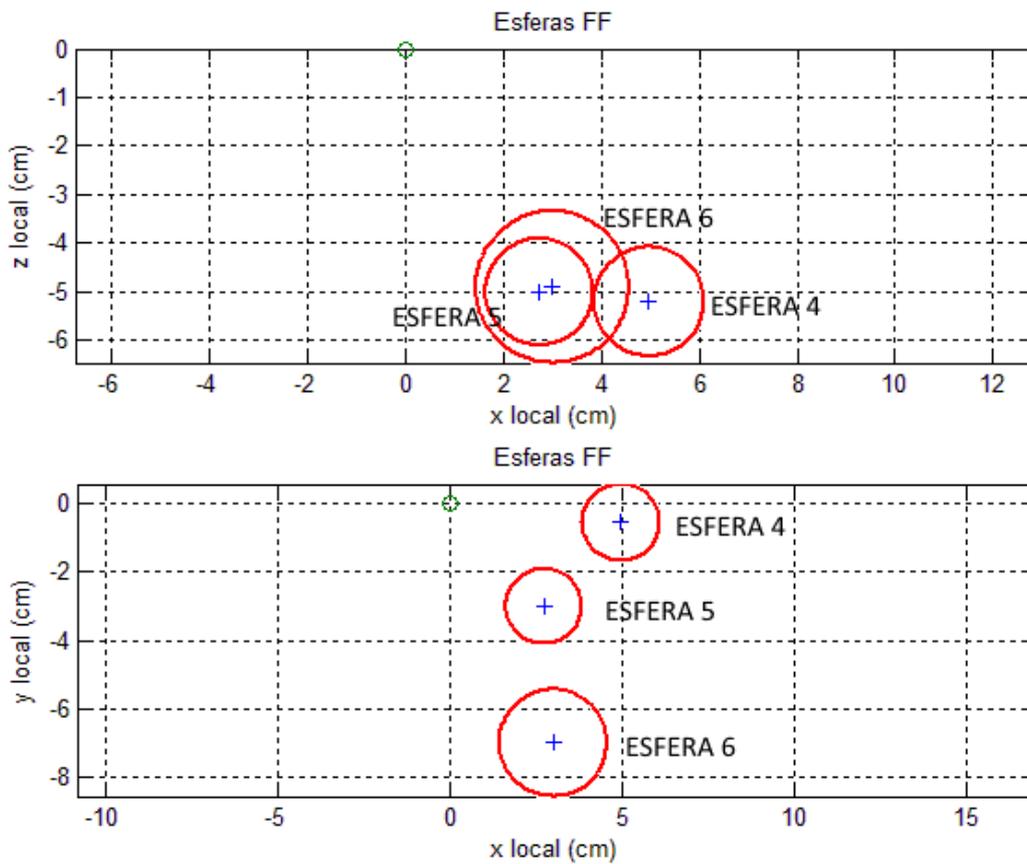


Figura 22. Representación en locales de las esferas del FF

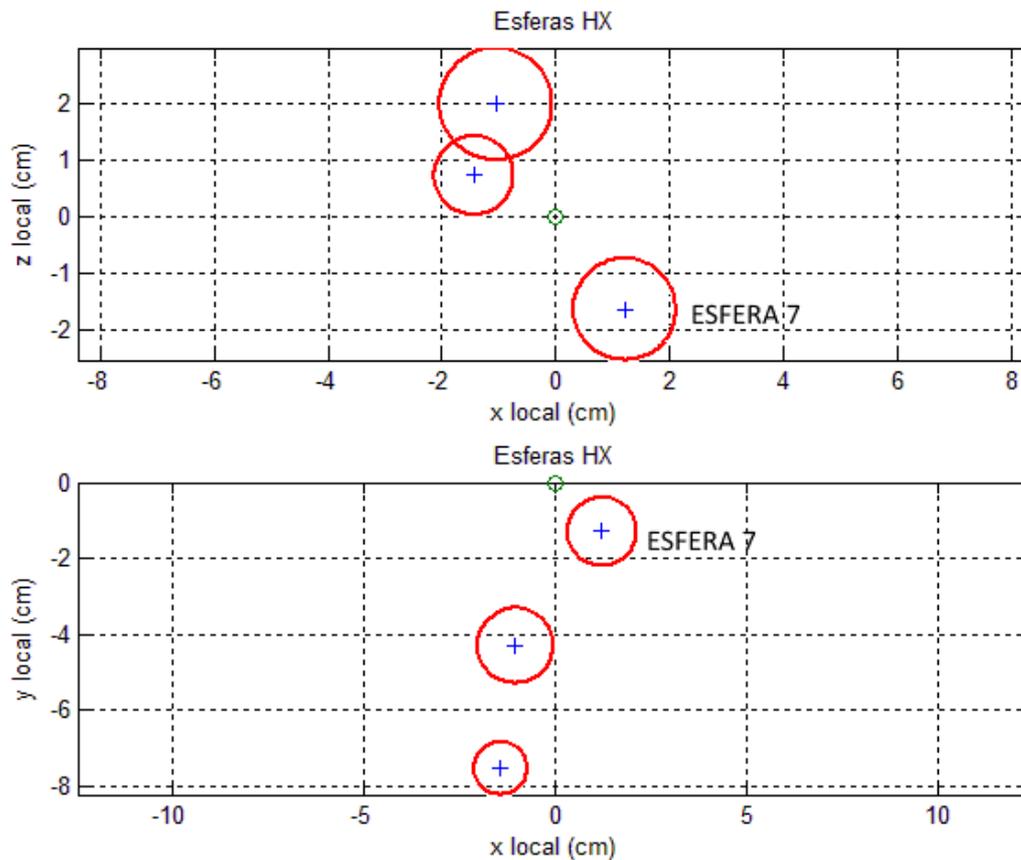


Figura 23. Representación en locales de las esferas del HX

Si comparamos los mejores modelos para 3, 6 y 9 esferas se comprueba que al aumentar el número de esferas los resultados mejoran considerablemente. Sin embargo, el error en el momento X es mayor en el caso de 9 esferas que en el de 6. Si nos fijamos en la figura 14 se observa que el momento en X calculado no consigue aproximar el máximo de esta función. Este hecho no logra explicarse ya que las esferas que penetran en dicho instante son las del FF y estas se encuentran en la parte más delantera de este segmento, por lo que deberían generar un momento en X de gran valor.

	NRMSE fuerza normal	NRSME momento X	NRSME momento Y	FUNCIÓN OBJ
3 ESFERAS	2.52	1.36	3.27	7.16
6 ESFERAS	1.10	0.95	1.18	3.24
9 ESFERAS	0.61	1.16	0.75	2.52

Tabla 11. Comparación entre los modelos de 3, 6 y 9 esferas

En el apartado de conclusiones, los resultados se han mejorado bastante respecto a un TFG anterior (9) que implementaba el modelo de esferas en un modelo monosegmento del pie. Esto lleva a pensar que el modelo multisegmento supone una mejor aproximación a la realidad física.

MEJOR VALOR DE LA FUNCIÓN OBJETIVO	
MODELO MONOSEGMENTO	17.37
MODELO MULTISEGMENTO	2.52

Tabla 12. Comparación entre los modelos mono y multisegmento

No obstante, si observamos la comparación entre el momento en ‘y’ y el medido experimentalmente se aprecian las dificultades que se han tenido para modelar el máximo de esta función. En el instante en que se produce dicho máximo se produce también el segundo máximo de la fuerza normal. A pesar de tener las 3 esferas del Forefoot colocadas en la zona donde se supone que se genera ese máximo, correspondiente al instante en que todo el peso del cuerpo se apoya en la zona delantera del pie para impulsarse hacia delante, además de la esfera correspondiente al Hallux, no se ha conseguido obtener una buena aproximación de ese máximo. Esto hace pensar que este modelo multisegmento es muy sensible al posicionamiento de las esferas en la parte delantera del pie.

En general, una de las estrategias que mejor ha funcionado hasta llegar a este resultado ha sido el ir modificando los límites en cada optimización haciendo que las esferas se situasen en mejores posiciones. Para ello es de vital importancia examinar todas las gráficas que se generan y que son similares a las presentadas anteriormente ya que identifican qué esfera está generando más o menos fuerza, en que momento comienza a generar esta y en que momento se despega del suelo...en definitiva dándonos pista acerca de cómo podemos ir mejorando los resultados.

La otra estrategia que supuso una mejora notable en el proceso fue la de aumentar el tamaño de población de 15 a 50 provocando una búsqueda más robusta capaz de no quedarse en los mínimos relativos de la función objetivo.

Como mejoras futuras existen algunos trabajos de reciente publicación que modelan el contacto ya no a través de elementos puntuales, sino mediante modelos hipervolumétricos (10). La principal ventaja que se obtiene es que las fuerzas calculadas de esta manera no presentan los ‘picos’ o ‘saltos’ que se aprecian en las gráficas anteriores si no que las fuerzas calculadas presentan una mayor suavidad (11).

Otra mejora que se puede realizar es la de utilizar un clúster para trabajos futuros pues los tiempos necesarios para obtener buenos resultados pueden llegar hasta las 10 horas o incluso más. En este sentido, se realizó un profile al ejecutar la función `busqueda_minimo` que muestra que se invierte gran tiempo en generar la matriz de rotación que pasa de ejes globales a ejes locales. Este aspecto se podría mejorar en el futuro modificando la función `Rot_Mat` construyendo la matriz de rotación de otra forma. El profile generado por MATLAB se presenta en el anexo B:

REFERENCIAS

1. **VALIANT, G.A.** *Transmission and Attenuation of Heelstrike Accelerations*. s.l. : Champaign, III, Human Kinetics Books, 1990.
2. **MEGLAN, D.A.** *Enhanced Analysis of Human Locomotion*. 1991.
3. **GILCHRIST, L.A. y WINTER, D.A.** *A Two-Part, Viscoelastic Foot Model for Use in Gait Simulations*. s.l. : Journal of Biomechanics, 29 nro. 6 (1996) 795-798.
4. **CARSON, M.C. et al.** *Kinematic analysis of a multi-segment foot model for research and clinical applications: a repeatability analysis*. s.l. : Journal of Biomechanics 34 (2001) 1299-1307, 2001.
5. **SILVA, M. et al.** *A Biomechanical Multibody Foot Model for Forward Dynamic Analysis*.
6. **WOJTYRA, M.** *Multibody Simulation Model of Human Walking*. s.l. : Mechanics Based Design of Structures and Machines: An International Journal, 31:3 (2003) 357-379, 2003.
7. **IBÁÑEZ, O. et al.** *An experimental study on the applicability of evolutionary algorithms to craniofacial superimposition in forensic identification*. s.l. : Information Sciences 179 (2009) 3998–4028, 2008.
8. **TITIANOVA, E.B. et al.** *Footprint analysis of gait using a pressure sensor system*. s.l. : Journal of Electromyography and Kinesiology 14 (2004) 275–281, 2003.
9. **MORALES, R.** *Modelado del Contacto Pie-Suelo mediante un Algoritmo basado en Técnicas Estocásticas*. s.l. : TFG Universidad de Sevilla, 2015.
10. **LOPES, D.S. et al.** *A superellipsoid-plane model for simulating foot-ground contact during human gait*. s.l. : Computer Methods in Biomechanics and Biomedical Engineering, 19:9, 954-963, 2016.
11. **SHOURIJEH, M.S. y MCPHEE, J.** *Foot-ground contact modeling within human gait*. s.l. : Multibody Syst Dyn (2015) 35:393–407, 2015.

ANEXO A

En este anexo se incluyen los códigos de MATLAB que se han utilizado.

Rutina busqueda_minimo

```
% Rutina principal del proceso de Optimización
```

```
close all
```

```
clear
```

```
clc
```

```
global x0
```

```
x0=[ 2.4632
```

```
6.0000
```

```
15.0000
```

```
4.9703
```

```
2.7323
```

```
3.0000
```

```
1.2110
```

```
-1.0522
```

```
-1.4367
```

```
-1.0000
```

```
-3.3411
```

```
-4.7578
```

```
-0.5496
```

-3.0000
-7.0000
-1.2916
-4.2993
-7.5296
-2.9443
-2.6465
-2.7000
-5.2000
-5.0000
-4.9000
-1.6355
1.9995
0.7333
1.0000
1.0000
1.0209
1.1207
1.0971
1.5635
0.9014
0.9915
0.6950
16.0950
5.3752
4.7231
3.4303
1.0000
1.0000
11.1646
9.5000
9.0137
0.1000
0.1000
0.1000
0.3433
0.1000
1.9857
1.7510
0.1000
1.9076]; % Vector inicial

```

opts.PopSize=50;
opts.ParentNumber=7;
opts.LBounds = [[2 6 10 3 2 1.5 -3 -4 -3] [-1 -4 -6 -3 -5 -7 -3 -6 -8]...
    [-4 -4 -3.4 -5.2 -5 -4.9 -1.8 0 0] [1 1 1 1 1 1 0.5 0.5 0.5]...
    [1 1 1 1 1 1 1 1 1] [0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1]]';
opts.UBounds = [[4 10 15 5 4 3 2 0 3] [1 0 -2 0 -3 -5 2 -1 -5]...
    [-2.5 -2.5 -2.7 -4.5 -4.5 -4.5 -1.6 2 3] [3 3 1.3 1.6 1.7 3 1 1 1]...
    [20 20 20 10 10 10 20 20 20] [5 5 5 5 5 5 5 5 5]]';
tic
[xmin,fmin,counteval,stopflag,out,bestever]=...
    cmaes('optimizacion_ESFERAS_Oxford_normal_y_momentos',x0,[],opts)
t=toc;
toc=toc/60 % Minutos que dura la optimización
beep

```

Función optimizacion_ESFERAS_Oxford_normal_y_momentos

```

function f=optimizacion_ESFERAS_Oxford_normal_y_momentos(x)

% Esta función recibe como argumento de entrada el vector de parametros a
% optimizar y calcula el valor de la función objetivo según la ecuación
%(4-1)

load('Joaquin08d') %Inicializamos las variables con las que trabaja el programa
%Estas variables han salido de ejecutar el programa MAIN.m

GRF2=zeros(83,6);

for i=7:83 % Estos instantes de tiempo corresponden a la fase de apoyo del pie
derecho
    [GRF2(i,:),~,~,~,~,~]=GRF_oxford_normal_esferas_mod(position,i,x);
end

f=norm(GRF(7:83,3)-GRF2(7:83,3))/(max(GRF(7:83,3))-min(GRF(7:83,3)))+...
    norm(GRF(7:83,4)-GRF2(7:83,4))/(max(GRF(7:83,4))-min(GRF(7:83,4)))+...
    norm(GRF(7:83,5)-GRF2(7:83,5))/(max(GRF(7:83,5))-min(GRF(7:83,5)));

% La matriz GRF corresponde a las fuerzas medidas experimentalmente

```

Función GRF_oxford_normal_esferas_mod

```
function [GRF, Pie_der, VPie_der, Fz, Fzelas, Fzamor,pz]=GRF_oxford_normal_esferas_mod(position,q,x)
```

```
% Esta función recibe como argumentos de entrada los datos cinematicos a  
% traves de position, el instante de tiempo q y el vector de parametros a  
% optimizar x y devuelve, entre otros, las fuerzas de reacción del suelo en  
% el instante q.
```

```
global Options
```

```
global AMTI
```

```
GRF=zeros(1,6);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Cálculo de posiciones y velocidades %%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Numero de esferas
```

```
np = size(x,1)/6;
```

```
% Coordenadas locales de los centros de las esferas(se multiplica por 0.01  
% para pasar a m)
```

```
ptospieloc(1,1:np)=0.01*x(1:np)';
```

```
ptospieloc(2,1:np)=0.01*x(np+1:2*np)';
```

```
ptospieloc(3,1:np)=0.01*x(2*np+1:3*np)';
```

```
% Radios de las esferas
```

```
radios(1,1:np)=0.01*x(3*np+1:4*np)';
```

```
pieder=zeros(3,np,size(position,1));
```

```
for i=q-1:q+1
```

```
% Datos cinematicos
```

```
pr_HF(1:4) = position(i,7*3-3:7*3);% Parámetros de Euler. Pie derecho
```

```
pr_FF(1:4) = position(i,7*2-3:7*2);
```

```
pr_HX(1:4) = position(i,7*1-3:7*1);
```

```
Rr_HF = Rot_Mat(pr_HF(1:4)); % Matriz de rotación
```

```
Rr_FF = Rot_Mat(pr_FF(1:4));
```

```

Rr_HX = Rot_Mat (pr_HX(1:4));

% Cambio a coordenadas globales
for j=1:np/3
pieder(1:3,j,i)=Rr_HF*ptospieloc(:,j) + position(i,7*3-6:7*3-4)';
end
for j=np/3+1:2*np/3
pieder(1:3,j,i)=Rr_FF*ptospieloc(:,j) + position(i,7*2-6:7*2-4)';
end
for j=2*np/3+1:np
pieder(1:3,j,i)=Rr_HX*ptospieloc(:,j) + position(i,7*1-6:7*1-4)';
end
end

dt=1/Options.VideoFrameRate;

vpiederglob=zeros(3,np);

% Calculo de la velocidad
for k=1:np
    for z=1:3
        vpiederglob(z,k)=1/(2*dt)*(pieder(z,k,q+1)-pieder(z,k,q-1));
    end
end

% Posición de los centros de las esferas en coordenadas globales
Pie_der(:,:)=pieder(:,:,q);

% Velocidad de las esferas en coordenadas globales
VPie_der(:,:)=vpiederglob(:,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Parámetros viscoelásticos %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k =10000*x(4*np+1:5*np); % N/m

```

```

cmax = 1000*x(5*np+1:6*np); % Ns/m
h = 0.01; % m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Cálculo de GRF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Inicialización de las variables
dz=zeros(1,np);
vz=zeros(1,np);
pz=zeros(1,np);
c=zeros(1,np);
Fzelas=zeros(1,np);
Fzamor=zeros(1,np);
Fz=zeros(1,np);
r=zeros(3,2);

for i =1:np

    dz(i)=radios(i) - pieder(3,i,q); % ecuación (2-3)
    vz(i)=vpiederglob(3,i);
    pz(i) = max([0 dz(i)]); % ecuación (2-4)

    switch Options.NormalForce
        case 'wojtyra' % ecuación (2-6)
            if pz(i) < h
                c(i) = cmax(i)*(3*pz(i)^2/h^2 - 2*pz(i)^3/h^3);
            else
                c(i) = cmax(i);
            end

        % Definición Fz. Ecuación (2-5)
            if dz(i) > 0
                Fzelas(i)=k(i)*dz(i);
                Fzamor(i)=-c(i)*vz(i);
                Fz(i) = max([0 Fzelas(i)+Fzamor(i)]);
            else
                Fz(i) = 0;
                Fzelas(i)=0;
            end
        end
    end
end

```

```

    Fzamor(i)=0;
end

% Resultantes de las fuerzas
    r(:,i) = pieder(:,i,q) - AMTI.pos(:,1:3)'; % Vector que va desde el
    % punto de toma de momentos de la plataforma hasta el punto de
    % contacto
    GRF(1:3)= GRF(1:3) + [0 0 Fz(i)]; % Sumatorio de las fuerzas de
    % todos los puntos de contacto
    GRF(4:6)= GRF(4:6) + cross(r(:,i)',[0 0 Fz(i)]);% Sumatorio de los
    % momentos de fuerza de todos los puntos de contacto

end

end

```

Rutina postprocesado

```

% Rutina principal de análisis de los resultados de cada optimización

clear
close all
clc

load variablesmaes
plot_fuerzas_y_esferas(xmin) % xmin es la solución obtenida en la última
    % ejecución de la rutina busqueda_minimo

plot_estatico(xmin)
 analisis(xmin);

```

Función plot_fuerzas_y_esferas

```

function plot_fuerzas_y_esferas(xplot)

%Esta funcion realiza todas las gráficas que sirven para analizar la
%solución obtenida

load('Joaquin08d')

```

```

np=size(xplot,1)/6; % Nro de esferas

global radios
radios=0.01*xplot(3*np+1:4*np);

global xsph
global ysph
global zsph
[xsph,ysph,zsph]=sphere(20);

global pie_der; % coordenadas de los centros de las esferas
pie_der=zeros(3,np,size(position,1));

for i=7:83
    [GRFplot(i,:),pie_der(:, :,i),~,Fz(i,:),Fzelas(i,:),Fzamor(i,:),pz(i, :)] = ...
        GRF_oxford_normal_esferas_mod(position,i,xplot);
end

figure()

% Gráfica de la fuerza normal
subplot(1,3,1)
plot(GRF(7:83,3), 'b', 'LineWidth', 2)
hold on;
plot(GRFplot(7:83,3), 'r', 'LineWidth', 2)
title('F_N')
xlabel('% Gait cycle'); ylabel('N')

% Gráfica del momento en x
subplot(1,3,2)
plot(GRF(7:83,4), 'b', 'LineWidth', 2)
hold on;
plot(GRFplot(7:83,4), 'r', 'LineWidth', 2)
title('M_X')
xlabel('% Gait cycle'); ylabel('Nm')

```

```

% Gráfica del momento en y
subplot(1,3,3)
plot(GRF(7:83,5), 'b', 'LineWidth',2)
hold on;
plot(GRFplot(7:83,5), 'r', 'LineWidth',2)
title('M_Y')
xlabel('% Gait cycle'); ylabel('Nm')

% Tabla con la posicion y radio de cada esfera
tabla=[(1:np);xplot(1:np)';xplot(np+1:2*np)';xplot(2*np+1:3*np)'. ...
        ;xplot(3*np+1:4*np)'];
fprintf('          x          y          z          radio\n')
fprintf('Esfera %d: %8.5f %8.5f %8.5f %8.5f \n',tabla)

figure ()
% Gráfica con la contribución de cada esfera
mpdc15 = distinguishable_colors(15);
plot(GRF(7:83,3), 'b', 'LineWidth',2)
hold on
plot(GRFplot(7:83,3), 'r', 'LineWidth',2)
hold on
for i=1:size(Fz,2)
    plot(Fz(7:83,i), 'Color',mpdc15(i,:))
end
legend('Real', 'Calculado', 'Esfera 1', 'Esfera 2', 'Esfera 3', 'Esfera 4', ...
        'Esfera 5', 'Esfera 6', 'Esfera 7', 'Esfera 8', 'Esfera 9', 'Esfera 10', ...
        'Esfera 11', 'Esfera 12')
title('F_N')
xlabel('% Gait cycle'); ylabel('N')
grid on

% Representación 3D
for i=7:83
    if i==7
        dibuja_esferas(i,MARKERS)
        resp=input('pulse 1 para ver la animacion: ','s');
        if resp~='1'
            break
        end
    else

```

```

        dibuja_esferas(i,MARKERS)
    end

end

figure
hold on
for i=1:size(pz,2)
    plot(pz(7:83,i), 'Color',mpdc15(i,:))
end
title('Pseudo penetracion')
xlabel('% Gait cycle'); ylabel('cm')
legend('Esfera 1','Esfera 2','Esfera 3','Esfera 4','Esfera 5','Esfera 6'...
    , 'Esfera 7','Esfera 8','Esfera 9','Esfera 10','Esfera 11','Esfera 12')

figure
hold on
for i=1:size(Fzamor,2)
    plot(Fzamor(7:83,i), 'Color',mpdc15(i,:))
end
title('Fuerza de amortiguamiento')
xlabel('% Gait cycle'); ylabel('N')
legend('Esfera 1','Esfera 2','Esfera 3','Esfera 4','Esfera 5','Esfera 6'...
    , 'Esfera 7','Esfera 8','Esfera 9','Esfera 10','Esfera 11','Esfera 12')

figure
hold on
for i=1:size(Fzelas,2)
    plot(Fzelas(7:83,i), 'Color',mpdc15(i,:))
end
title('Fuerza elastica')
xlabel('% Gait cycle'); ylabel('N')
legend('Esfera 1','Esfera 2','Esfera 3','Esfera 4','Esfera 5','Esfera 6'...
    , 'Esfera 7','Esfera 8','Esfera 9','Esfera 10','Esfera 11','Esfera 12')

```

Función dibuja_esferas

```
function dibuja_esferas(i,MARKERS)
```

```
% Esta función realiza la animación en 3D del movimiento de las esferas
```

```
global pie_der % Coordenadas de los centros de las esferas en globales
```

```
global radios % Radios de las esferas
```

```
global xsph
```

```
global ysph
```

```
global zsph
```

```
persistent OrigenHF % Orígenes de cada segmento
```

```
persistent OrigenFF
```

```
persistent OrigenHX
```

```
persistent L1 % Líneas que unen los orígenes de cada segmento
```

```
persistent L2
```

```
persistent sph_HF % Esferas
```

```
persistent sph_HF2
```

```
persistent sph_HF3
```

```
persistent sph_HF4
```

```
persistent sph_FF
```

```
persistent sph_FF2
```

```
persistent sph_FF3
```

```
persistent sph_FF4
```

```
persistent sph_HX
```

```
persistent sph_HX2
```

```
persistent sph_HX3
```

```
persistent sph_HX4
```

```
nesf=size(radios,1);
```

```
if i==7
```

```
    figure()
```

```
    %axis([0.15 0.4 0 0.6 -0.1 0.3])
```

```
    %grid on
```

```
    %daspect=[1 1 1];
```

```
    OrigenHF=plot3(MARKERS.RHEE(i,1),MARKERS.RHEE(i,2),...
```

```
        MARKERS.RHEE(i,3),'ob','XDataSource','MARKERS.RHEE(i,1)',...
```

```
        'YDataSource','MARKERS.RHEE(i,2)','ZDataSource',...
```

```
        'MARKERS.RHEE(i,3)');
```

```
    hold on
```

```

OrigenFF=plot3(MARKERS.RP1M(i,1),MARKERS.RP1M(i,2),...
    MARKERS.RP1M(i,3),'or','XDataSource','MARKERS.RP1M(i,1)',...
    'YDataSource','MARKERS.RP1M(i,2)','ZDataSource',...
    'MARKERS.RP1M(i,3)');
OrigenHX=plot3(MARKERS.RHLX(i,1),MARKERS.RHLX(i,2),...
    MARKERS.RHLX(i,3),'ok','XDataSource','MARKERS.RHLX(i,1)',...
    'YDataSource','MARKERS.RHLX(i,2)','ZDataSource',...
    'MARKERS.RHLX(i,3)');

L1=plot3([MARKERS.RHEE(i,1),MARKERS.RP1M(i,1)],...
    [MARKERS.RHEE(i,2),MARKERS.RP1M(i,2)], [MARKERS.RHEE(i,3),...
    MARKERS.RP1M(i,3)], 'XDataSource',...
    '[MARKERS.RHEE(i,1),MARKERS.RP1M(i,1)]',...
    'YDataSource','[MARKERS.RHEE(i,2),MARKERS.RP1M(i,2)]',...
    'ZDataSource','[MARKERS.RHEE(i,3),MARKERS.RP1M(i,3)]');
L2=plot3([MARKERS.RP1M(i,1),MARKERS.RHLX(i,1)],...
    [MARKERS.RP1M(i,2),MARKERS.RHLX(i,2)], [MARKERS.RP1M(i,3),...
    MARKERS.RHLX(i,3)], 'XDataSource',...
    '[MARKERS.RP1M(i,1),MARKERS.RHLX(i,1)]','YDataSource',...
    '[MARKERS.RP1M(i,2),MARKERS.RHLX(i,2)]','ZDataSource',...
    '[MARKERS.RP1M(i,3),MARKERS.RHLX(i,3)]');

[xfloor,yfloor]=meshgrid(0.15:0.001:0.4,0:0.001:0.6);
zfloor=zeros(size(xfloor));

mesh(xfloor,yfloor,zfloor)
if nesf==6

    sph_HF=mesh(radios(1)*xsph+pie_der(1,1,i),...
        radios(1)*ysph+pie_der(2,1,i),radios(1)*zsph+...
        pie_der(3,1,i),'XDataSource',...
        'radios(1)*xsph+pie_der(1,1,i)','YDataSource',...
        'radios(1)*ysph+pie_der(2,1,i)','ZDataSource',...
        'radios(1)*zsph+pie_der(3,1,i)','EdgeColor','blue',...
        'FaceColor',[255,0,0]/255,'FaceAlpha',.5,'Marker','.');
    sph_HF2=mesh(radios(2)*xsph+pie_der(1,2,i),radios(2)*ysph+...
        pie_der(2,2,i),radios(2)*zsph+pie_der(3,2,i),...
        'XDataSource','radios(2)*xsph+pie_der(1,2,i)',...
        'YDataSource','radios(2)*ysph+pie_der(2,2,i)',...
        'ZDataSource','radios(2)*zsph+pie_der(3,2,i)',...

```

```

        'EdgeColor', 'blue', 'FaceColor', [0,255,0]/255,...
        'FaceAlpha', .5, 'Marker', '.');
sph_FF=mesh(radios(3)*xsph+pie_der(1,3,i), radios(3)*ysph+...
pie_der(2,3,i), radios(3)*zsph+pie_der(3,3,i), ...
'XDataSource', 'radios(3)*xsph+pie_der(1,3,i)', ...
'YDataSource', 'radios(3)*ysph+pie_der(2,3,i)', ...
'ZDataSource', 'radios(3)*zsph+pie_der(3,3,i)', ...
'EdgeColor', 'red', 'FaceColor', [255,0,0]/255,...
'FaceAlpha', .5, 'Marker', '.');
sph_FF2=mesh(radios(4)*xsph+pie_der(1,4,i), radios(4)*ysph+...
pie_der(2,4,i), radios(4)*zsph+pie_der(3,4,i), ...
'XDataSource', 'radios(4)*xsph+pie_der(1,4,i)', ...
'YDataSource', 'radios(4)*ysph+pie_der(2,4,i)', ...
'ZDataSource', 'radios(4)*zsph+pie_der(3,4,i)', ...
'EdgeColor', 'red', 'FaceColor', [0,255,0]/255,...
'FaceAlpha', .5, 'Marker', '.');
sph_HX=mesh(radios(5)*xsph+pie_der(1,5,i), radios(5)*ysph+...
pie_der(2,5,i), radios(5)*zsph+pie_der(3,5,i), ...
'XDataSource', 'radios(5)*xsph+pie_der(1,5,i)', ...
'YDataSource', 'radios(5)*ysph+pie_der(2,5,i)', ...
'ZDataSource', 'radios(5)*zsph+pie_der(3,5,i)', ...
'EdgeColor', 'black', 'FaceColor', [255,0,0]/255,...
'FaceAlpha', .5, 'Marker', '.');
sph_HX2=mesh(radios(6)*xsph+pie_der(1,6,i), radios(6)*ysph+...
pie_der(2,6,i), radios(6)*zsph+pie_der(3,6,i), ...
'XDataSource', 'radios(6)*xsph+pie_der(1,6,i)', ...
'YDataSource', 'radios(6)*ysph+pie_der(2,6,i)', ...
'ZDataSource', 'radios(6)*zsph+pie_der(3,6,i)', ...
'EdgeColor', 'black', 'FaceColor', [0,255,0]/255,...
'FaceAlpha', .5, 'Marker', '.');

grid on
axis([0.15 0.4 0 0.6 -0.03 0.12])
legend([sph_HF, sph_HF2, sph_FF, sph_FF2, sph_HX, sph_HX2], ...
'Esfera 1', 'Esfera 2', 'Esfera 3', 'Esfera 4', 'Esfera 5', ...
'Esfera 6')

elseif nesf==9

sph_HF=mesh(radios(1)*xsph+pie_der(1,1,i), radios(1)*ysph+...

```

```

pie_der(2,1,i),radios(1)*zsph+pie_der(3,1,i),...
'XDataSource','radios(1)*xsph+pie_der(1,1,i)',...
'YDataSource','radios(1)*ysph+pie_der(2,1,i)',...
'ZDataSource','radios(1)*zsph+pie_der(3,1,i)',...
'EdgeColor','blue','FaceColor',[255,0,0]/255,...
'FaceAlpha',.5,'Marker','.');
```

```

sph_HF2=mesh(radios(2)*xsph+pie_der(1,2,i),radios(2)*ysph+...
pie_der(2,2,i),radios(2)*zsph+pie_der(3,2,i),...
'XDataSource','radios(2)*xsph+pie_der(1,2,i)',...
'YDataSource','radios(2)*ysph+pie_der(2,2,i)',...
'ZDataSource','radios(2)*zsph+pie_der(3,2,i)',...
'EdgeColor','blue','FaceColor',[0,255,0]/255,...
'FaceAlpha',.5,'Marker','.');
```

```

sph_HF3=mesh(radios(3)*xsph+pie_der(1,3,i),radios(3)*ysph+...
pie_der(2,3,i),radios(3)*zsph+pie_der(3,3,i),...
'XDataSource','radios(3)*xsph+pie_der(1,3,i)',...
'YDataSource','radios(3)*ysph+pie_der(2,3,i)',...
'ZDataSource','radios(3)*zsph+pie_der(3,3,i)',...
'EdgeColor','blue','FaceColor',[0,0,255]/255,...
'FaceAlpha',.5,'Marker','.');
```

```

sph_FF=mesh(radios(4)*xsph+pie_der(1,4,i),radios(4)*ysph+...
pie_der(2,4,i),radios(4)*zsph+pie_der(3,4,i),...
'XDataSource','radios(4)*xsph+pie_der(1,4,i)',...
'YDataSource','radios(4)*ysph+pie_der(2,4,i)',...
'ZDataSource','radios(4)*zsph+pie_der(3,4,i)',...
'EdgeColor','red','FaceColor',[255,0,0]/255,...
'FaceAlpha',.5,'Marker','.');
```

```

sph_FF2=mesh(radios(5)*xsph+pie_der(1,5,i),radios(5)*ysph+...
pie_der(2,5,i),radios(5)*zsph+pie_der(3,5,i),...
'XDataSource','radios(5)*xsph+pie_der(1,5,i)',...
'YDataSource','radios(5)*ysph+pie_der(2,5,i)',...
'ZDataSource','radios(5)*zsph+pie_der(3,5,i)',...
'EdgeColor','red','FaceColor',[0,255,0]/255,...
'FaceAlpha',.5,'Marker','.');
```

```

sph_FF3=mesh(radios(6)*xsph+pie_der(1,6,i),radios(6)*ysph+...
pie_der(2,6,i),radios(6)*zsph+pie_der(3,6,i),...
'XDataSource','radios(6)*xsph+pie_der(1,6,i)',...
'YDataSource','radios(6)*ysph+pie_der(2,6,i)',...
'ZDataSource','radios(6)*zsph+pie_der(3,6,i)',...
'EdgeColor','red','FaceColor',[0,0,255]/255,...
'FaceAlpha',.5,'Marker','.');
```

```

sph_HX=mesh(radios(7)*xsph+pie_der(1,7,i),radios(7)*ysph+...
pie_der(2,7,i),radios(7)*zsph+pie_der(3,7,i),...
'XDataSource','radios(7)*xsph+pie_der(1,7,i)',...
'YDataSource','radios(7)*ysph+pie_der(2,7,i)',...
'ZDataSource','radios(7)*zsph+pie_der(3,7,i)',...
'EdgeColor','black','FaceColor',[255,0,0]/255,...
'FaceAlpha',.5, 'Marker','.' );
sph_HX2=mesh(radios(8)*xsph+pie_der(1,8,i),radios(8)*ysph+...
pie_der(2,8,i),radios(8)*zsph+pie_der(3,8,i),...
'XDataSource','radios(8)*xsph+pie_der(1,8,i)',...
'YDataSource','radios(8)*ysph+pie_der(2,8,i)',...
'ZDataSource','radios(8)*zsph+pie_der(3,8,i)',...
'EdgeColor','black','FaceColor',[0,255,0]/255,...
'FaceAlpha',.5, 'Marker','.' );
sph_HX3=mesh(radios(9)*xsph+pie_der(1,9,i),radios(9)*ysph+...
pie_der(2,9,i),radios(9)*zsph+pie_der(3,9,i),...
'XDataSource','radios(9)*xsph+pie_der(1,9,i)',...
'YDataSource','radios(9)*ysph+pie_der(2,9,i)',...
'ZDataSource','radios(9)*zsph+pie_der(3,9,i)',...
'EdgeColor','black','FaceColor',[0,0,255]/255,...
'FaceAlpha',.5, 'Marker','.' );

grid on
axis([0.15 0.4 0 0.6 -0.03 0.12])
legend([sph_HF,sph_HF2,sph_HF3,sph_FF,sph_FF2,sph_FF3,sph_HX,...
sph_HX2,sph_HX3], 'Esfera 1', 'Esfera 2', 'Esfera 3', ...
'Esfera 4', 'Esfera 5', 'Esfera 6', 'Esfera 7', 'Esfera 8', ...
'Esfera 9')

elseif nesf==12

sph_HF=mesh(radios(1)*xsph+pie_der(1,1,i),radios(1)*ysph+...
pie_der(2,1,i),radios(1)*zsph+pie_der(3,1,i),...
'XDataSource','radios(1)*xsph+pie_der(1,1,i)',...
'YDataSource','radios(1)*ysph+pie_der(2,1,i)',...
'ZDataSource','radios(1)*zsph+pie_der(3,1,i)',...
'EdgeColor','blue','FaceColor',[255,0,0]/255,...
'FaceAlpha',.5, 'Marker','.' );
sph_HF2=mesh(radios(2)*xsph+pie_der(1,2,i),radios(2)*ysph+...
pie_der(2,2,i),radios(2)*zsph+pie_der(3,2,i),...

```

```

        'XDataSource', 'radios(2)*xsph+pie_der(1,2,i)', ...
        'YDataSource', 'radios(2)*ysph+pie_der(2,2,i)', ...
        'ZDataSource', 'radios(2)*zsph+pie_der(3,2,i)', ...
        'EdgeColor', 'blue', 'FaceColor', [0,255,0]/255, ...
        'FaceAlpha', .5, 'Marker', '.');
sph_HF3=mesh(radios(3)*xsph+pie_der(1,3,i), radios(3)*ysph+...
pie_der(2,3,i), radios(3)*zsph+pie_der(3,3,i), ...
        'XDataSource', 'radios(3)*xsph+pie_der(1,3,i)', ...
        'YDataSource', 'radios(3)*ysph+pie_der(2,3,i)', ...
        'ZDataSource', 'radios(3)*zsph+pie_der(3,3,i)', ...
        'EdgeColor', 'blue', 'FaceColor', [0,0,255]/255, ...
        'FaceAlpha', .5, 'Marker', '.');
sph_HF4=mesh(radios(4)*xsph+pie_der(1,4,i), radios(4)*ysph+...
pie_der(2,4,i), radios(4)*zsph+pie_der(3,4,i), ...
        'XDataSource', 'radios(4)*xsph+pie_der(1,4,i)', ...
        'YDataSource', 'radios(4)*ysph+pie_der(2,4,i)', ...
        'ZDataSource', 'radios(4)*zsph+pie_der(3,4,i)', ...
        'EdgeColor', 'blue', 'FaceColor', [255,215,0]/255, ...
        'FaceAlpha', .5, 'Marker', '.');
sph_FF=mesh(radios(5)*xsph+pie_der(1,5,i), radios(5)*ysph+...
pie_der(2,5,i), radios(5)*zsph+pie_der(3,5,i), ...
        'XDataSource', 'radios(5)*xsph+pie_der(1,5,i)', ...
        'YDataSource', 'radios(5)*ysph+pie_der(2,5,i)', ...
        'ZDataSource', 'radios(5)*zsph+pie_der(3,5,i)', ...
        'EdgeColor', 'red', 'FaceColor', [255,0,0]/255, ...
        'FaceAlpha', .5, 'Marker', '.');
sph_FF2=mesh(radios(6)*xsph+pie_der(1,6,i), radios(6)*ysph+...
pie_der(2,6,i), radios(6)*zsph+pie_der(3,6,i), ...
        'XDataSource', 'radios(6)*xsph+pie_der(1,6,i)', ...
        'YDataSource', 'radios(6)*ysph+pie_der(2,6,i)', ...
        'ZDataSource', 'radios(6)*zsph+pie_der(3,6,i)', ...
        'EdgeColor', 'red', 'FaceColor', [0,255,0]/255, ...
        'FaceAlpha', .5, 'Marker', '.');
sph_FF3=mesh(radios(7)*xsph+pie_der(1,7,i), radios(7)*ysph+...
pie_der(2,7,i), radios(7)*zsph+pie_der(3,7,i), ...
        'XDataSource', 'radios(7)*xsph+pie_der(1,7,i)', ...
        'YDataSource', 'radios(7)*ysph+pie_der(2,7,i)', ...
        'ZDataSource', 'radios(7)*zsph+pie_der(3,7,i)', ...
        'EdgeColor', 'red', 'FaceColor', [0,0,255]/255, ...
        'FaceAlpha', .5, 'Marker', '.');
sph_FF4=mesh(radios(8)*xsph+pie_der(1,8,i), radios(8)*ysph+...

```

```

pie_der(2,8,i), radios(8)*zsph+pie_der(3,8,i), ...
'XDataSource', 'radios(8)*xsph+pie_der(1,8,i)', ...
'YDataSource', 'radios(8)*ysph+pie_der(2,8,i)', ...
'ZDataSource', 'radios(8)*zsph+pie_der(3,8,i)', ...
'EdgeColor', 'red', 'FaceColor', [255,215,0]/255, ...
'FaceAlpha', .5, 'Marker', '.');

sph_HX=mesh(radios(9)*xsph+pie_der(1,9,i), radios(9)*ysph+...
pie_der(2,9,i), radios(9)*zsph+pie_der(3,9,i), ...
'XDataSource', 'radios(9)*xsph+pie_der(1,9,i)', ...
'YDataSource', 'radios(9)*ysph+pie_der(2,9,i)', ...
'ZDataSource', 'radios(9)*zsph+pie_der(3,9,i)', ...
'EdgeColor', 'black', 'FaceColor', [255,0,0]/255, ...
'FaceAlpha', .5, 'Marker', '.');

sph_HX2=mesh(radios(10)*xsph+pie_der(1,10,i), radios(10)*...
ysph+pie_der(2,10,i), radios(10)*zsph+pie_der(3,10,i), ...
'XDataSource', 'radios(10)*xsph+pie_der(1,10,i)', ...
'YDataSource', 'radios(10)*ysph+pie_der(2,10,i)', ...
'ZDataSource', 'radios(10)*zsph+pie_der(3,10,i)', ...
'EdgeColor', 'black', 'FaceColor', [0,255,0]/255, ...
'FaceAlpha', .5, 'Marker', '.');

sph_HX3=mesh(radios(11)*xsph+pie_der(1,11,i), radios(11)*...
ysph+pie_der(2,11,i), radios(11)*zsph+pie_der(3,11,i), ...
'XDataSource', 'radios(11)*xsph+pie_der(1,11,i)', ...
'YDataSource', 'radios(11)*ysph+pie_der(2,11,i)', ...
'ZDataSource', 'radios(11)*zsph+pie_der(3,11,i)', ...
'EdgeColor', 'black', 'FaceColor', [0,0,255]/255, ...
'FaceAlpha', .5, 'Marker', '.');

sph_HX4=mesh(radios(12)*xsph+pie_der(1,12,i), radios(12)*...
ysph+pie_der(2,12,i), radios(12)*zsph+pie_der(3,12,i), ...
'XDataSource', 'radios(12)*xsph+pie_der(1,12,i)', ...
'YDataSource', 'radios(12)*ysph+pie_der(2,12,i)', ...
'ZDataSource', 'radios(12)*zsph+pie_der(3,12,i)', ...
'EdgeColor', 'black', 'FaceColor', [255,215,0]/255, ...
'FaceAlpha', .5, 'Marker', '.');

grid on
axis([0.15 0.4 0 0.6 -0.03 0.12])
legend([sph_HF, sph_HF2, sph_HF3, sph_HF4, sph_FF, sph_FF2, sph_FF3, ...
sph_FF4, sph_HX, sph_HX2, sph_HX3, sph_HX4], 'Esfera 1', ...
'Esfera 2', 'Esfera 3', 'Esfera 4', 'Esfera 5', 'Esfera 6', ...
'Esfera 7', 'Esfera 8', 'Esfera 9', 'Esfera 10', 'Esfera 11', ...

```

```

        'Esfera 12')

end

daspect([1 1 1])
axis equal

else
    figure(3)
    grid on
    axis([0.15 0.4 0 0.6 -0.03 0.12])
    %axis equal
    %view(-90,0);
    pause(0.1)
    if nesf==6
        refreshdata(OrigenHF, 'caller')
        refreshdata(OrigenFF, 'caller')
        refreshdata(OrigenHX, 'caller')
        refreshdata(L1, 'caller')
        refreshdata(L2, 'caller')
        refreshdata(sph_HF, 'caller')
        refreshdata(sph_HF2, 'caller')
        refreshdata(sph_FF, 'caller')
        refreshdata(sph_FF2, 'caller')
        refreshdata(sph_HX, 'caller')
        refreshdata(sph_HX2, 'caller')
        drawnow
    elseif nesf==9
        refreshdata(OrigenHF, 'caller')
        refreshdata(OrigenFF, 'caller')
        refreshdata(OrigenHX, 'caller')
        refreshdata(L1, 'caller')
        refreshdata(L2, 'caller')
        refreshdata(sph_HF, 'caller')
        refreshdata(sph_HF2, 'caller')
        refreshdata(sph_HF3, 'caller')
        refreshdata(sph_FF, 'caller')
        refreshdata(sph_FF2, 'caller')
        refreshdata(sph_FF3, 'caller')
        refreshdata(sph_HX, 'caller')
    end
end

```

```

refreshdata(sph_HX2, 'caller')
refreshdata(sph_HX3, 'caller')
drawnow
elseif nesf==12
refreshdata(OrigenHF, 'caller')
refreshdata(OrigenFF, 'caller')
refreshdata(OrigenHX, 'caller')
refreshdata(L1, 'caller')
refreshdata(L2, 'caller')
refreshdata(sph_HF, 'caller')
refreshdata(sph_HF2, 'caller')
refreshdata(sph_HF3, 'caller')
refreshdata(sph_HF4, 'caller')
refreshdata(sph_FF, 'caller')
refreshdata(sph_FF2, 'caller')
refreshdata(sph_FF3, 'caller')
refreshdata(sph_FF4, 'caller')
refreshdata(sph_HX, 'caller')
refreshdata(sph_HX2, 'caller')
refreshdata(sph_HX3, 'caller')
refreshdata(sph_HX4, 'caller')
end

end

end

```

Función plot_estatico

```

function plot_estatico(xplot)

% Esta función realiza una gráfica 3D de la situación de las esferas en el
% ensayo estático

% Ángulo que se gira toda la gráfica para una mejor comprensión
alfa=23.5;
mat_rot=[cosd(alfa) -sind(alfa) 0;sind(alfa) cosd(alfa) 0;0 0 1];

% Datos obtenidos de la prueba del ensayo estático
Origen_FF=[0.3421;0.5667;0.0523];
Origen_HF=[0.3000;0.3961;0.0356];

```

```

Origen_HX=[0.3731;0.6441;0.0180];

RHFANAT=[0.1656 -0.9573 -0.0347;0.9717 0.1634 0;0.0058 -0.0337 0.9852];
RFFANAT=[0.2111 -0.9721 0;0.9721 0.2111 0;0 0 1];
RHX=[0.1548 -0.9721 0;0.7129 0.2111 0;0 0 0.7334];

pr_HF=getEulerParam(RHFANAT);
pr_FF=getEulerParam(RFFANAT);
pr_HX=getEulerParam(RHX);

Rr_HF=Rot_Mat(pr_HF);
Rr_FF=Rot_Mat(pr_FF);
Rr_HX=Rot_Mat(pr_HX);

np=size(xplot,1)/6; % Nro de esferas

ptospieloc(1,1:np)=0.01*xplot(1:np)';
ptospieloc(2,1:np)=0.01*xplot(np+1:2*np)';
ptospieloc(3,1:np)=0.01*xplot(2*np+1:3*np)';

for j=1:np/3
pieder(1:3,j)=Rr_HF*ptospieloc(:,j) + Origen_HF-[0.3;0.3961;0];
end
for j=np/3+1:2*np/3
pieder(1:3,j)=Rr_FF*ptospieloc(:,j) + Origen_FF-[0.3;0.3961;0];
end
for j=2*np/3+1:np
pieder(1:3,j)=Rr_HX*ptospieloc(:,j) + Origen_HX-[0.3;0.3961;0];
end

% Se establece como origen el origen HF
Origen_FF=Origen_FF-[0.3;0.3961;0];
Origen_HF=Origen_HF-[0.3;0.3961;0];
Origen_HX=Origen_HX-[0.3;0.3961;0];
origenes=[Origen_FF Origen_HF Origen_HX];
origenes=mat_rot*origenes;
pieder=mat_rot*pieder;

```

```

radios=0.01*xplot(3*np+1:4*np);

[xsph,ysph,zsph]=sphere(20);

figure()
plot3(origenes(1,:),origenes(2,:),origenes(3,),'o')
hold on
grid on

[xfloor,yfloor]=meshgrid(-0.03:0.001:0.03,0:0.001:0.3);
zfloor=zeros(size(xfloor));

mesh(xfloor,yfloor,zfloor)

for i=1:np
mesh(radios(i)*xsph+pieder(1,i),radios(i)*ysph+pieder(2,i),...
    radios(i)*zsph+pieder(3,i));
end

axis equal

figure()
plot3(origenes(1,:),origenes(2,:),origenes(3,),'o')
hold on
grid on

[xfloor,yfloor]=meshgrid(-0.03:0.001:0.03,0:0.001:0.3);
zfloor=zeros(size(xfloor));

%mesh(xfloor,yfloor,zfloor)

for i=1:np
plot3(pieder(1,i),pieder(2,i),pieder(3,i),'+')
end

axis equal

```

Función analisis

```
function [tabla_resultados]=analisis(xplot)

% Esta funcion realiza una tabla de los parametros de cada esfera y las
% representa en coordenadas locales

np=size(xplot,1)/6; % Número de esferas
neps=np/3; % Número de esferas en cada segmento

x_loc_HF=xplot(1:neps);
x_loc_FF=xplot(neps+1:2*neps);
x_loc_HX=xplot(2*neps+1:3*neps);

y_loc_HF=xplot(np+1:np+neps);
y_loc_FF=xplot(np+neps+1:np+2*neps);
y_loc_HX=xplot(np+2*neps+1:np+3*neps);

z_loc_HF=xplot(2*np+1:2*np+neps);
z_loc_FF=xplot(2*np+neps+1:2*np+2*neps);
z_loc_HX=xplot(2*np+2*neps+1:2*np+3*neps);

radio_HF=xplot(3*np+1:3*np+neps);
radio_FF=xplot(3*np+neps+1:3*np+2*neps);
radio_HX=xplot(3*np+2*neps+1:3*np+3*neps);
radio=[radio_HF;radio_FF;radio_HX];

k=10000*xplot(4*np+1:5*np);
c=1000*xplot(5*np+1:6*np);

tabla_resultados=[(1:np)' radio k c];
fprintf('          radio          k          c          \n')
fprintf('Esfera %d: %8.5f %8.5f %8.5f \n',tabla_resultados')

figure
subplot(2,1,1)
plot(x_loc_HF,z_loc_HF,'+',0,0)
viscircles([x_loc_HF,z_loc_HF],radio_HF);
xlabel('x local (cm)'); ylabel('z local (cm)')
grid on
```

```

axis equal
title('Esferas HF')
subplot(2,1,2)
plot(x_loc_HF,y_loc_HF,'+',0,0)
viscircles([x_loc_HF,y_loc_HF],radio_HF);
grid on
axis equal
title('Esferas HF')
xlabel('x local (cm)'); ylabel('y local (cm)');

figure
subplot(2,1,1)
plot(x_loc_FF,z_loc_FF,'+',0,0)
viscircles([x_loc_FF,z_loc_FF],radio_FF);
xlabel('x local (cm)'); ylabel('z local (cm)')
grid on
axis equal
title('Esferas FF')
subplot(2,1,2)
plot(x_loc_FF,y_loc_FF,'+',0,0)
viscircles([x_loc_FF,y_loc_FF],radio_FF);
grid on
axis equal
title('Esferas FF')
xlabel('x local (cm)'); ylabel('y local (cm)')

figure
subplot(2,1,1)
plot(x_loc_HX,z_loc_HX,'+',0,0)
viscircles([x_loc_HX,z_loc_HX],radio_HX);
xlabel('x local (cm)'); ylabel('z local (cm)')
grid on
axis equal
title('Esferas HX')
subplot(2,1,2)
plot(x_loc_HX,y_loc_HX,'+',0,0)
viscircles([x_loc_HX,y_loc_HX],radio_HX);
grid on
axis equal
title('Esferas HX')
xlabel('x local (cm)'); ylabel('y local (cm)')

```


ANEXO B

En este anexo se incluyen el profile generado por MATLAB al ejecutar `busqueda_minimo`.

Profile Summary

Generated 11-Sep-2016 16:59:31 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
busqueda_minimo	1	13920.407 s	0.004 s	
cmaes	1	13920.357 s	174.416 s	
...cion_ESFERAS_Oxford_normal_y_momentos	130952	13734.730 s	2433.597 s	
GRF_oxford_normal_esferas_mod	10083304	11301.133 s	8469.621 s	
Rot_Mat	90749736	1721.694 s	661.415 s	
cross	90749736	1109.818 s	1109.818 s	
matrizG	90749736	545.479 s	545.479 s	
matrizL	90749736	514.800 s	514.800 s	
cmaes>vintobounds	133571	8.611 s	8.611 s	
etime	7848	0.586 s	0.586 s	
median	9986	0.527 s	0.318 s	
num2str	2649	0.515 s	0.250 s	
fgetl	5237	0.413 s	0.413 s	
cmaes>myprctile	2619	0.216 s	0.216 s	
median>meanof	7744	0.209 s	0.209 s	
cmaes>flush	2619	0.159 s	0.030 s	
cmaes>isooctave	2624	0.129 s	0.129 s	
num2str>strvcat	1589	0.104 s	0.024 s	
cmaes>myrange	7811	0.097 s	0.097 s	
num2str>cellPrintf	1589	0.083 s	0.083 s	
mean	2621	0.083 s	0.083 s	
strjust	1589	0.080 s	0.078 s	