

ESTUDIO DE COEFICIENTES/FUNCIONES DE ASIMETRÍA EN LA MARCHA HUMANA

Autor: José Carlos Lara Gómez

Tutor: Jacek Dusza

ÍNDICE:

Currículum Vitae.

Resumen (Summary).

1. Introducción.
 - 1.1. Motivación.
 - 1.2. Descripción del proyecto.
2. Antecedentes.
 - 2.1. Equipamiento utilizado para la toma de datos.
 - 2.2. Procedimiento para la toma de datos.
3. Modelos simétricos.
 - 3.1. Doble paso medio.
 - 3.2. Separación de los pasos.
 - 3.3. Creación de los modelos simétricos mediante extensiones funcionales.
 - 3.4. Comparación de los modelos.
4. Ángulos de la marcha.
 - 4.1. Ángulo alpha.
 - 4.2. Ángulo beta.
 - 4.3. Ángulo gamma.
5. Coeficientes de asimetría e idea de las funciones de asimetría.
6. Posibles fórmulas de asimetría.
7. Funcionamiento del programa.
8. Resultados.
 - 8.1. Presentación de las distintas funciones de asimetría.
 - 8.2. Criterios para catalogar un paciente.
 - 8.3. Comparación entre distintos pacientes en función del nuevo criterio.
9. Conclusiones.
 - 9.1. Criterio de evaluación de los distintos pacientes.
 - 9.2. División de la función de asimetría en distintas fases.
10. Líneas futuras.
11. Bibliografía.

Anexo.

CURRICULUM VITAE



PERSONAL INFORMATION

Name	LARA GÓMEZ, JOSÉ CARLOS
Address	C/ FLORES DE LEMUS, N 7, 4, 6 A JAÉN 23009 SPAIN
Phone	+34 619 54 81 14
Fax	
Email	LaraGomezJC@gmail.com
Nacionality	Spanish
Date of birth	09/09/1991

EDUCATION AND TRAINING

• Dates	2003-2007 Secondary School title I.E.S. Santa Catalina de Alejandría Jaén 2007-2009 Technological Baccalaureate title I.E.S. Santa Catalina de Alejandría
---------	---

Jaén

2009-2014 Telecommunications Engineer title

E.T.S. de Ingenierías Informática y de Telecomunicación
Granada

**PERSONAL SKILLS
AND COMPETENCES**

MOTHER TONGUE

SPANISH

OTHER LANGUAGES

ENGLISH: UPPER INTERMEDIATE

FRENCH: LOW

COMPUTER SKILLS

Programming: Java, C, Matlab, VHDL and C.
Knowledge of office.

DRIVING LICENSE

LICENSE B (SPAIN)

RESUMEN / SUMMARY:

Este proyecto consiste en el estudio de los coeficientes y funciones de asimetría en tres concretos ángulos que se dan en la marcha humana.

Se parten de los datos obtenidos de colocar sensores infrarrojos en diversas partes del cuerpo a distintas personas mientras caminaban. A partir del programa realizado por Asier Sancho Hidalgo, se consigue obtener la información correspondiente al paso derecho y al izquierdo y a partir de estos obtener los correspondientes modelos equivalentes para el paso contrario. Así pues, a partir de ambos modelos se puede simular una marcha completa en función del paso izquierdo o del paso derecho y comparar ambos modelos entre sí.

El programa que he desarrollado permite estudiar esas asimetrías entre ambos modelos fijándonos en tres importantes ángulos que se dan en la marcha humana. Así pues, podemos representar las funciones de asimetría entre ambos modelos y calcular el coeficiente de asimetría utilizando diversas expresiones.

This project considers the study of asymmetry coefficients and functions in three specific angles that occur in the human gait.

We start from the data obtained from infrared sensors placed in various parts of the bodies of different people as they walked. Using the program of Asier Sancho Hidalgo we obtain the information of the right step and the left step and from these we obtain the appropriate equivalent models for the opposite step. Thus, from both models it is possible to simulate a full gait in function of the left or the right step and it is possible to compare both models between each other.

The program I have developed allows us to study these asymmetries between the two models by looking at three important angles that occur in human walking. Thus, we can present the asymmetry functions between both models and calculate the asymmetry coefficient using various expressions.

1. INTRODUCCIÓN:

1.1 Motivación:

En este proyecto se presenta un estudio de la asimetría entre el movimiento de la pierna derecha y la pierna izquierda (y sus correspondientes relaciones con la columna) en la marcha humana. Partiendo de estos datos podríamos saber si una persona padece alguna enfermedad o problema por el cuál esta persona anda de cierta manera, en función de los resultados obtenidos.

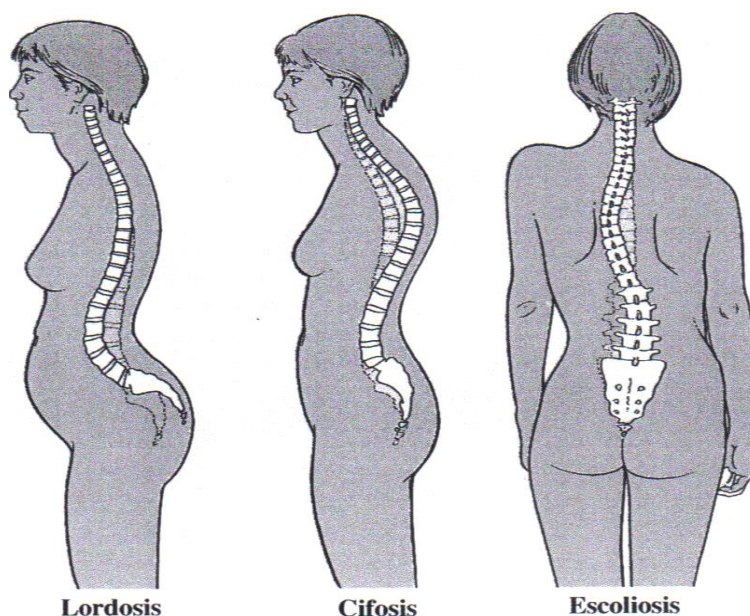


Figura 1: Distintas patologías de la columna

En la actualidad podemos encontrar diversos proyectos e investigaciones de telecomunicaciones orientados a la medicina. En nuestro caso, los datos de los que se parte fueron obtenidos al colocar sensores infrarrojos en puntos fijos del cuerpo humano a distintos pacientes mientras andaban. Esto es debido a un acuerdo que existe entre la Universidad de Valencia y el Hospital la Fe de Valencia con la Warsaw University of Technology para el estudio de enfermedades relacionadas con la columna vertebral.

Con estas investigaciones es posible el desarrollo de avances en el campo de la medicina, así como mejoras en los procedimientos ya existentes. Por ejemplo, utilizando infrarrojos para obtener datos sobre el paciente y a partir de estos detectar anomalías

podría evitar el uso de rayos X, que son perjudiciales para las personas. Así pues, todos estos avances ayudan en el desarrollo de la eSalud, que no es más que la práctica de cuidados sanitarios apoyados en tecnologías de la información y de las comunicaciones (TIC).

1.2 Descripción del proyecto:

La finalidad de este proyecto es el desarrollo de un programa que permita obtener las representaciones de las funciones de asimetría de diversos ángulos que se dan en la marcha y tienen cierta importancia, así como de los coeficientes de asimetría utilizando diversas expresiones. La definición de dichas funciones de asimetrías se verá más adelante.

El programa parte de los datos obtenidos a partir del programa de Asier Sancho Hidalgo^[1] sobre los modelos de paso derecho y paso izquierdo. Así pues, a partir de estas dos matrices de datos, podemos obtener los resultados de una manera clara y sencilla, mediante unos pasos que se irán mostrando por pantalla. Se busca que una persona que no tenga dominio sobre Matlab^[2] sea capaz de ejecutar el programa sin ningún problema y de una manera muy intuitiva.

Al ejecutar el programa se nos pedirán las matrices de los modelos de paso derecho y paso izquierdo de un paciente concreto. Una vez introducidos tendremos la opción de seleccionar entre los distintos ángulos que queremos estudiar, introducir unos nuevos datos sobre otro paciente o salir del programa.

Si seleccionamos alguno de los ángulos, se nos mostrará una imagen con la representación de dicho ángulo según los dos modelos y otro menú que nos permitirá seleccionar entre diversas expresiones para obtener el coeficiente de asimetría y la representación gráfica de la función de asimetría.

Podremos visualizar tantas funciones como deseemos interactuando con el menú y en el momento que deseemos volver atrás tendremos una opción para ello.

2. ANTECEDENTES:

2.1 Equipamiento utilizado para la toma de datos:

Los datos iniciales fueron tomados por parte del laboratorio SATI^[3] de la Universidad de Valencia. Para ello, se colocaron 26 reflectores de infrarrojos en distintos puntos del cuerpo a los distintos sujetos y se les hizo andar a paso normal por una máquina de correr.

En las siguientes figuras se pueden apreciar los puntos donde se colocaron dichos reflectores y la nombrada máquina:

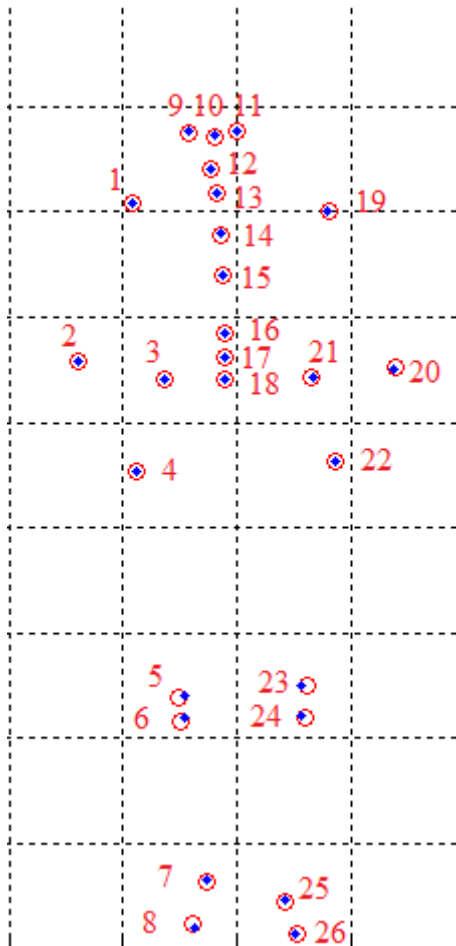


Figura 2: Puntos de colocación de los sensores infrarrojos



Figura 3: Máquina de correr utilizada para la toma de datos

Para la obtención de los datos se hizo uso de cuatro cámaras equipadas con cañones de infrarrojos y filtros de imagen de infrarrojos que actuaban simultáneamente.



Figura 4: Posición de las distintas cámaras infrarrojo

El modelo de las cámaras utilizadas es Sony XC55. Entre algunas características que poseen destacamos:

- Escáner progresivo monocromo.
- 1/3" IT escáner progresivo CCD.
- Montaje C.
- Sensibilidad: 400 lx (F5.6).
- Mínima iluminación: 0.5 lx (AGC, F1.4).
- 659 x 494 pixels.
- Equipadas con un filtro de infrarrojos plástico del tipo Edmund Thermoset ADC (CR-39), ideal para bloquear la luz visible.
- Equipadas con un cañón de ondas infrarrojas para reflejar las señales de los reflectores del sujeto. Estos cuentan con 56 leds infrarrojos del tipo HSDL-4400, de dimensión 2 x 2 mm, intensidad de 17 mW/s y velocidad de 40 ns.



Figura 5: Cámara Sony XC55 utilizada

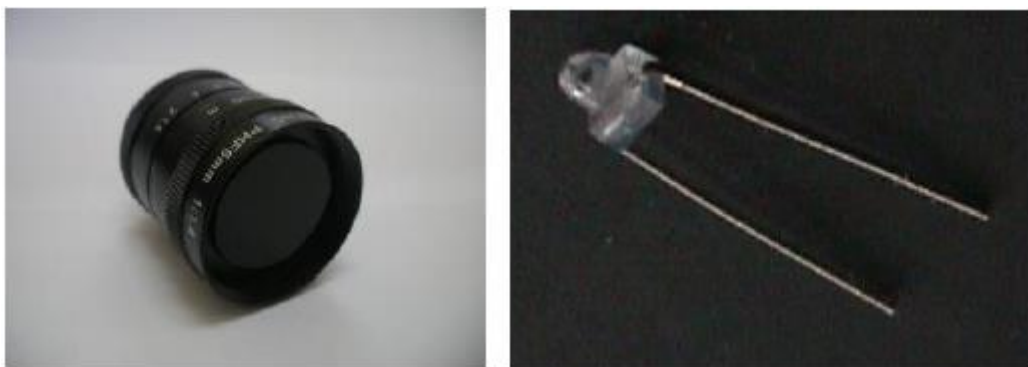


Figura 6: Filtro plástico CR-39 y led HSDL-4400

Estas cámaras iban conectadas a una tarjeta de captura de imágenes ensamblada en un ordenador. El modelo de dicha tarjeta es Framegrabber Matrox Genesis-LC, capaz de capturar señales analógicas a 140 MHz y permitiendo la sincronización y el control de las 4 señales capturadas simultáneamente.



Figura 7: Tarjeta de captura Framegrabber Matrox Genesis-LC

En cuanto al software utilizado cabe destacar:

- *Matrox Image Library (MIL)*: Librería de programación en alto nivel con un completo conjunto de funciones optimizadas para el tratamiento de imágenes: captura, transferencia, procesamiento, coincidencia de patrones, análisis Blob, visualización, etc.
- *Matrox Intellicam*: Programa basado en Windows que permite a los usuarios configurar de forma interactiva el hardware de captura de imágenes Matrox para adquirir de una fuente una determinada imagen.
- *Matrox Inspector*: Software basado en Windows que proporciona acceso interactivo a un amplio conjunto de operaciones de imágenes.

2.2 Procedimiento para la toma de datos:

Para la realización de la toma de datos en primer lugar tenemos que proceder con la calibración de la cámara. Para ello se hace uso de dos elementos: Planos coplanares para la calibración y el algoritmo de Heikkila, que permite minimizar el error de píxel en la imagen. A partir de la correcta calibración obtenemos los parámetros intrínsecos y extrínsecos:

- *Parámetros intrínsecos*: Coeficientes de reflexión coaxial y tangencial, eficiencia focal de la lente, factor de escala y punto principal.
- *Parámetros extrínsecos*: Posición de la cámara (T_x, T_y, T_z) y orientación de la cámara (α, β, γ).

El segundo paso es la captura de datos. Para ello, se van capturando datos de los reflectores 15 veces por segundo durante 30 segundos. Pese a que estos datos contienen información de las cuatro cámaras, se desea tener un único punto de cada uno de los 26 reflectantes colocados en el sujeto. Para obtener esto, se hace uso de la triangulación de los puntos, ya que cada uno de estos generará una trayectoria dependiendo de la colocación de la cámara. Así pues, el cruce de dichas trayectorias será el punto deseado.

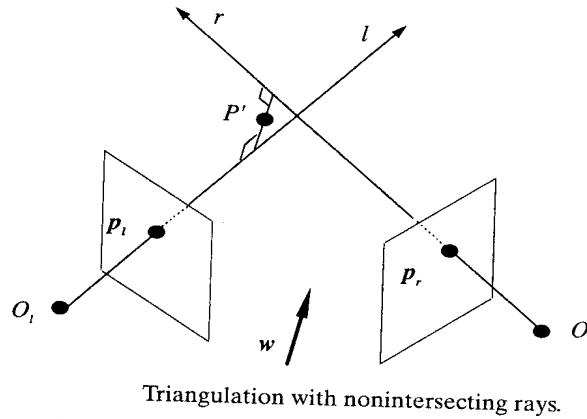


Figura 8: Triangulación de los puntos

Los puntos finales los calculamos utilizando la siguiente expresión:

$$ap_l - bR^T p_r + c(p_l \times R^T p_r) = T$$

Obtenemos el mínimo segmento entre dos rayos:

- Punto inicial: ap_l .
- Punto final: $T + bR^T p_r$

La solución será el punto medio de este segmento.

Una vez se han triangulado los puntos, ya podemos representar un esquema en tres dimensiones del sujeto caminando en la cinta.

3. MODELOS SIMÉTRICOS:

3.1 Doble paso medio:

Los datos utilizados para este proyecto están basados en el modelo de doble paso medio. Esto consiste en partir de la base de que el paso derecho y el paso izquierdo son similares en la marcha humana. Así pues, si se presentan asimetrías podrían estar asociadas a anomalías que padece el sujeto.

El movimiento del cuerpo humano se puede caracterizar en cuatro dimensiones: las tres variables espaciales y la variable de tiempo. El uso de las técnicas del modelo de doble paso medio hace posible controlar y caracterizar la cinemática del cuerpo del sujeto comparando los datos obtenidos con los datos de un paso normal en cada instante de tiempo.

La idea principal consiste en sustituir el movimiento natural cíclico del cuerpo en las tres dimensiones espaciales, por un movimiento equivalente periódico sintetizado artificialmente. El número de parámetros necesarios para caracterizar el movimiento se ve disminuido con respecto a los necesarios para caracterizar el movimiento en tiempo real. La matriz de datos consistirá en 79 x 100 elementos.

Para la realización de este modelo, se han de seguir una serie de pasos. En primer lugar, se ha de cambiar el punto de referencia de los datos al punto donde está situado el centro de masas. Para ello, se calcula el pseudo centro de gravedad (PCG) mediante los marcadores situados a la derecha e izquierda de la espina iliaca (RIS y LIS) y los marcadores que miden las desviaciones del fémur (ROF y LOF). El pseudo centro de gravedad, vendrá definido por las siguientes expresiones:

$$x_{PCG}(i) = \frac{1}{4}(x_{RIS}(i) + x_{LIS}(i) + x_{ROF}(i) + x_{LOF}(i))$$

$$y_{PCG}(i) = \frac{1}{4}(y_{RIS}(i) + y_{LIS}(i) + y_{ROF}(i) + y_{LOF}(i))$$

$$z_{PCG}(i) = \frac{1}{4}(z_{RIS}(i) + z_{LIS}(i) + z_{ROF}(i) + z_{LOF}(i))$$

El segundo paso consiste en la segmentación de los datos. Para ello se emplean los puntos de las rodillas con la finalidad de buscar el punto de cruce entre ellas. Así pues, usaremos este punto para dividir la matriz principal en dos submatrices que se corresponden con los datos de cada uno de los pasos. Puesto que estas submatrices pueden tener distintas dimensiones, habrá que normalizarlas.

Por tanto, el tercer paso consiste en la normalización de dichas matrices de datos. Para ello se emplea la Transformada de Fourier. Mediante dicha transformada, pasamos los datos al dominio espectral y comprobamos que las componentes iniciales y finales son las que mayor información poseen. Así pues, si necesitamos ampliar una matriz, añadiremos ceros y si necesitamos reducirla eliminaremos las componentes centrales necesarias.

El último paso consiste en la realización de la Transformada de Fourier inversa, para volver a pasar los datos al dominio temporal. Una vez en el dominio temporal, tendremos que unir las dos submatrices y así obtener la matriz final del modelo de doble paso medio.

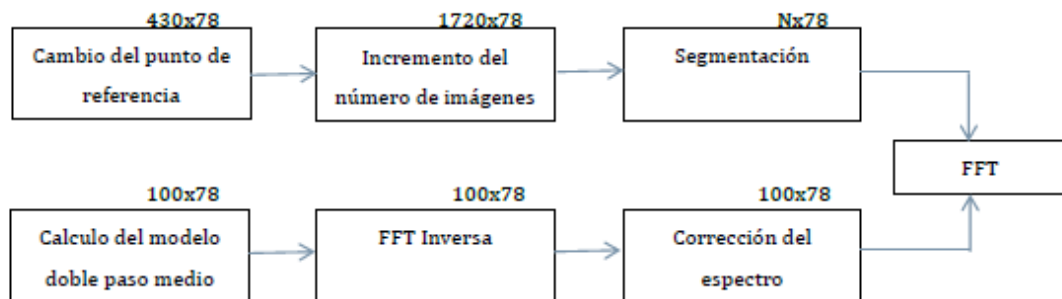


Figura 9: Esquema para el modelo de doble paso medio

3.2 Separación de los pasos:

Una vez obtenidos los datos con los que trabajar, lo que tenemos que hacer es separar la matriz de datos en dos matrices de datos normalizadas: matriz de paso derecho y matriz de paso izquierdo. Esto se consigue a partir del script de Matlab (separamatriz.m)

creado por Asier Sancho Hidalgo. Con dicho script, obtendríamos las matrices de los pasos derecho e izquierdo.

3.3 Creación de los modelos simétricos mediante extensiones funcionales:

Una vez se tienen las matrices de los pasos separados, se han de realizar los dos modelos. Para ello, se ha de predecir el movimiento de la pierna izquierda a partir del de la derecha y viceversa, de tal forma que tendríamos dos modelos de movimiento completo basados en cada una de las piernas. Podemos obtener dichos modelos a partir de los scripts de Matlab `modelopasoderecho.m` y `modelopasoizquierdo.m` creados por Asier Sancho Hidalgo.

3.4 Comparación de los modelos:

Llegados a este punto, contamos con dos modelos de paso que representan la marcha humana en función de cada una de las piernas.

Estos modelos se tratan básicamente de unas matrices de datos de igual longitud que representan el movimiento a lo largo del tiempo. Al coincidir los distintos valores que componen las matrices en cada una de sus posiciones con el mismo valor temporal, podemos realizar comparaciones entre ambos modelos.

Estos modelos de paso derecho y paso izquierdo son de los que partiremos para realizar el estudio de las funciones de asimetría.

4. ÁNGULOS EN LA MARCHA:

Para el presente proyecto, se han elegido tres ángulos que resultan de relevante importancia en el estudio de la asimetría en la marcha humana. Estos ángulos son denominados como alpha, beta y gamma a lo largo de todo el proyecto y se corresponden a los mostrados en la siguiente imagen:

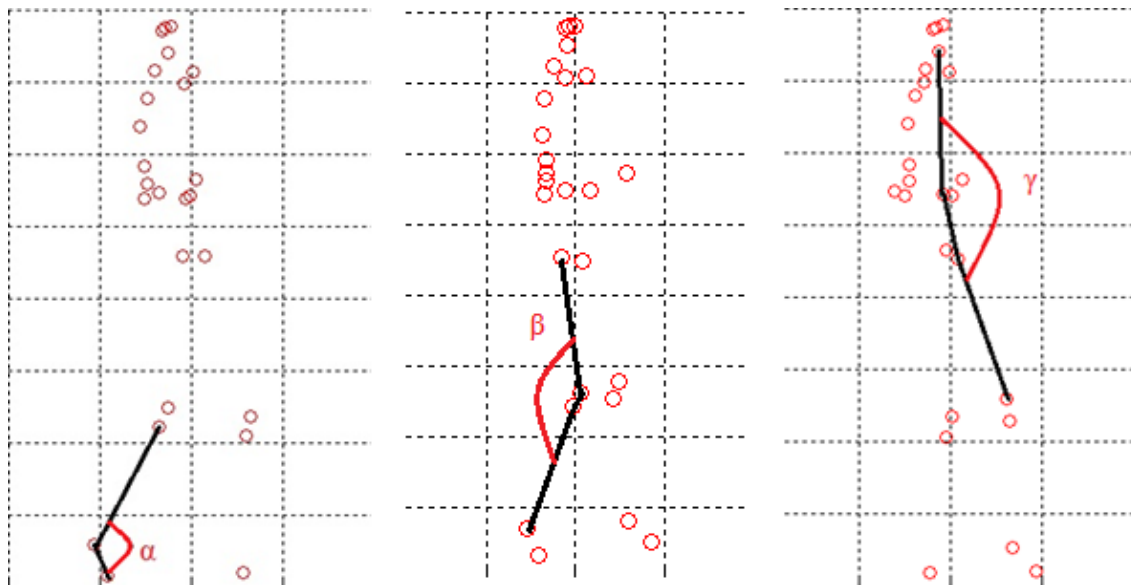


Figura 10: Distintos ángulos a estudiar

4.1 Ángulo alpha:

Si observamos la figura del ángulo alpha, podemos ver que es el formado por las rectas equivalentes de unir los puntos 6-7 y 7-8 para la pierna derecha y 24-25 y 25-26 para la pierna izquierda, tal y como se muestra en la siguiente figura:

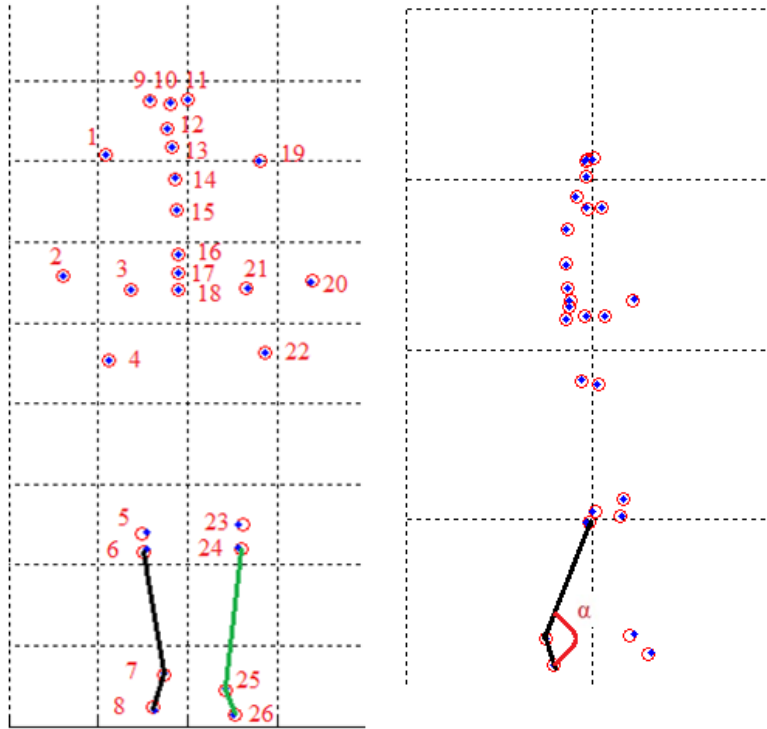


Figura 11: Ángulo alpha

Hay que tener en cuenta que cada uno de los puntos tiene tres componentes que corresponden con los ejes de coordenadas (y con 3 columnas consecutivas de la matriz de datos para cada uno de los puntos).

Así pues, para calcular este ángulo necesitaremos de dos vectores a los que denominaremos V76 y V78 para la pierna derecha y V2524 y V2625 para la pierna izquierda.

El valor del ángulo vendrá dado por:

$$\cos \alpha (i) = \frac{V76 \cdot V78}{|V76| \cdot |V78|} \text{ (pierna derecha)}$$

$$\cos \alpha (i) = \frac{V2524 \cdot V2625}{|V2524| \cdot |V2625|} \text{ (pierna izquierda)}$$

Para obtener el valor de α utilizamos el arcocoseno, pasando directamente el valor a grados:

$$\alpha = \arccos(\cos\alpha(i)) \cdot \frac{180}{\pi}$$

Los valores de α serán próximos a 90° .

4.2 Ángulo beta:

Para el ángulo beta utilizamos el ángulo que forman las rectas de unir los puntos 4-5 y 6-7 para la pierna derecha y los puntos 22-23 y 24-25 para la pierna izquierda, tal y como se muestra en la siguiente figura:

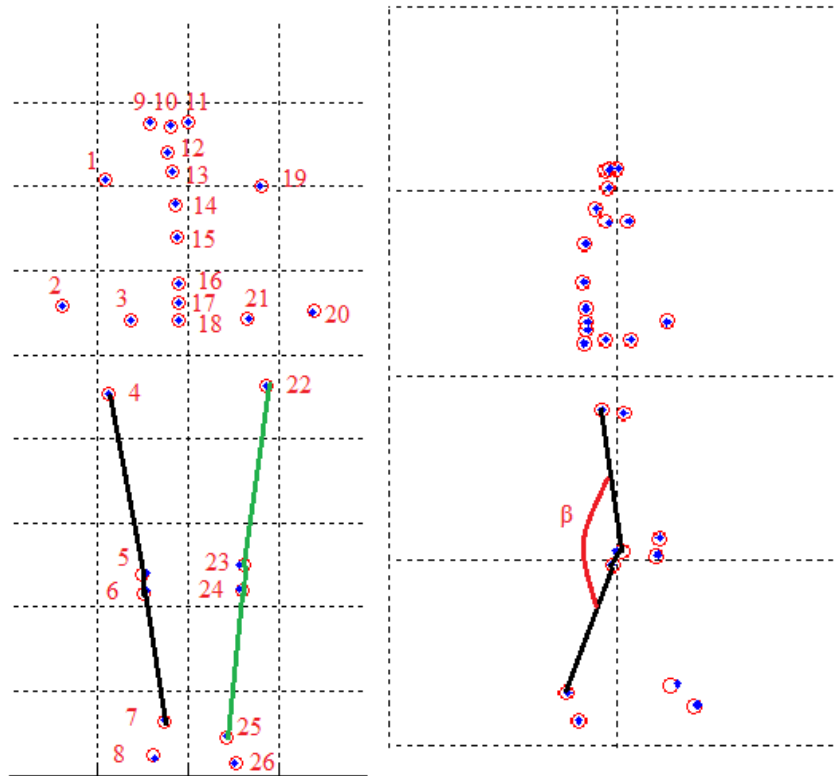


Figura 12: Ángulo beta

Para este ángulo definimos los vectores V54 y V67 para la pierna derecha y V2322 y V2524 para la pierna izquierda.

El valor del ángulo vendrá dado por:

$$\cos \beta(i) = \frac{V54 \cdot V67}{|V54| \cdot |V67|} \text{ (pierna derecha)}$$

$$\cos \beta(i) = \frac{V2322 \cdot V2524}{|V2322| \cdot |V2524|} \text{ (pierna izquierda)}$$

Para obtener el valor de β utilizamos el arcocoseno, pasando directamente el valor a grados:

$$\beta = \arccos(\cos \beta(i)) \cdot \frac{180}{\pi}$$

Los valores de β son siempre inferiores a 180° .

4.3 Ángulo gamma:

Para el ángulo gamma utilizamos el ángulo que forman las rectas de unir los puntos 4-5 y 12-18 para la pierna derecha y los puntos 22-23 y 12-18 para la pierna izquierda, tal y como se muestra en la siguiente figura:

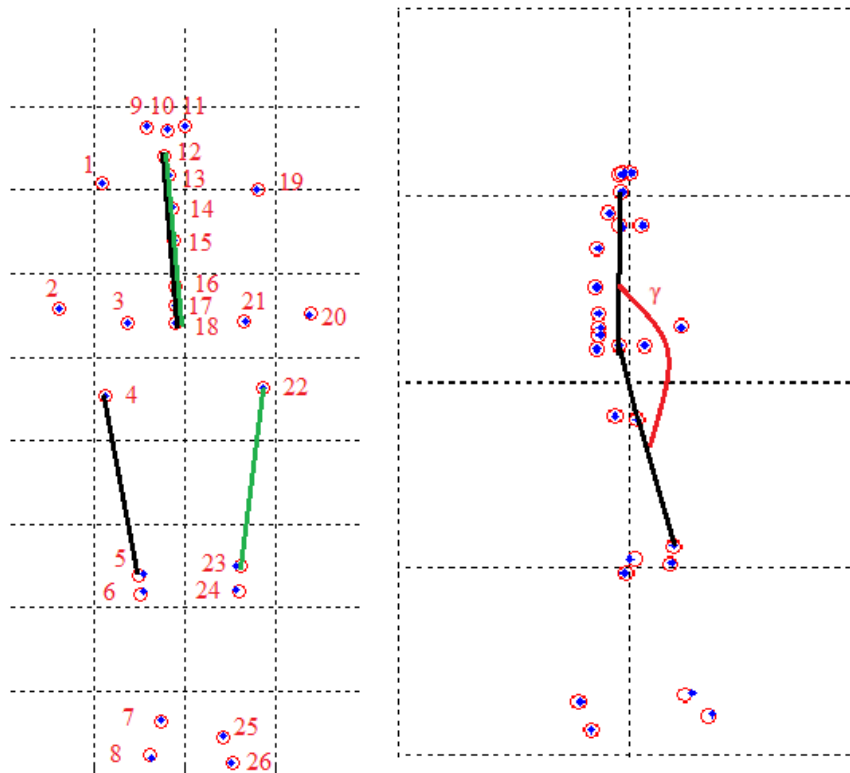


Figura 13: Ángulo gamma

Para este ángulo definimos los vectores V54 y V1218 para la pierna derecha y V2322 y V1218 para la pierna izquierda.

El valor del ángulo vendrá dado por:

$$\cos \beta(i) = \frac{V54 \cdot V1218}{|V54| \cdot |V1218|} \text{ (pierna derecha)}$$

$$\cos \beta(i) = \frac{V2322 \cdot V1218}{|V2322| \cdot |V1218|} \text{ (pierna izquierda)}$$

Para obtener el valor de γ utilizamos el arcocoseno, pasando directamente el valor a grados:

$$\gamma = \arccos(\cos \gamma(i)) \cdot \frac{180}{\pi}$$

Los valores de γ son siempre menores de 180° y próximos a este valor.

Existe otra definición el ángulo gamma. Ésta consiste en el ángulo entre los puntos 4-5 con una vertical desde el punto 4 para la pierna derecha y el ángulo entre los puntos 22-23 con una vertical desde el punto 22.

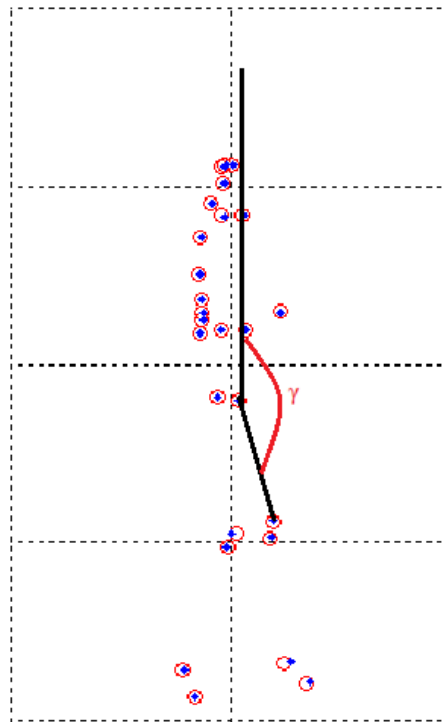


Figura 14: Forma alternativa del ángulo gamma

5. COEFICIENTES DE ASIMETRÍA E IDEA DE LAS FUNCIONES DE ASIMETRÍA:

Hasta el momento, sólo se había estudiado la asimetría global en el movimiento, por ejemplo, comparando los valores máximos o mínimos de los ángulos formados en ambas piernas.

Con la creación de los modelos de paso derecho y paso izquierdo, podemos simular el movimiento completo de una persona basándonos en los respectivos movimientos de la pierna derecha y la pierna izquierda. Estos modelos nos proporcionan unas matrices de valores de igual tamaño, de tal forma que podemos comparar los valores de los ángulos en cada pierna en un instante de tiempo deseado, puesto que estos movimientos estarían sincronizados.

Así pues, nace la idea del estudio de las funciones de asimetría. A partir de dichos modelos, podemos comparar la asimetría que se producen entre ambas piernas en los distintos instantes de tiempo, definiéndose así una función temporal.

A partir de los valores obtenidos de la función de asimetría podemos definir el coeficiente de asimetría, que mide el grado de asimetría de la distribución con respecto a la media aritmética de los valores obtenidos en los distintos instantes de tiempo de la función de asimetría:

$$CA = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

6. POSIBLES FÓRMULAS DE ASIMETRÍA:

Conociendo los valores de los ángulos en los distintos instantes de tiempo, podríamos definir diferentes funciones de asimetría. Para este proyecto se permite obtener 6 expresiones, cada una con sus correspondientes características. Estas son:

$$FA_1(i) = AngIzq(i) - AngDer(i) [^\circ]$$

$$FA_2(i) = \frac{AngIzq(i) - AngDer(i)}{AngIzq(i) + AngDer(i)} [\%]$$

$$FA_3(i) = \left| \frac{AngDer(i)}{AngIzq(i)} - \frac{AngIzq(i)}{AngDer(i)} \right| [\%]$$

$$FA_4(i) = \left(\frac{AngDer(i)}{AngIzq(i)} + \frac{AngIzq(i)}{AngDer(i)} - 2 \right) [\%]$$

$$FA_5(i) = \frac{AngDer(i) - AngIzq(i)}{\max(AngDer(i), AngIzq(i))} [\%]$$

$$FA_6(i) = \left| \frac{AngDer(i) - AngIzq(i)}{\max(AngDer(i), AngIzq(i))} \right| [\%]$$

A la hora de calcular dichas funciones de asimetría, se ha utilizado un criterio: Se observa en cada instante de tiempo la diferencia entre los ángulos derecho e izquierdo y si este valor es menor de 2° , consideramos el valor de la función igual a 0. Esto es debido a que somos incapaces de observar una diferencia en grados por debajo de este valor.

7. FUNCIONAMIENTO DEL PROGRAMA:

Para la ejecución del programa que realiza los cálculos de funciones y coeficientes de asimetría, se hace uso de Matlab. En mi caso, he programado los distintos scripts mediante la versión R2010a, por lo que es recomendada una versión compatible del programa con esta versión.

Antes de comenzar, en el programa deberemos asignar como “current folder”, aquel destino donde se encuentran los archivos de Matlab. En dicho destino, deberemos añadir también los datos de los pacientes que se desean estudiar.

Así pues, al ejecutar el comando “Principal”, el programa se iniciará. De una manera intuitiva podremos movernos por los distintos menús. En primer lugar, se nos pedirán los datos correspondientes al modelo de pierna derecha y de pierna izquierda. Deberemos escribir correctamente los nombres de los ficheros, ya que si se escribe el nombre de un fichero no existente, el programa se cerrará, y si mezclamos datos de distintos pacientes, los resultados obtenidos no tendrán sentido.

```
>> Principal
Seleccione el fichero con los datos referentes al movimiento de pierna derecha
Cargar archivo de entrada: |
```

En este momento podremos seleccionar entre los distintos ángulos de los que deseamos calcular las funciones y coeficientes de asimetría o salir del programa.

```
MENU

1.- Angulo alpha
2.- Angulo beta
3.- Angulo gamma
4.- Seleccionar nuevo paciente

0.- Salir
Seleccione una opcion:
```

Si seleccionamos cualquiera de los ángulos, nos aparecerá una ventana que muestra la representación temporal del ángulo seleccionado para ambas piernas. A su vez,

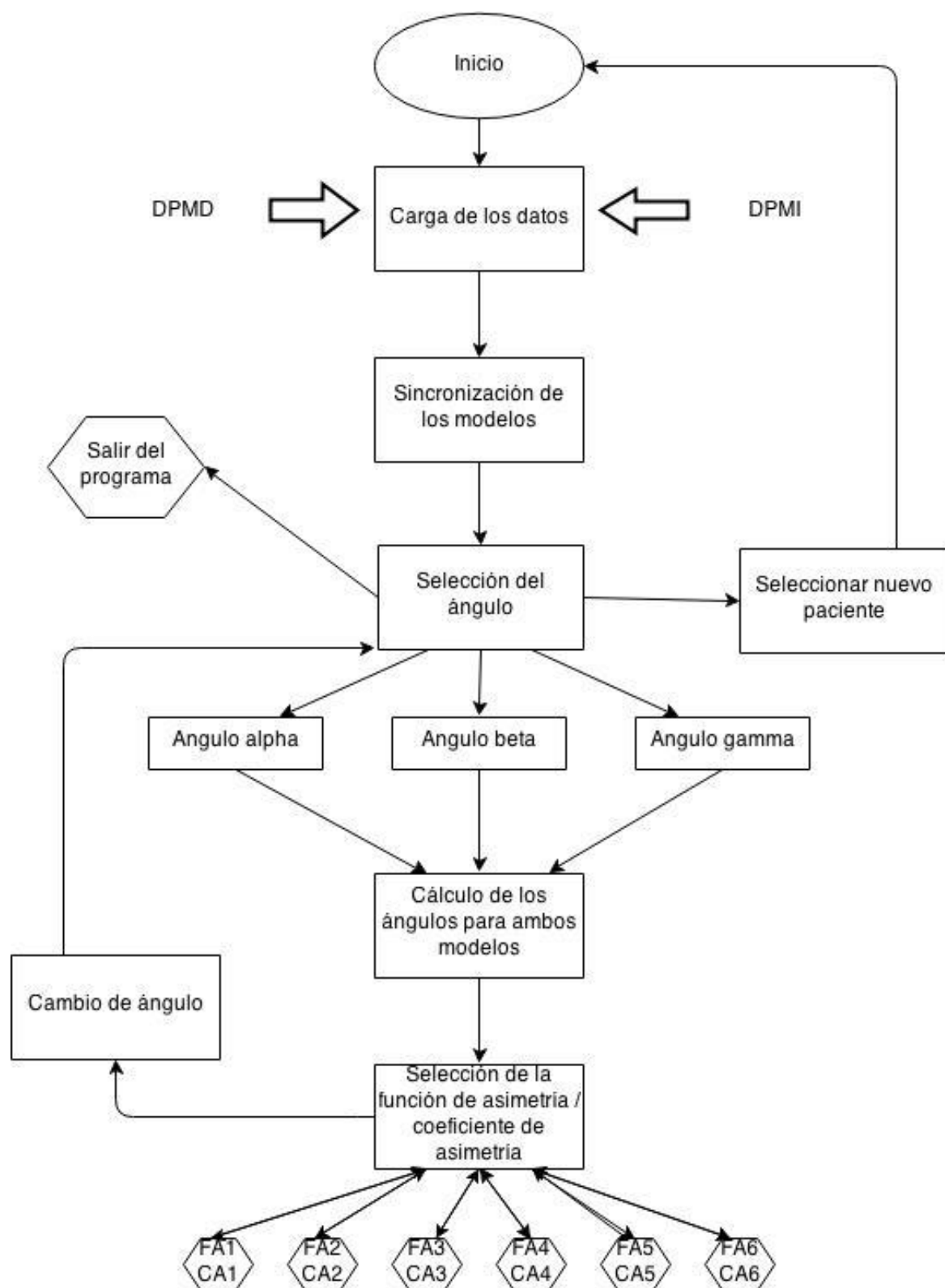
nos aparecerá otro menú para elegir entre las distintas expresiones de funciones de asimetría.

`Seleccione la función de asimetría que desea calcular (de 1 a 6):`

Ahora podremos ir seleccionando entre las distintas expresiones y se nos mostrarán los valores de las funciones de asimetría y su correspondiente coeficiente de asimetría. También tendremos la opción de volver al menú anterior.

Como se puede ver, el funcionamiento del programa es muy simple, incluso para personas que no cuentan con mucha experiencia con el uso de Matlab o conocimientos informáticos en general.

Adicionalmente, he creado otras funciones que nos permiten representar el movimiento de la persona andando. Estas funciones reciben como argumento dos ficheros de datos, que corresponderían con el modelo de paso izquierdo y paso derecho. Así pues, se representarían simultáneamente dichos valores y podríamos compararlos de una manera visual. Estos ficheros son graf3D (para mostrar el sujeto andando en las 3 dimensiones), graf2D_costado (para mostrar al sujeto andando visto desde el costado), graf2D_frente (para mostrar al sujeto andando visto desde en frente) y graf2D_suelo (para mostrar al sujeto andando visto desde abajo). Ejemplo: graf2D_costado(der,izq).



8. RESULTADOS:

8.1 Presentación de las distintas funciones de asimetría:

Como ya se ha mencionado anteriormente, se han definido distintas expresiones para las funciones de asimetría, cada una con sus propias características. Para un mismo paciente y un mismo ángulo (en este caso el ángulo α) obtenemos las siguientes funciones de asimetría:

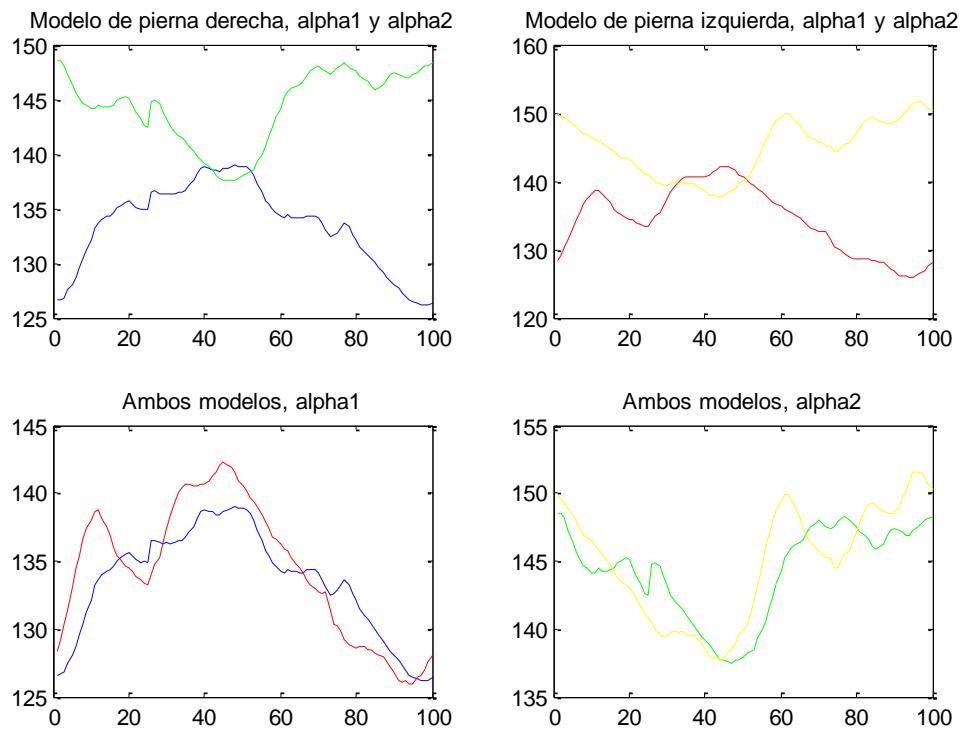


Figura 15: Ángulo α (ambos modelos)

Primera función: Se trata simplemente de la diferencia en grados entre ambos modelos para cada uno de los intervalos temporales.

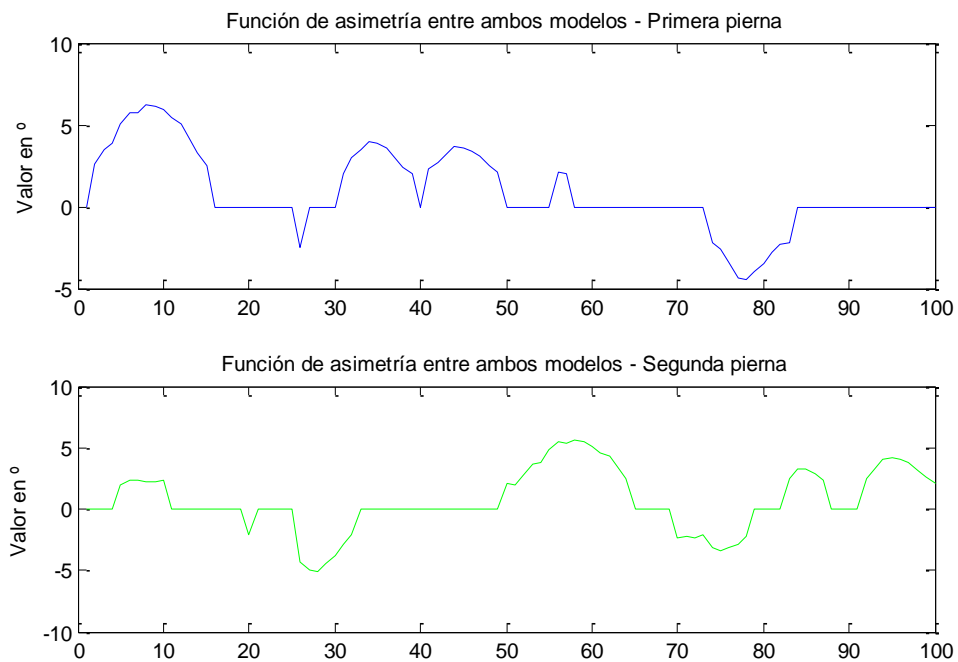


Figura 16: Ángulo alpha – Primera función

Segunda función: Con esta función se busca obtener una función similar a la anterior, pero dada en porcentaje.

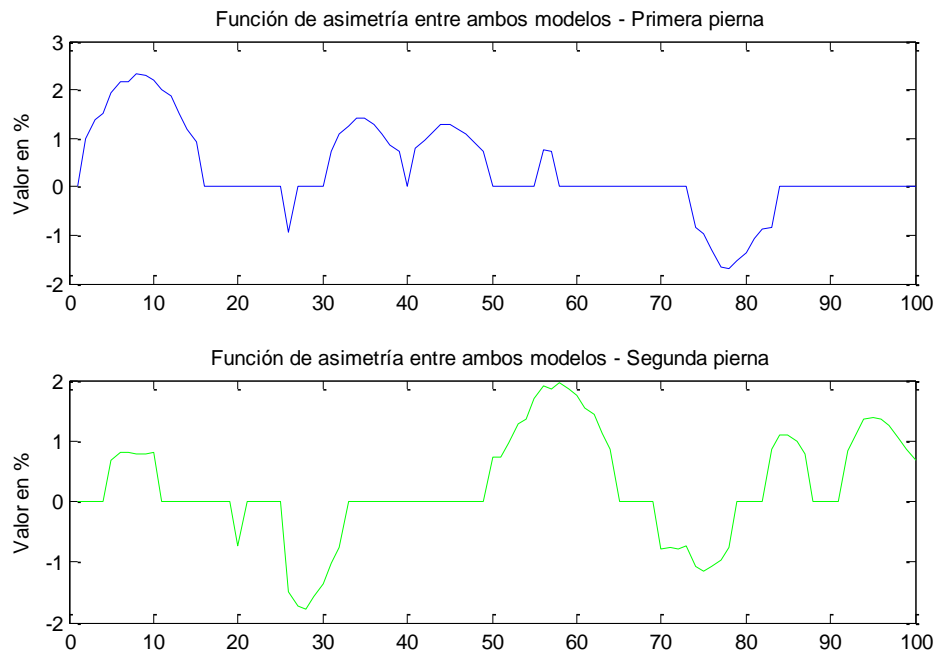


Figura 17: Ángulo alpha – Segunda función

Tercera función: Se trata de una función muy sensible a las variaciones. Presenta grandes saltos conforme las diferencias en los distintos instantes temporales son más grandes.

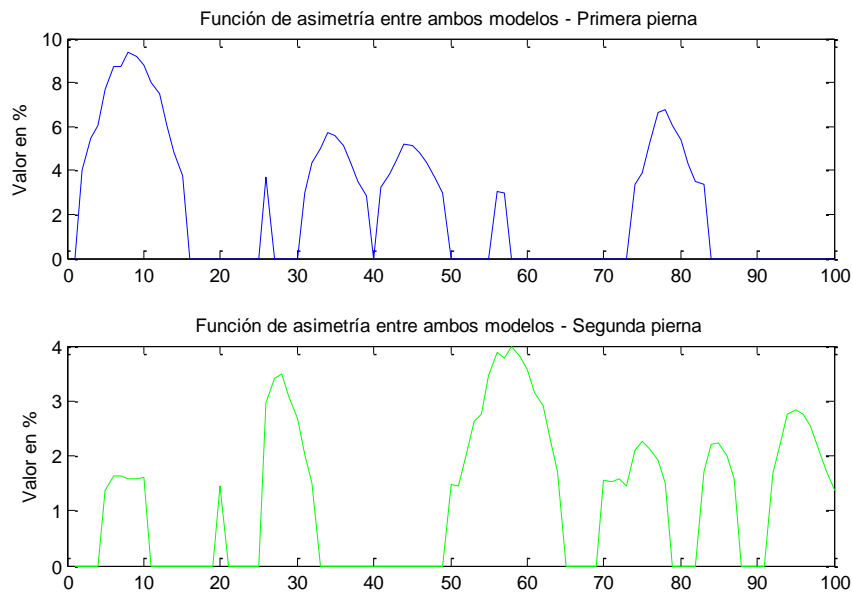


Figura 18: Ángulo alpha – Tercera función

Cuarta función: Al igual que la anterior, también es muy sensible a los cambios, pero los representa en una escala mucho menor (del 0 al 1 %).

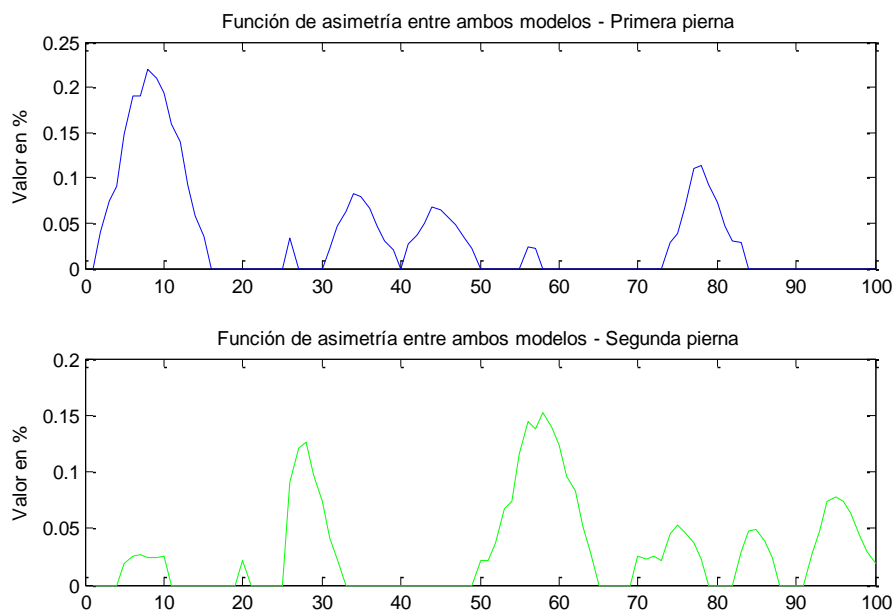


Figura 19: Ángulo alpha – Cuarta función

Quinta función: Esta función es más sensible en cada uno de los intervalos temporales al valor del ángulo que presente un mayor valor en cada caso.

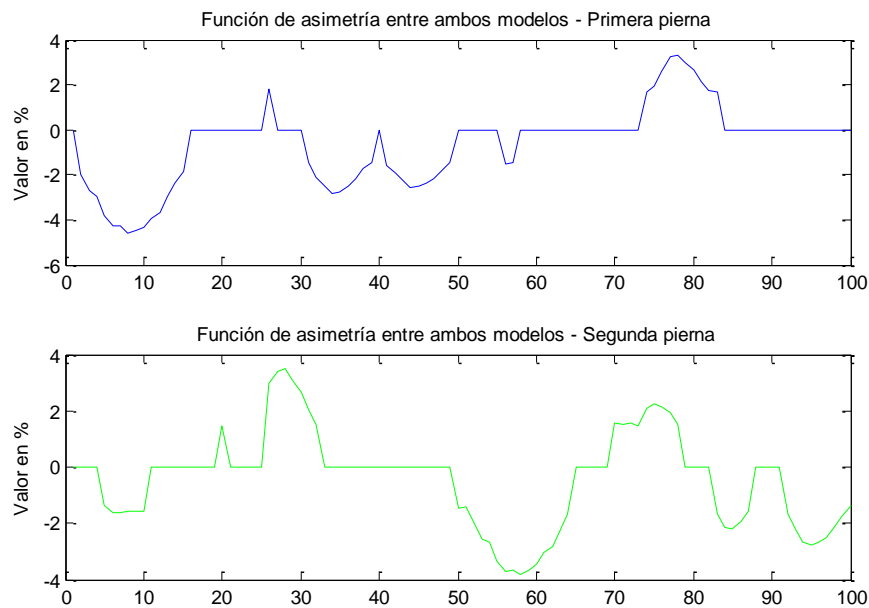


Figura 20: Ángulo alpha – Quinta función

Sexta función: Muy similar a la anterior, pero sólo en un intervalo positivo debido al valor absoluto.

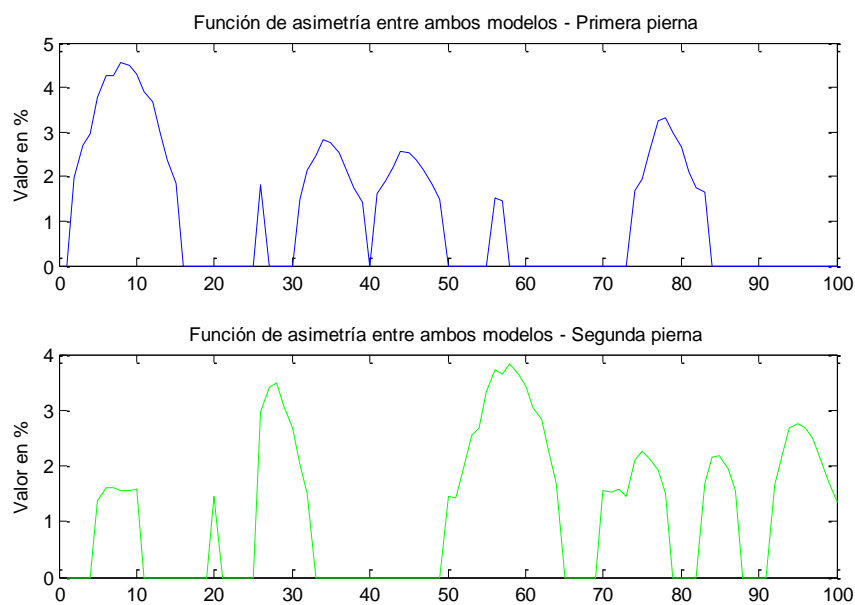


Figura 21: Ángulo alpha – Sexta función

8.2 Criterios para catalogar un paciente:

Existen distintos criterios por los que podemos definir si una persona anda bien o no. En el caso de Asier Sancho Hidalgo, el criterio que seleccionó fue que cuando el paso alcanza su máxima amplitud, la columna vertebral está lo más recta posible. Así pues, observando los instantes en los que la columna está recta y los instantes en los que el paso es máximo, podemos comparar estos instantes y en función de la diferencia entre ellos, catalogar a los distintos pacientes.

Así pues, analizamos tres pacientes que según este criterio resultaron bueno, malo y regular. Para ello, nos fijaremos en las dos primeras funciones de asimetría (la primera en grados y la segunda en porcentaje):

Paciente bueno:

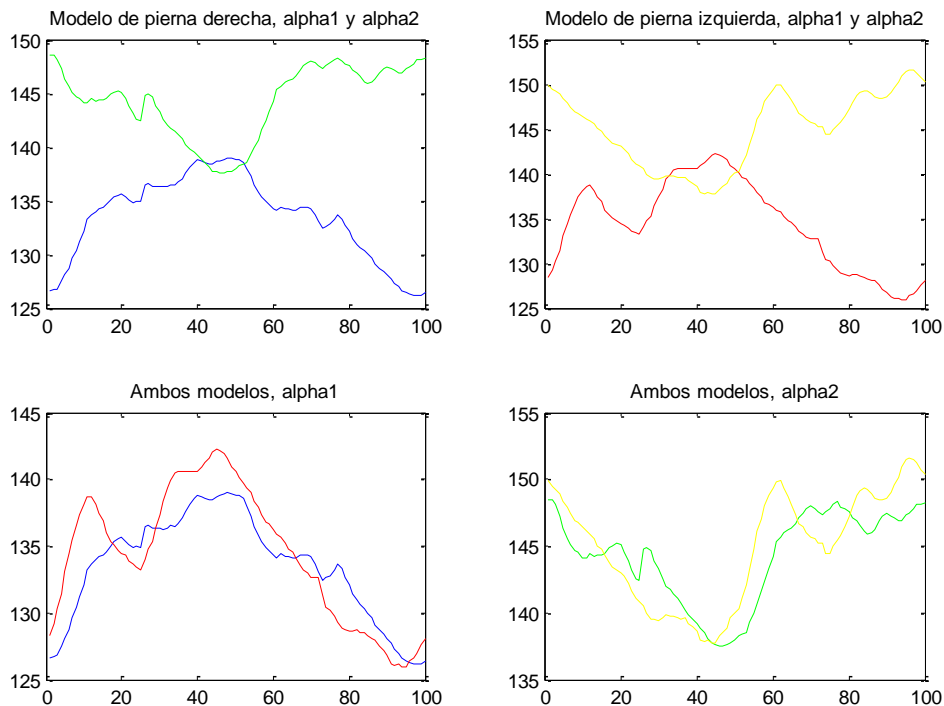


Figura 22: Ángulo alpha (ambos modelos)

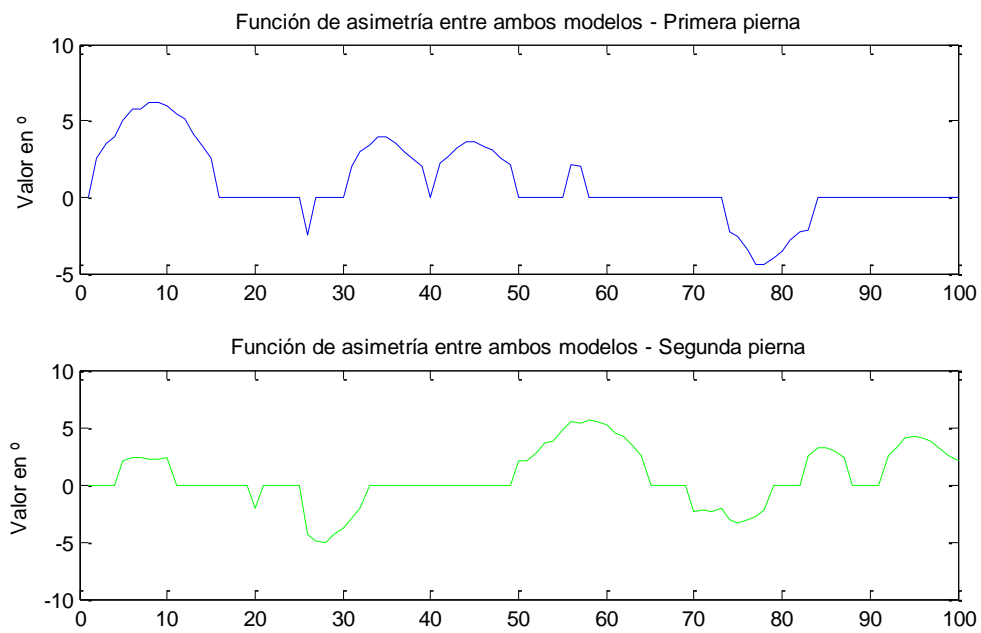


Figura 23: Ángulo alpha – Primera función

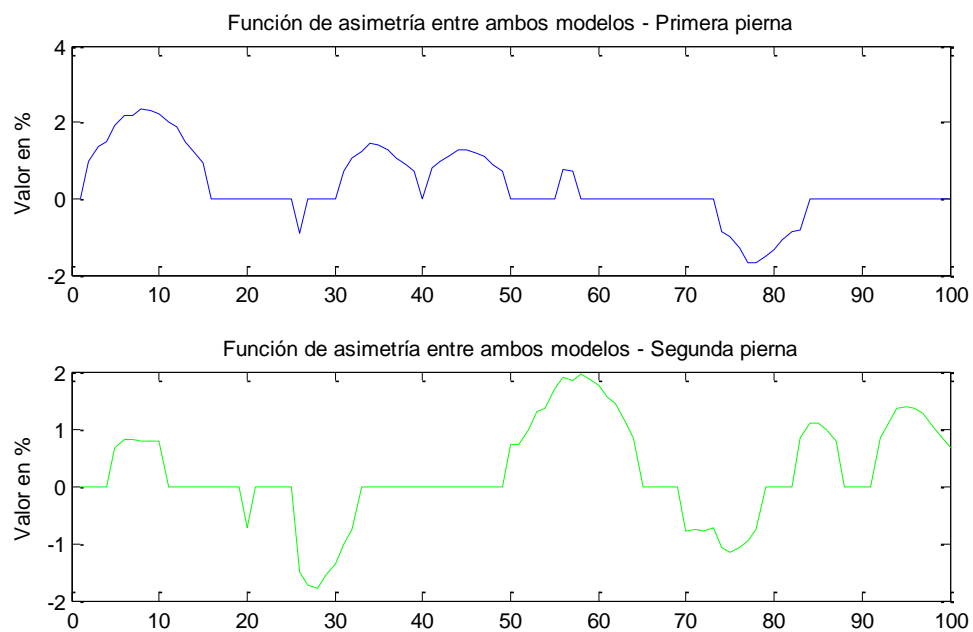


Figura 24: Ángulo alpha – Segunda función.

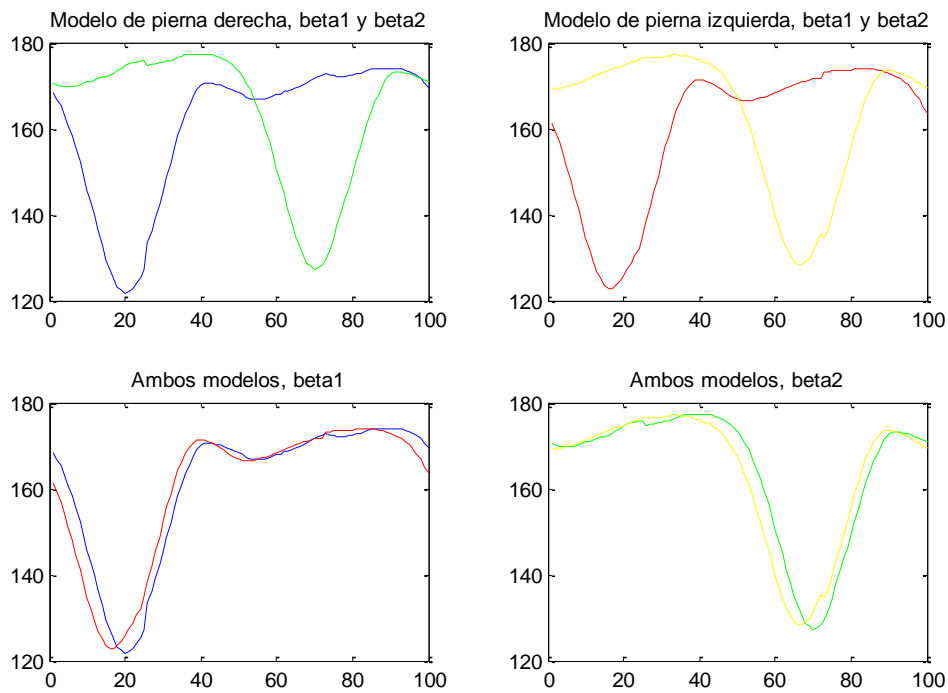


Figura 25: Ángulo beta (ambos modelos)

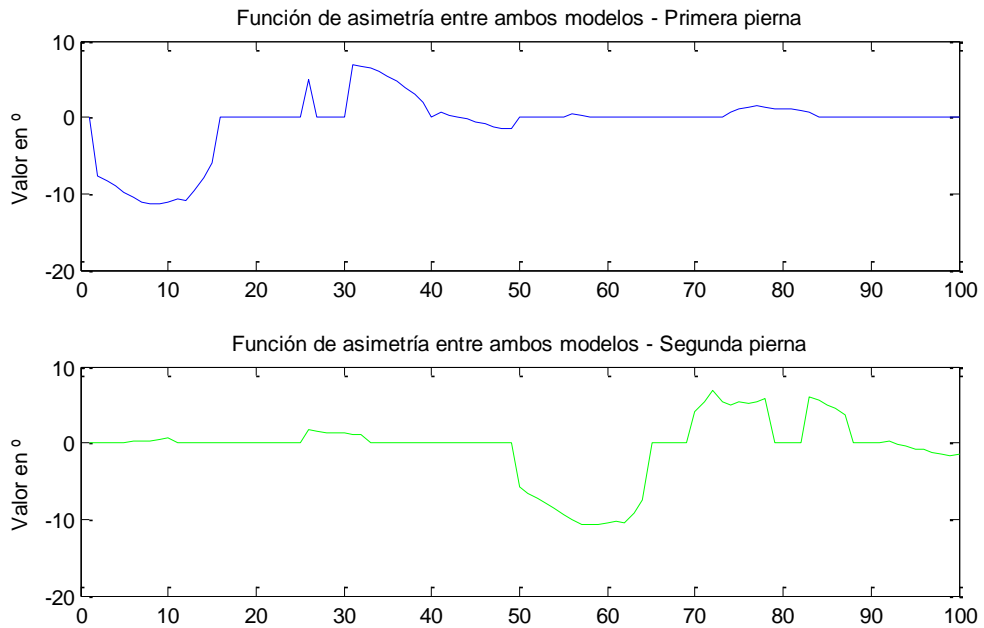


Figura 26: Ángulo beta – Primera función

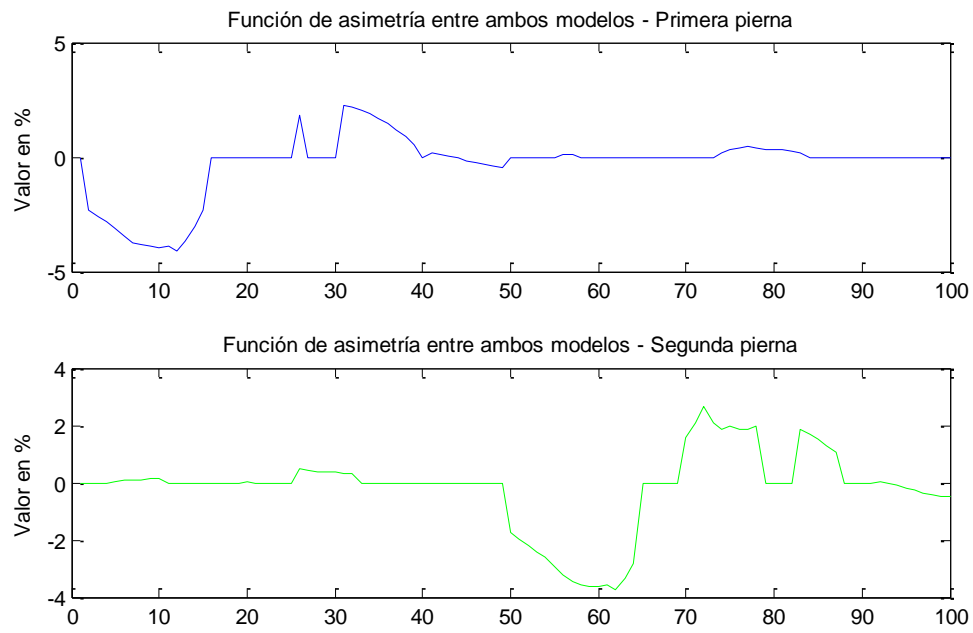


Figura 27: Ángulo beta – Segunda función

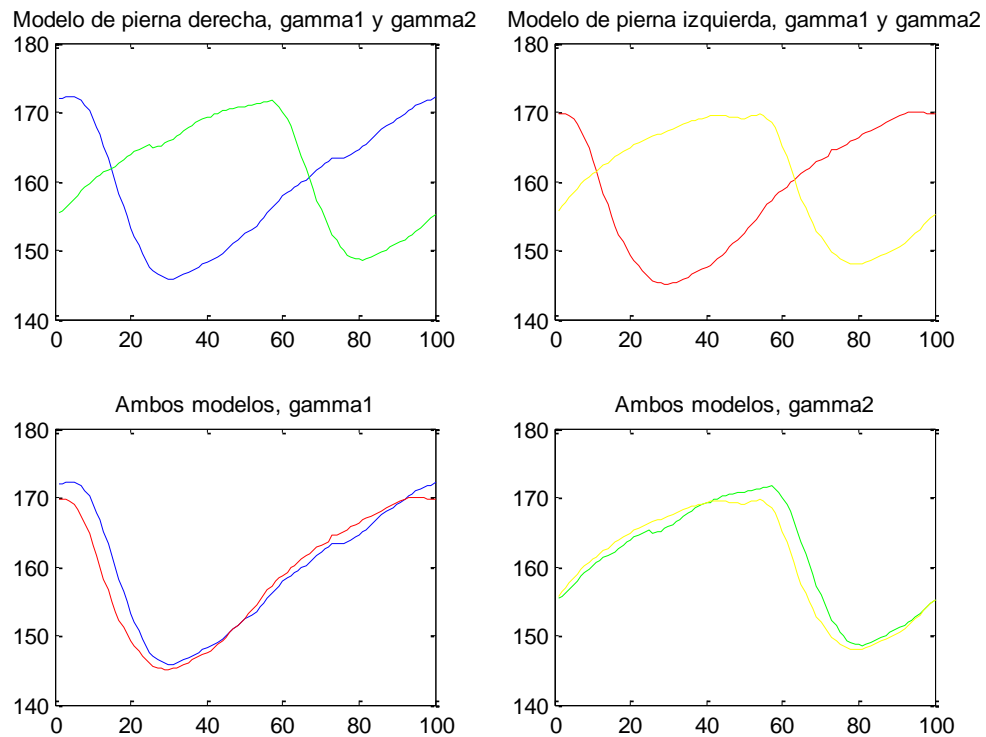


Figura 28: Ángulo gamma (ambos modelos)

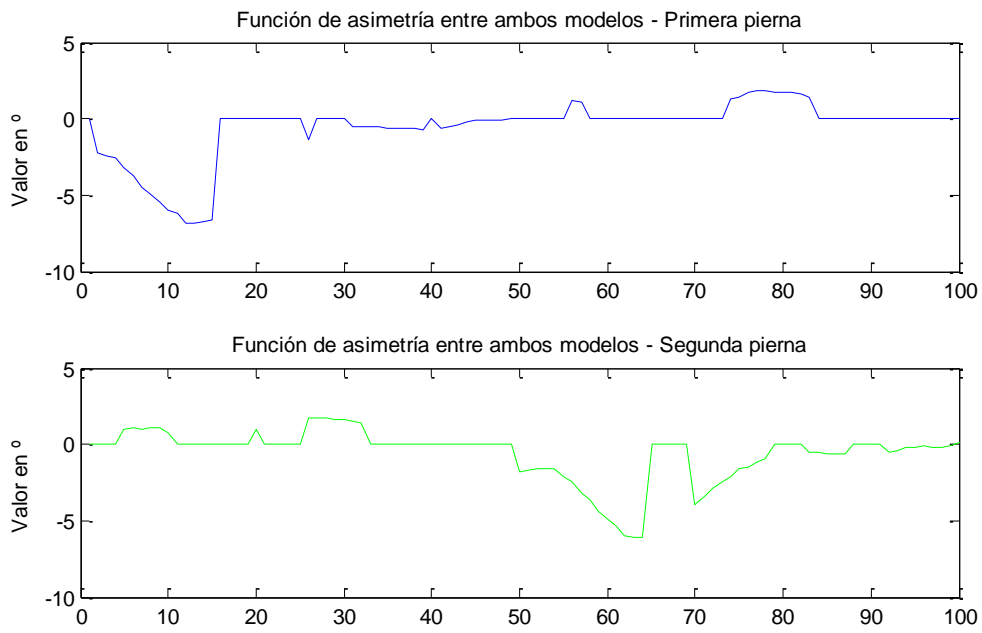


Figura 29: Ángulo gamma – Primera función

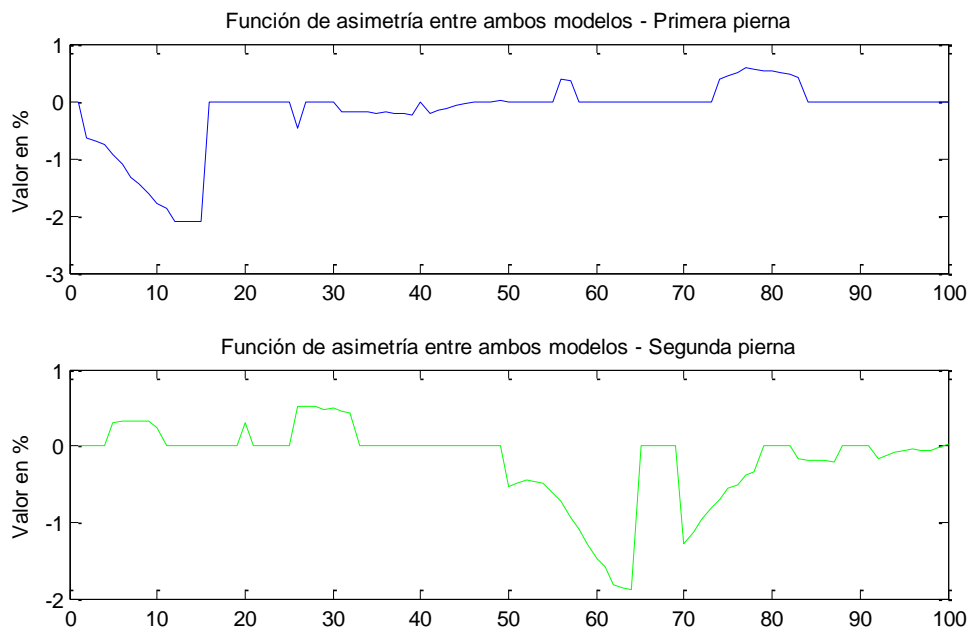


Figura 30: Ángulo gamma – Segunda función

Paciente regular:

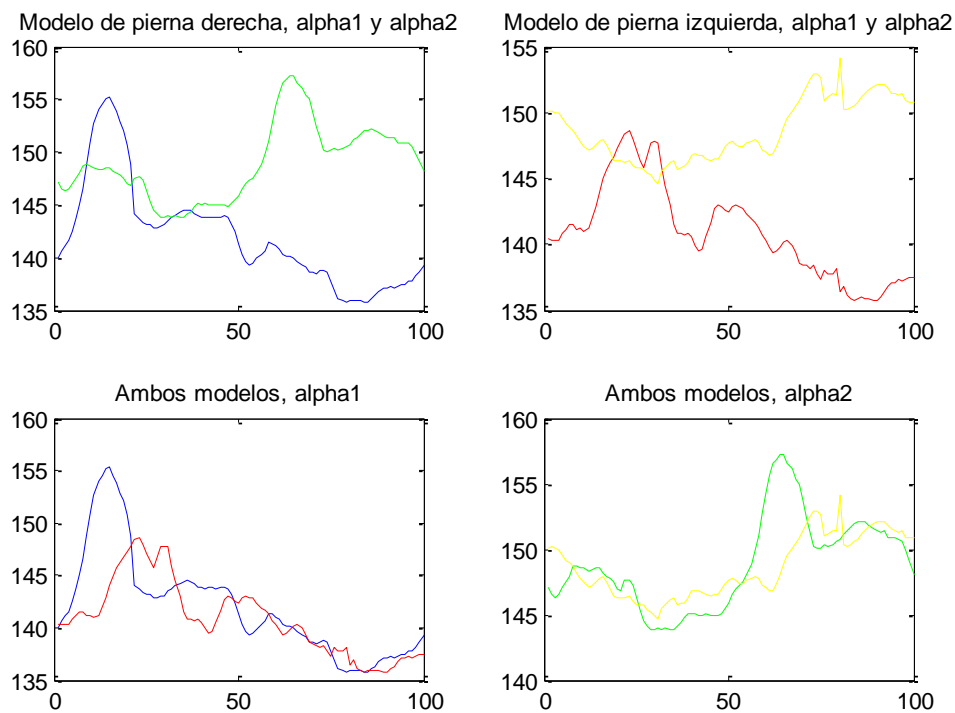


Figura 31: Ángulo α (ambos modelos)

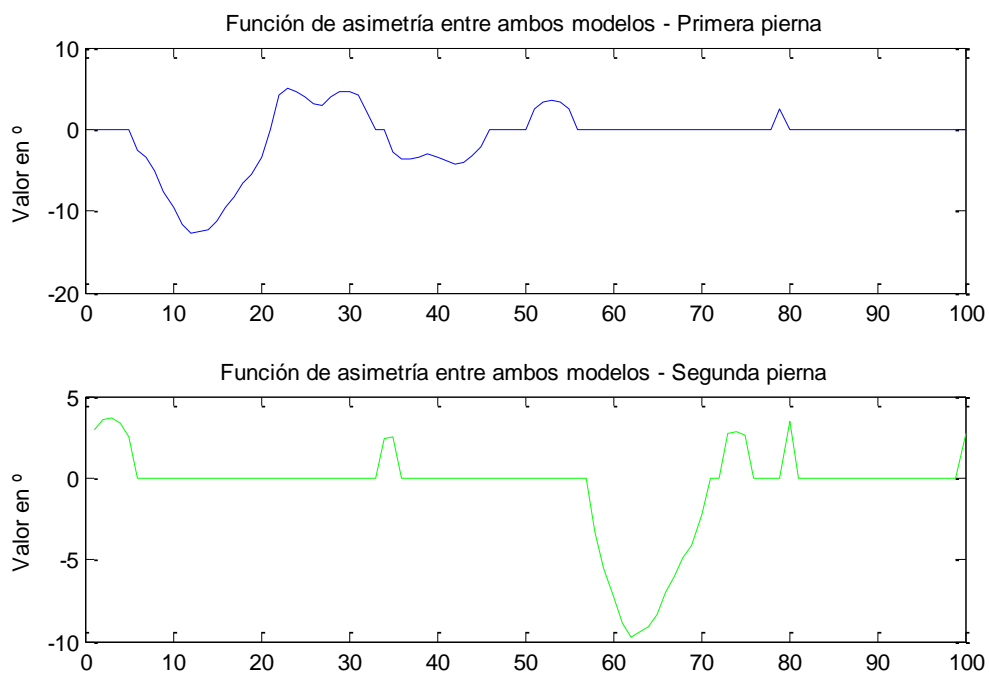


Figura 32: Ángulo α – Primera función

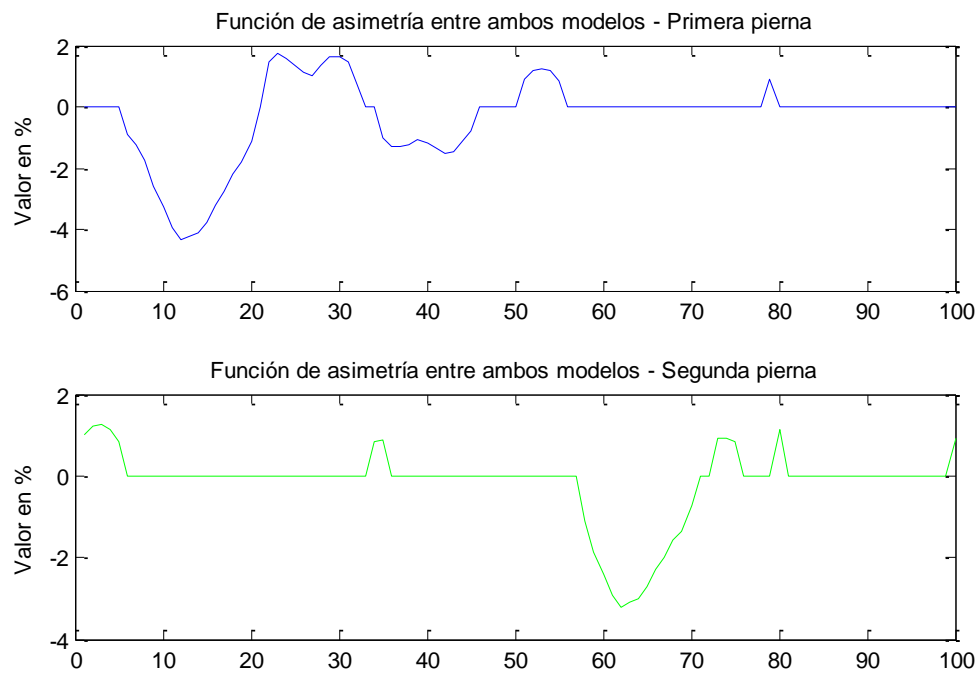


Figura 33: Ángulo alpha – Segunda función

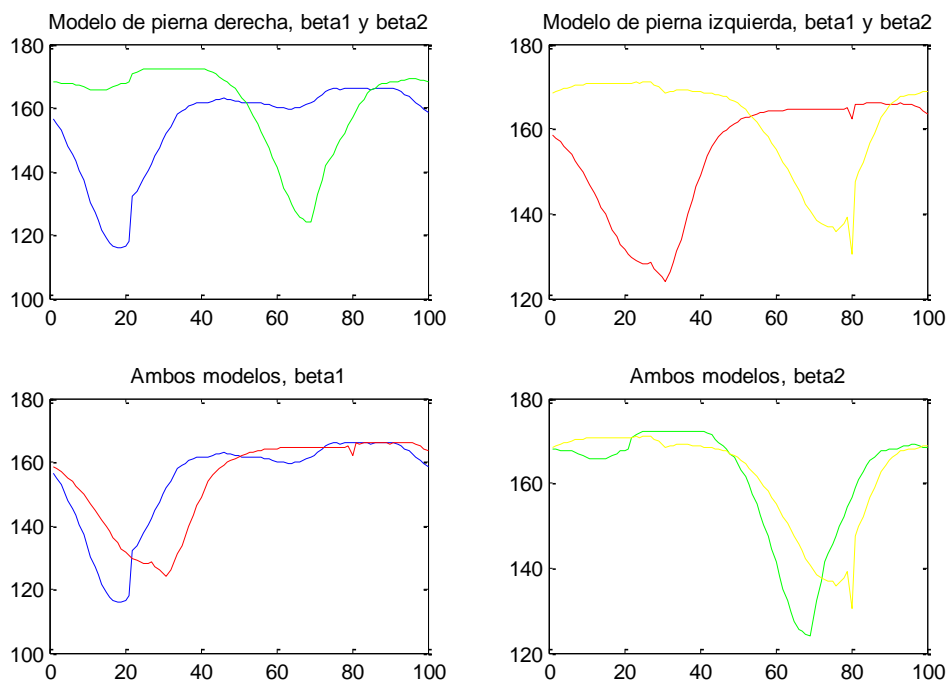


Figura 34: Ángulo beta (ambos modelos)

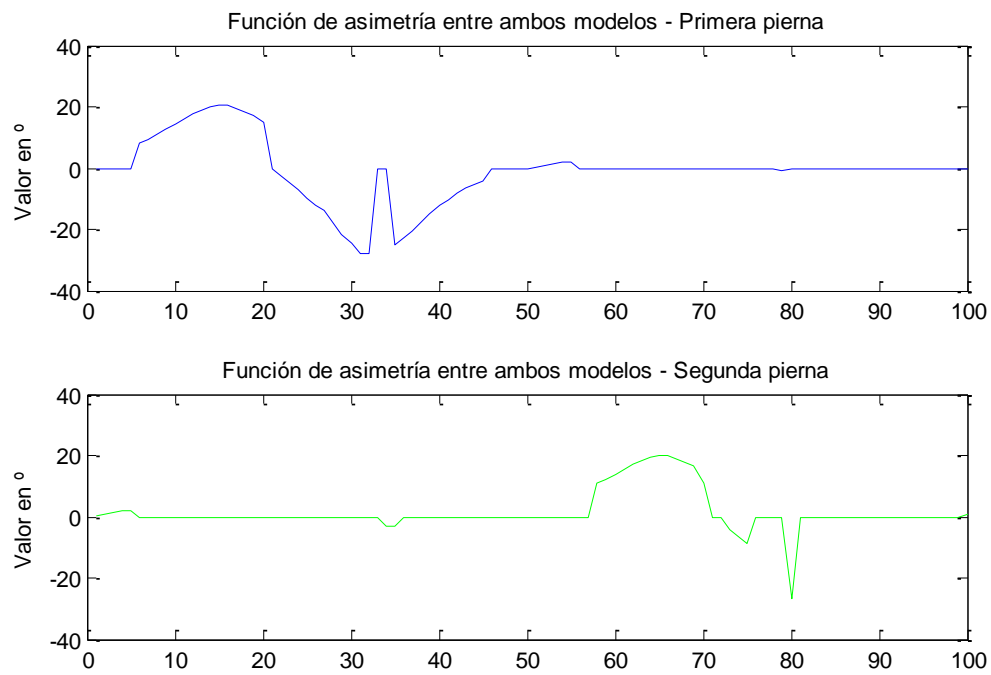


Figura 35: Ángulo beta – Primera función

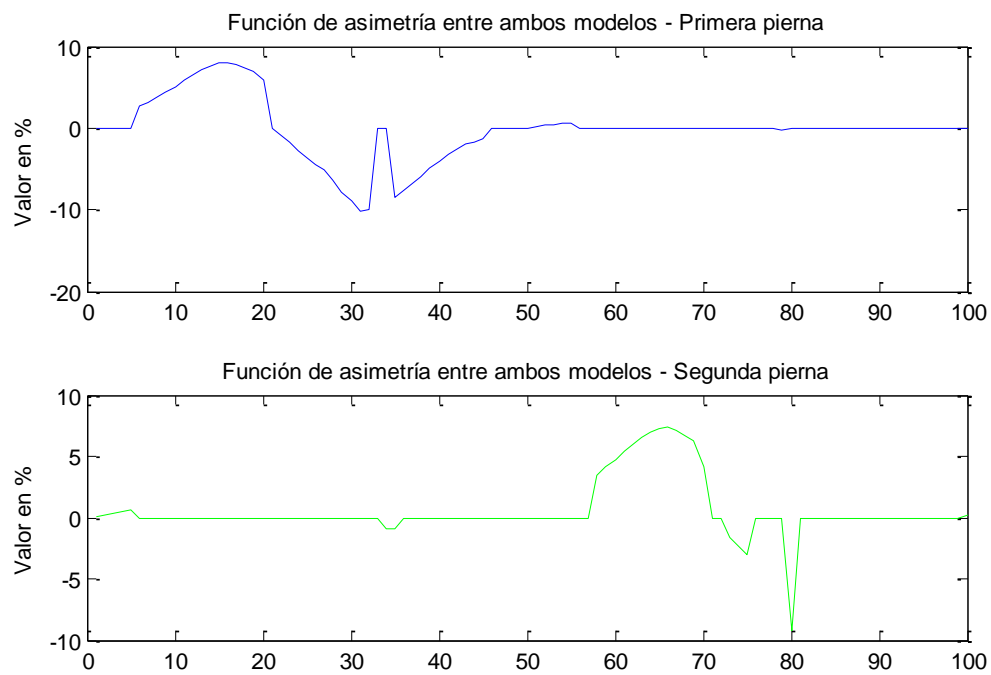


Figura 36: Ángulo beta – Segunda función

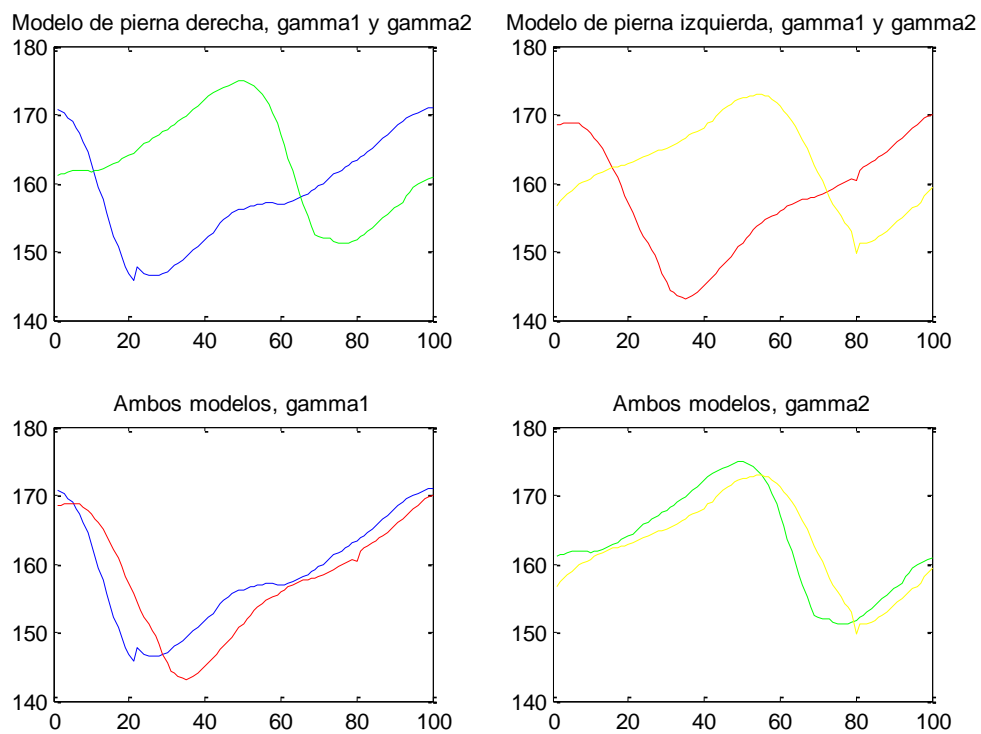


Figura 37: Ángulo gamma (ambas funciones)

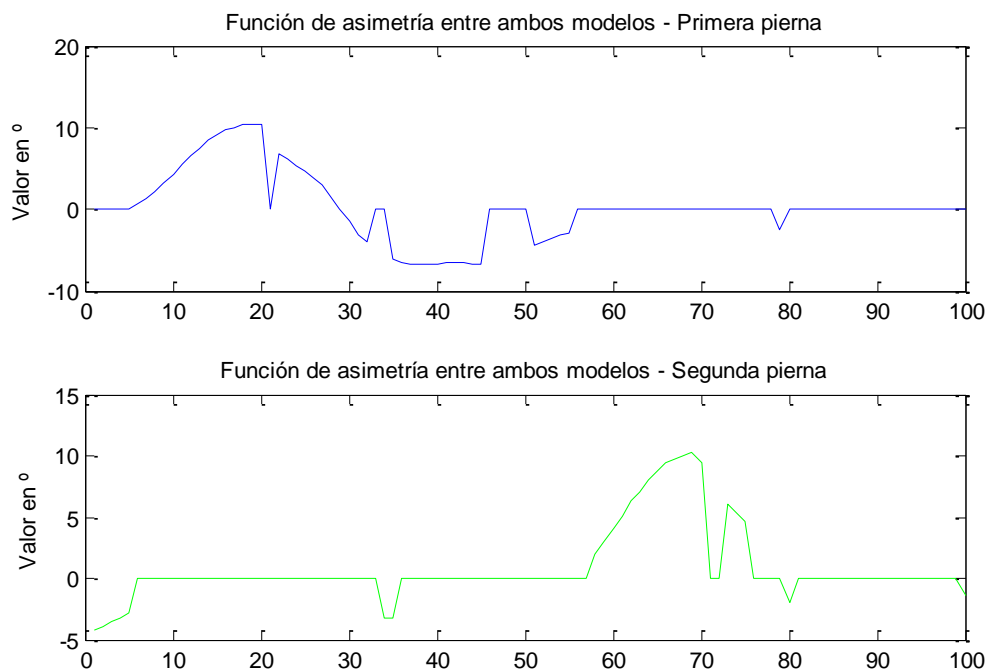


Figura 38: Ángulo gamma – Primera función

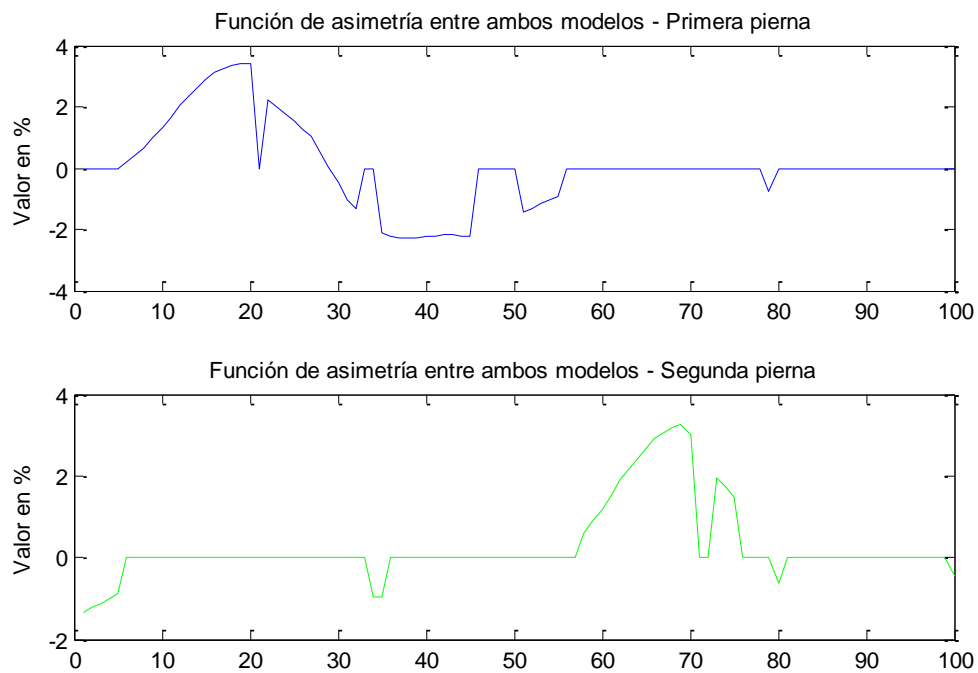


Figura 39: Ángulo gamma – Segunda función

Paciente malo:

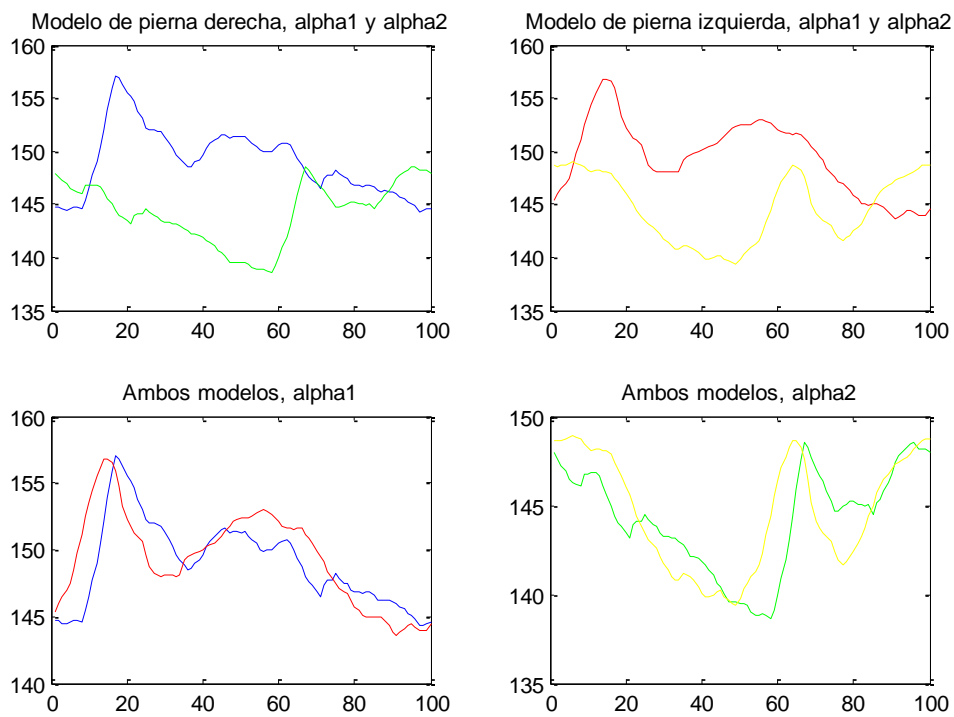


Figura 40: Ángulo alpha (ambos modelos)

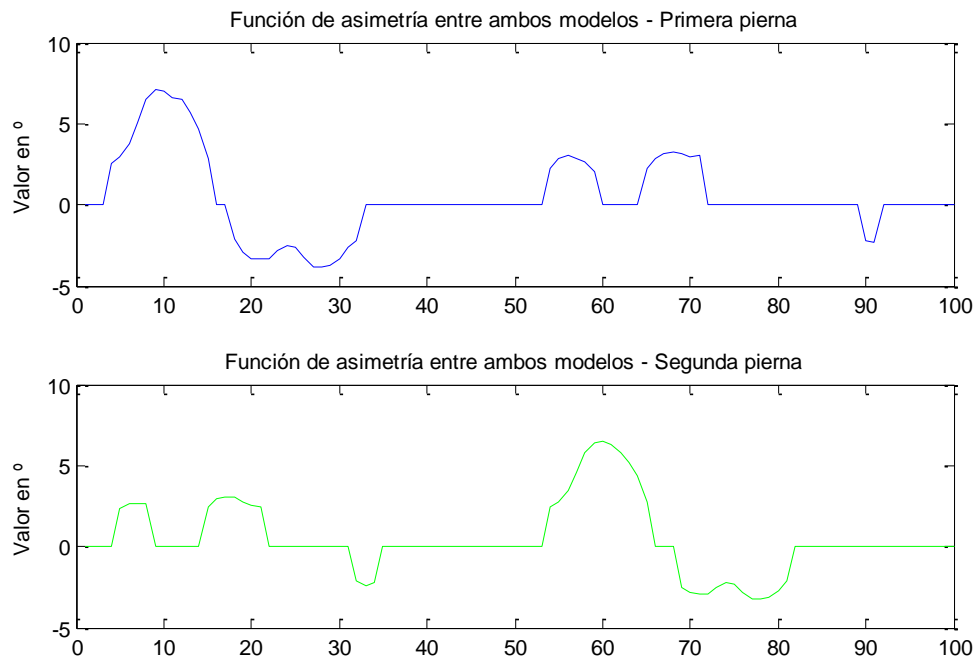


Figura 41: Ángulo alpha – Primera función

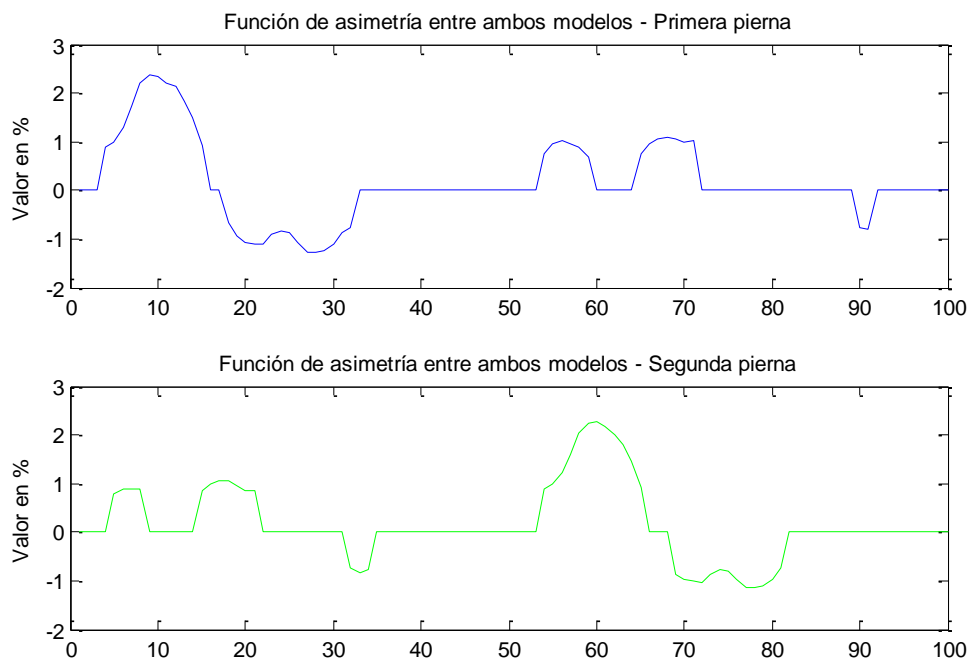


Figura 42: Ángulo alpha – Segunda función

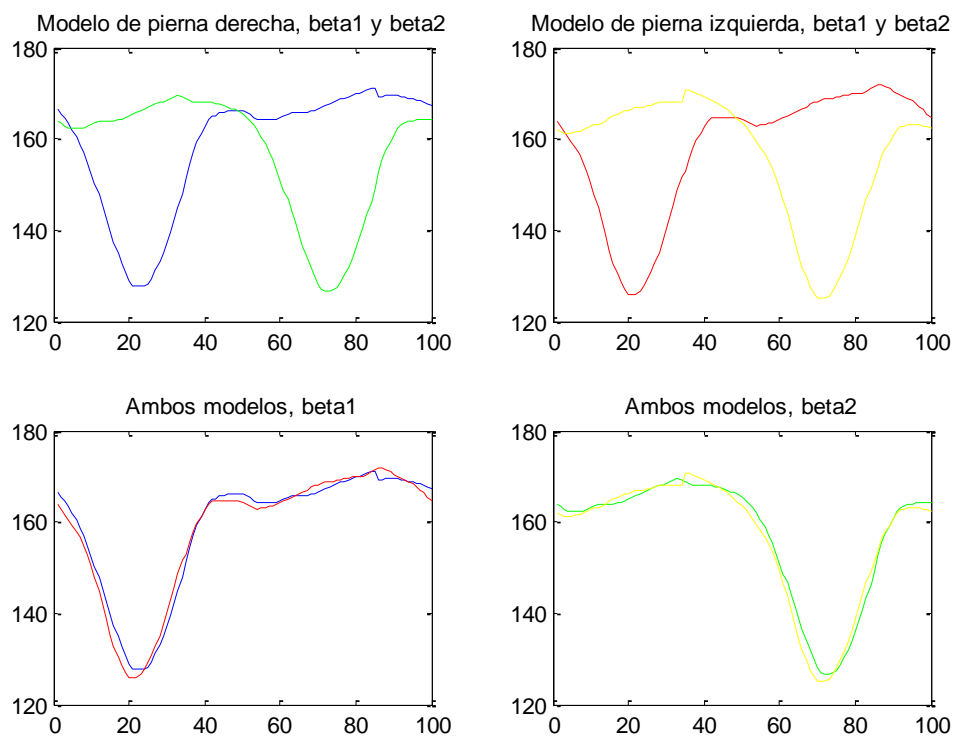


Figura 43: Ángulo beta (ambos modelos)

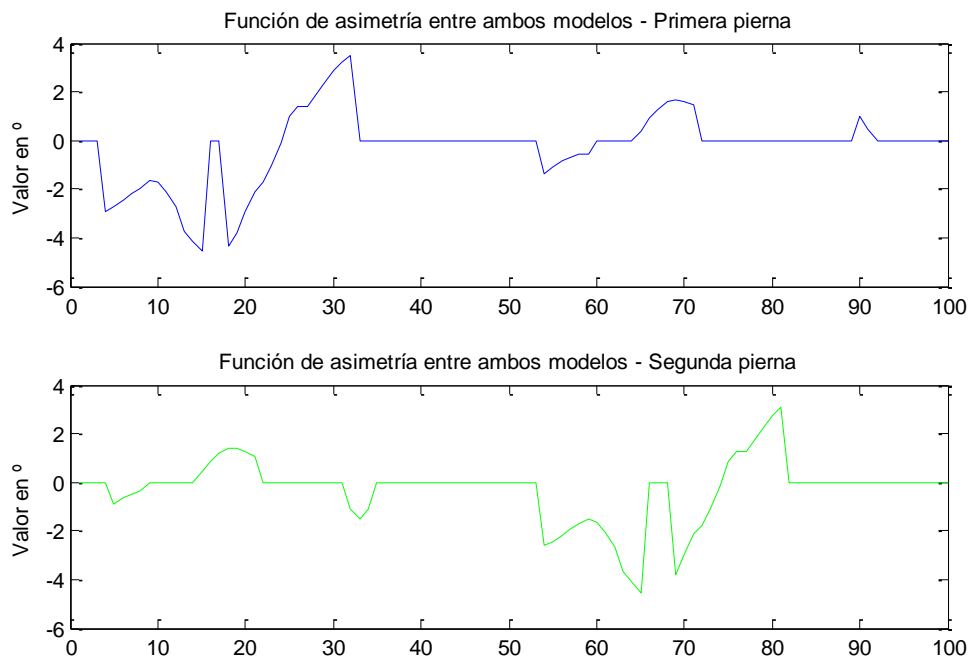


Figura 44: Ángulo beta – Primera función

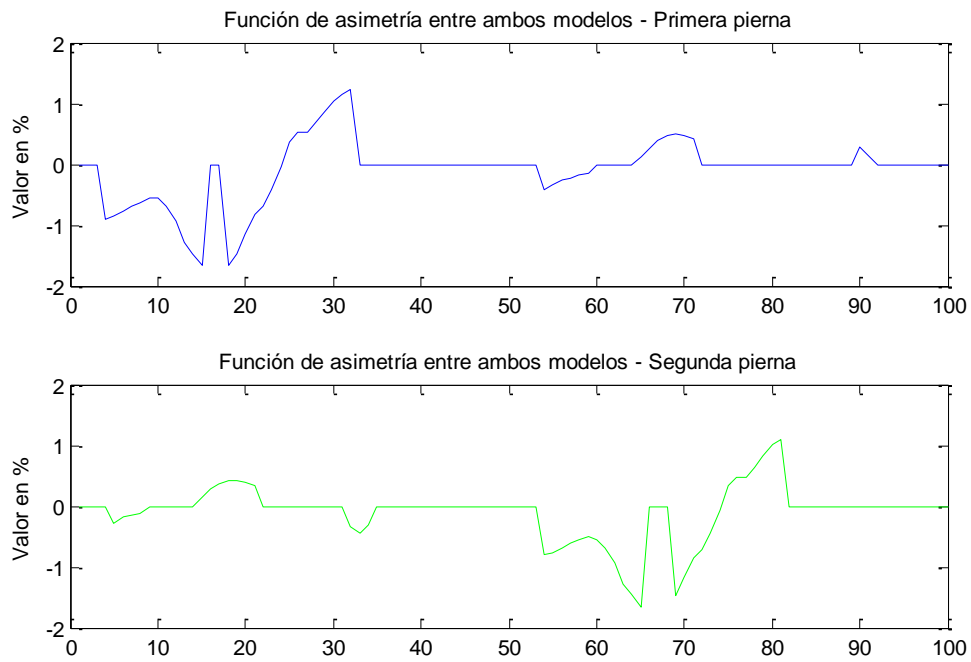


Figura 45: Ángulo beta – Segunda función

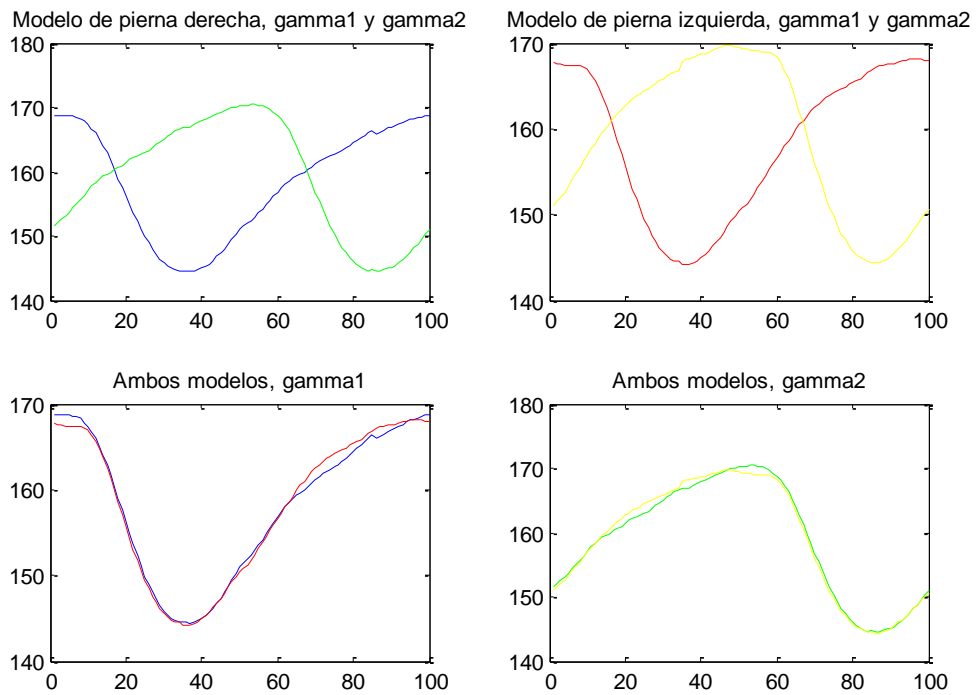


Figura 46: Ángulo gamma (ambos modelos)

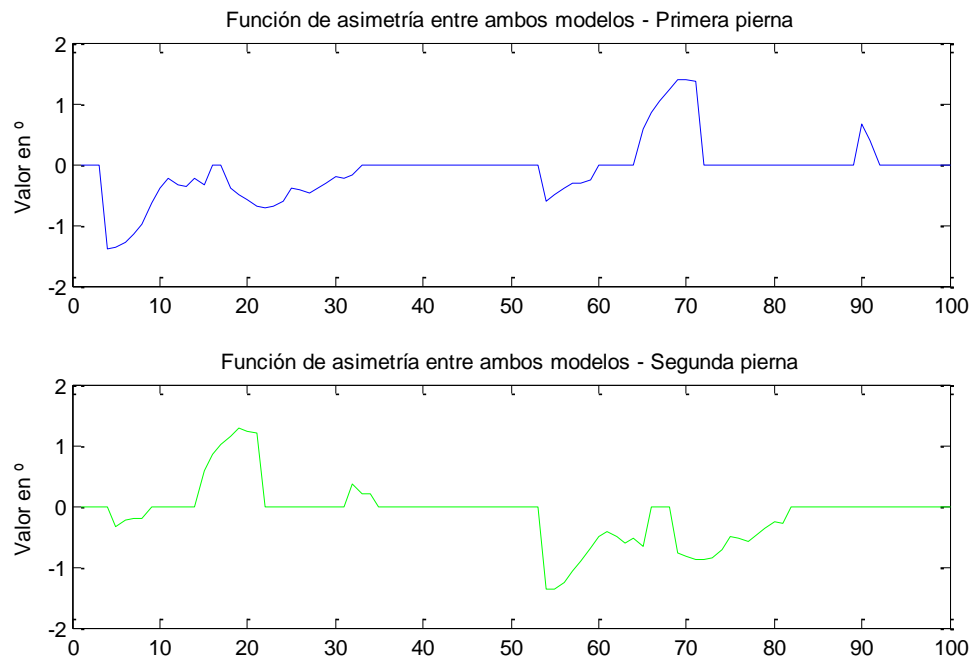


Figura 47: Ángulo gamma – Primera función

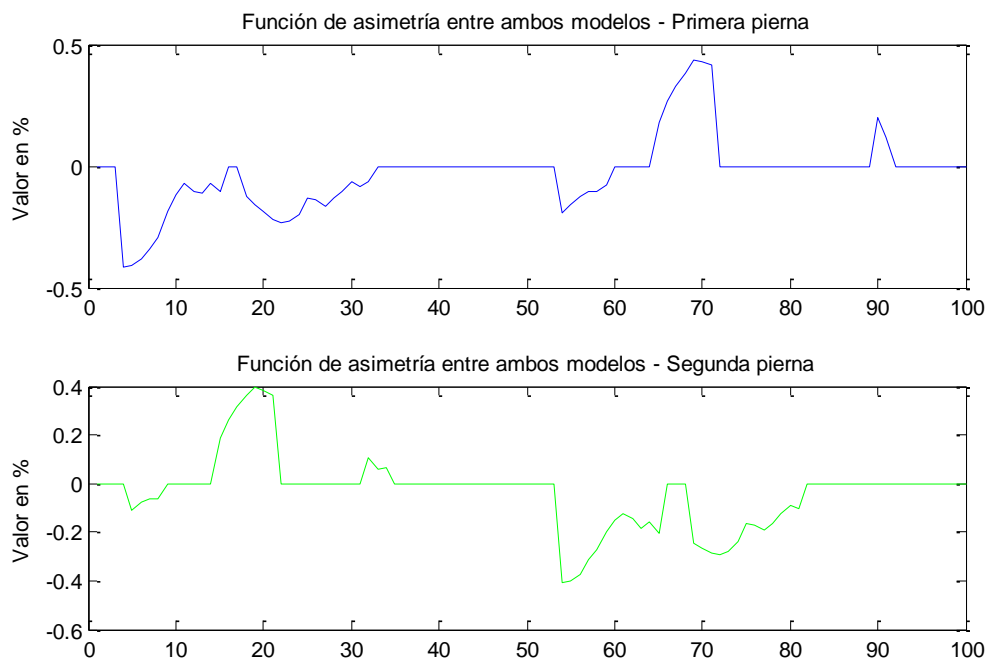


Figura 48: Ángulo gamma – Segunda función

	Paciente bueno	Paciente regular	Paciente malo
Ángulo alpha α	1ª pierna: 2.3° / 0.9 % 2ª pierna: 2.4° / 0.8 %	1ª pierna: 3.8° / 1.3 % 2ª pierna: 2.7° / 0.9 %	1ª pierna: 2.4° / 0.8 % 2ª pierna: 2.2° / 0.7 %
Ángulo beta β	1ª pierna: 3.9° / 1.3 % 2ª pierna: 4.1° / 1.4 %	1ª pierna: 10.0° / 3.6 % 2ª pierna: 6.5° / 2.3 %	1ª pierna: 1.4° / 0.5 % 2ª pierna: 1.3° / 0.4 %
Ángulo gamma γ	1ª pierna: 1.9° / 0.6 % 2ª pierna: 1.7° / 0.5 %	1ª pierna: 3.9° / 1.3 % 2ª pierna: 2.9° / 0.9 %	1ª pierna: 0.5° / 0.1 % 2ª pierna: 0.5° / 0.1 %

Si observamos los valores de los coeficientes de asimetría, comprobamos que el que presenta un menor valor es precisamente el paciente malo y que el que presenta un mayor valor es el paciente regular.

Por tanto, podemos concluir que el criterio utilizado por Asier Sancho Hidalgo no se puede aplicar a nuestro caso. Esto resulta obvio, puesto que lo que se está observando son los puntos de amplitud máxima del paso y de la columna recta, pero nosotros estamos observando distintos ángulos que se producen durante la marcha, por lo que una persona que cumpla el criterio anterior no quiere decir que esté andando de la misma manera con ambas piernas.

Así pues, podemos decir que necesitamos de un criterio más experimental, basado en observar a la persona y a partir de ahí definir unas cotas para cada uno de los distintos ángulos, de tal manera que al sobrepasar estas cotas podamos decir que se está produciendo algún problema en la forma de andar del sujeto.

8.3 Comparación entre distintos pacientes en función del nuevo criterio:

Para comprobar si un paciente presenta mayor o menor asimetría antes de evaluarlo mediante el programa desarrollado para este proyecto, lo más sencillo es simplemente observar la forma de caminar de dicho paciente. Para ello, podemos hacer uso de los ficheros adicionales denominados “grafX.m”, que nos simularía la marcha en

función del paso derecho y el izquierdo simultáneamente. A veces, este método no resultará del todo preciso, puesto que la asimetría tal vez no se termine de apreciar.

Para comprobar este criterio más experimental, se van a comparar dos pacientes que han sido observados. Uno de ellos, aparentemente camina de forma correcta, mientras que el otro presenta unos extraños movimientos en el tobillo (ángulo α).

Ángulo alpha del paciente malo:

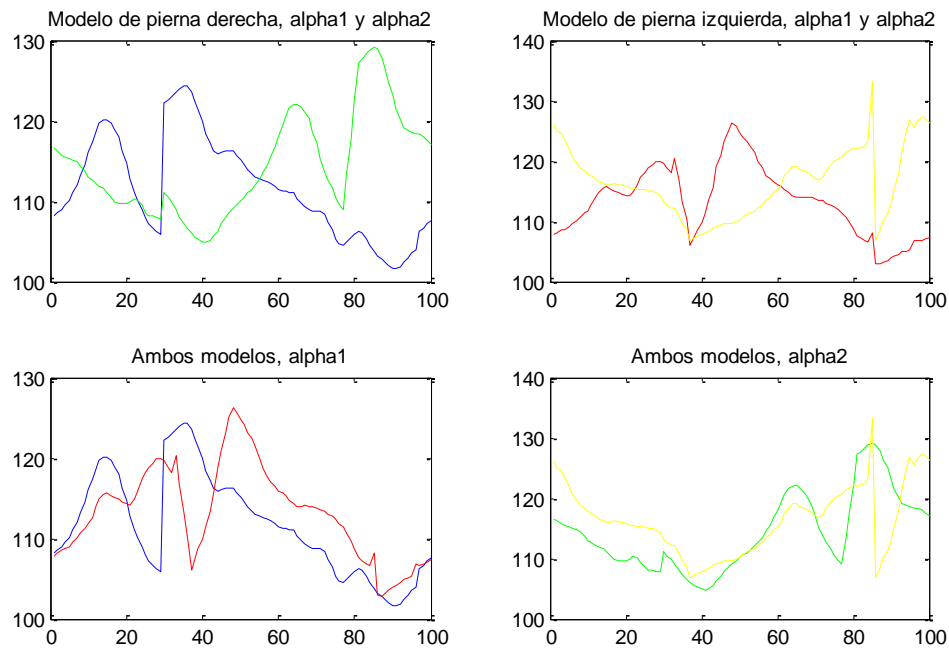


Figura 49: Ángulo alpha (ambos modelos)

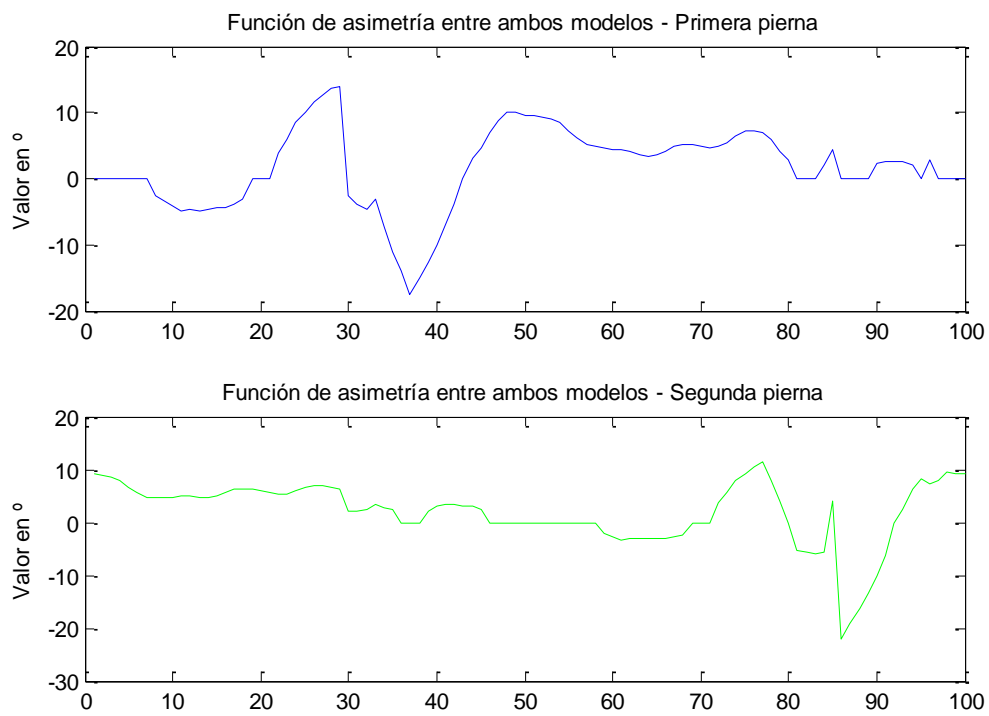


Figura 50: Ángulo alpha – Primera función

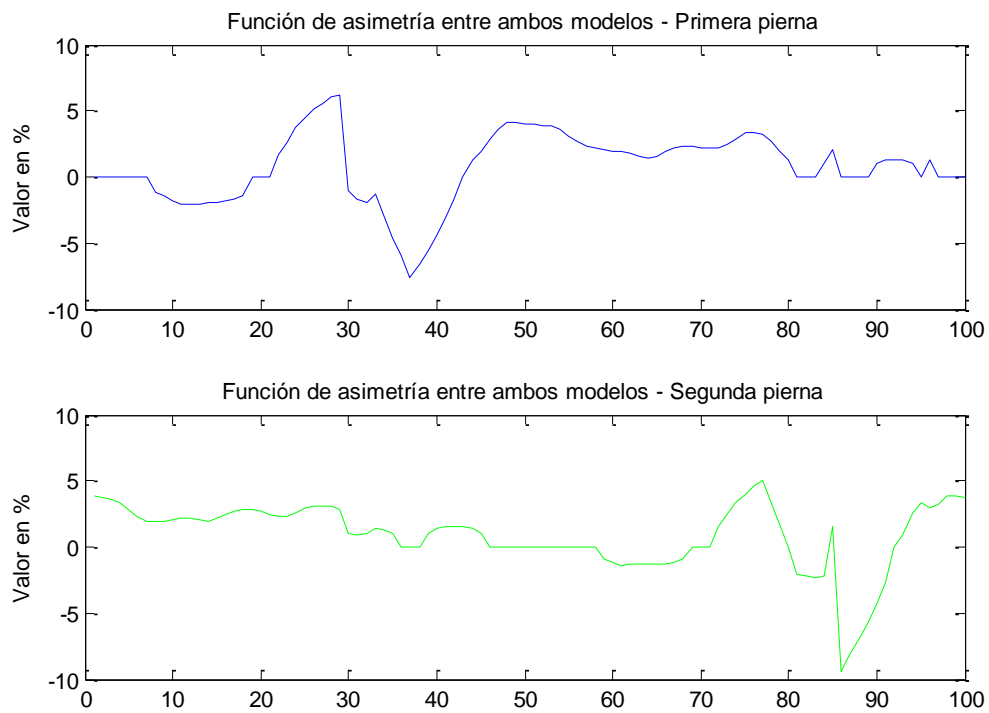


Figura 51: Ángulo alpha – Segunda función

Ángulo alpha del paciente bueno:

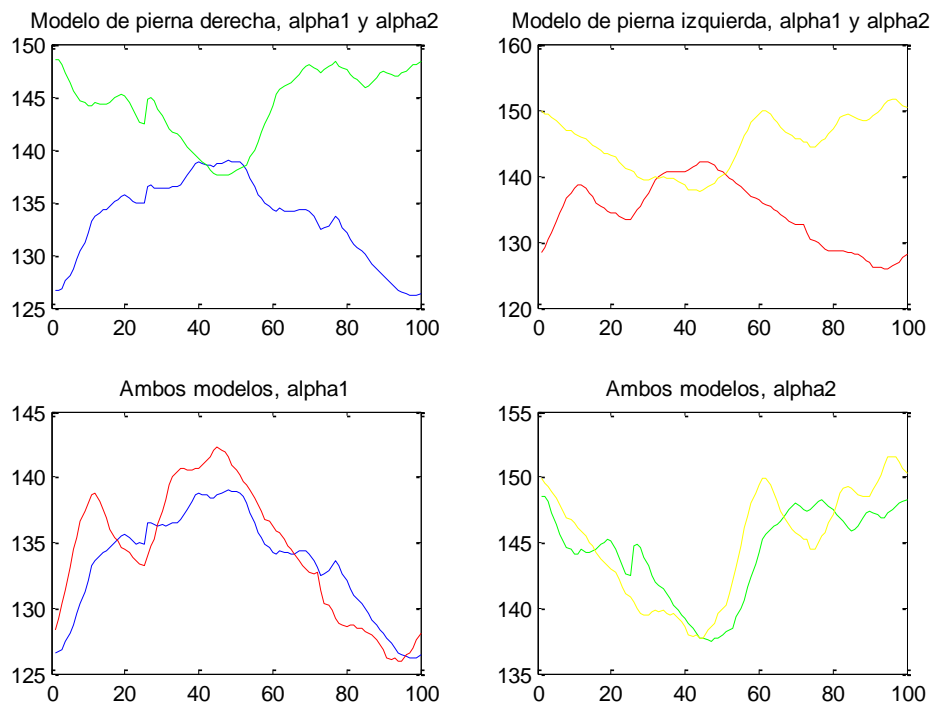


Figura 52: Ángulo alpha (ambos modelos)

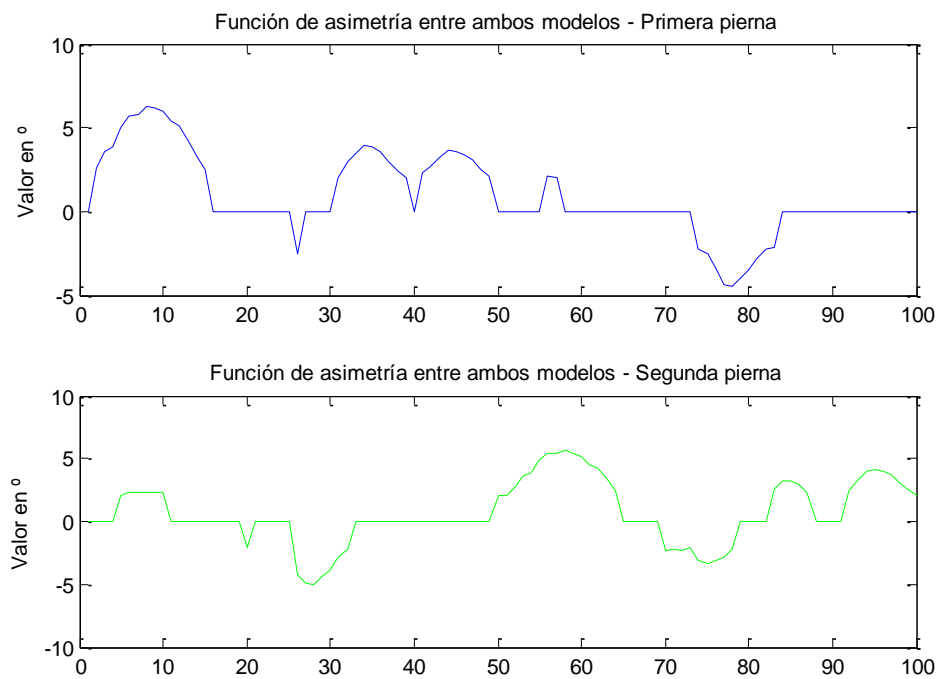


Figura 53: Ángulo alpha – Primera función

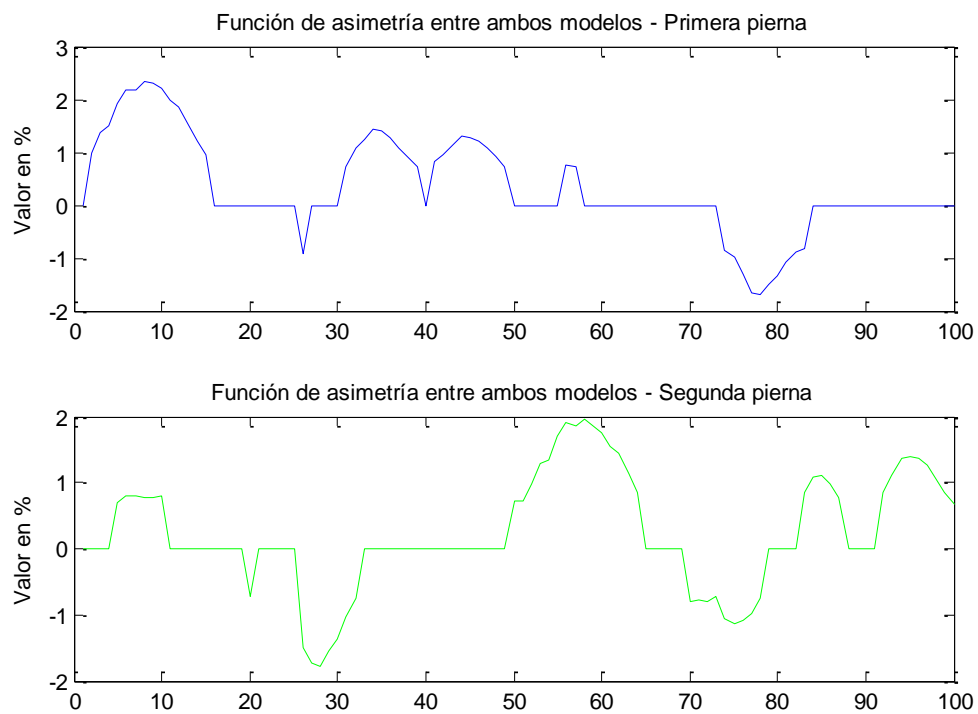


Figura 54: Ángulo alpha – Segunda función

Visualizando las gráficas podemos comprobar que el paciente malo presente unas asimetrías de $18^\circ / 7.5 \%$ en cierto intervalo, frente a los $6^\circ / 2 \%$ máximos que puede presentar el paciente bueno.

Así pues, queda patente que efectivamente la mejor forma de estudiar la asimetría (a falta de tal vez un posible futuro modelo matemático), es simplemente comprobar la función de asimetría.

Es posible que a simple vista, al observar a un paciente, no sean tan visibles ciertas asimetrías de considerable magnitud en ciertos ángulos. Por este motivo, la función de asimetría es de gran utilidad en estas situaciones.

9. CONCLUSIONES:

9.1 Criterio de evaluación de los distintos pacientes:

Como se ha demostrado anteriormente, el criterio elegido por Asier Sancho Hidalgo no es válido para el estudio de la asimetría en los distintos ángulos mediante la marcha humana.

Si observamos el coeficiente de asimetría, podemos analizar a un paciente de forma global. Pese a ello, este criterio tampoco sería válido, puesto que podría ocurrir que la función en algún punto tuviese un gran valor, pero en el resto de puntos apenas presentase asimetría. Por tanto, podría darse el caso de que el coeficiente de asimetría fuese mayor que en un paciente que presenta asimetrías a lo largo del ciclo de paso completo.

Así pues, podemos llegar a la conclusión de que la mejor manera de analizar a un paciente sería observando la función de asimetría a lo largo del tiempo. La representación gráfica es por tanto más útil que los coeficientes de asimetría a la hora de analizar a un paciente.

9.2 División de la función de asimetría en distintas fases:

Conociendo la función de asimetría a lo largo de un paso completo, podríamos dividir dicha función en diversas etapas o fases, de tal manera que pudiésemos diferenciar distintas anomalías en función de dónde se presentan estas asimetrías.

Así pues, si tenemos en cuenta el criterio de Jacquelin Perry^[4], podríamos diferenciar las distintas fases:

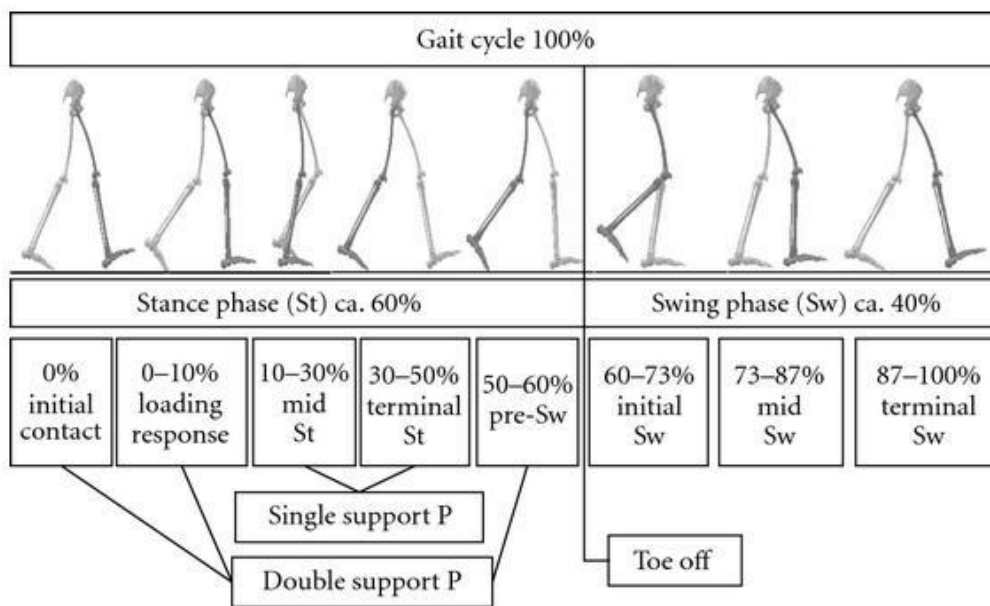


Figura 55: División de la marcha humana en distintas fases según Perry

Lo primero que tendríamos que hacer es sincronizar los datos de nuestro paciente, de tal forma que el paso comience por el punto de contacto inicial. Así pues, tomando los distintos porcentajes que se nos muestran, podríamos dividir nuestra función de asimetría en distintas fases.

En este proyecto presento un ejemplo de dicha división en fases. Para ello, se han usado distintos colores para diferenciar las distintas fases. La distribución de colores queda de la siguiente manera:

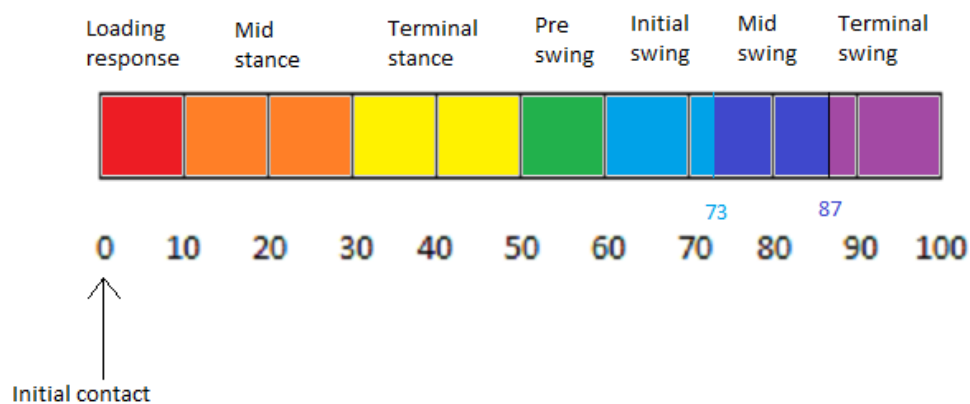


Figura 56: Representación de la marcha humana en bandas de colores

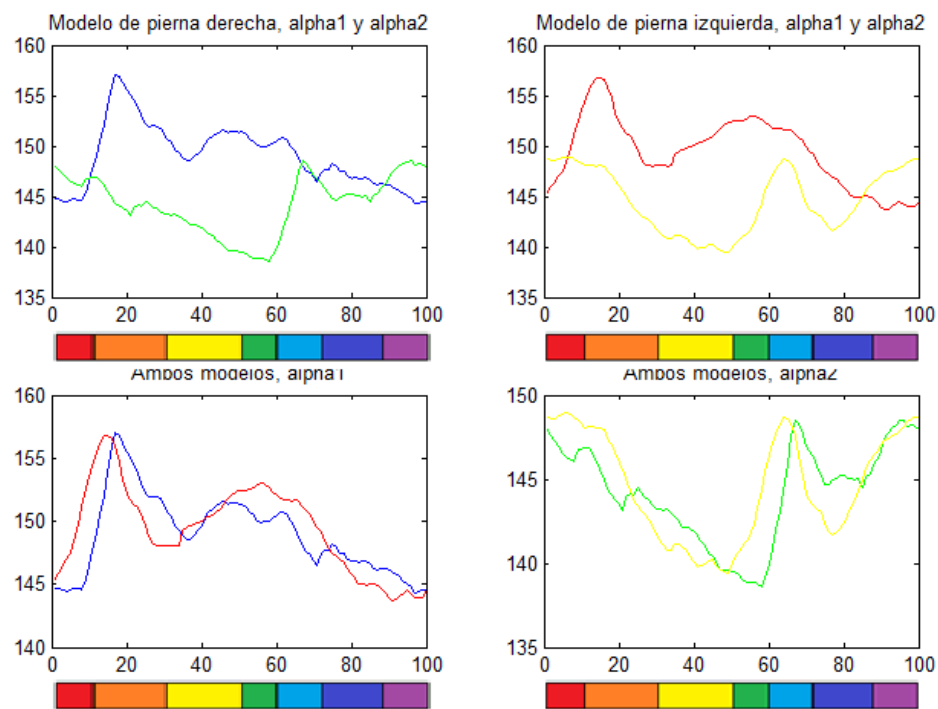


Figura 57: Ángulo α (ambos modelos)

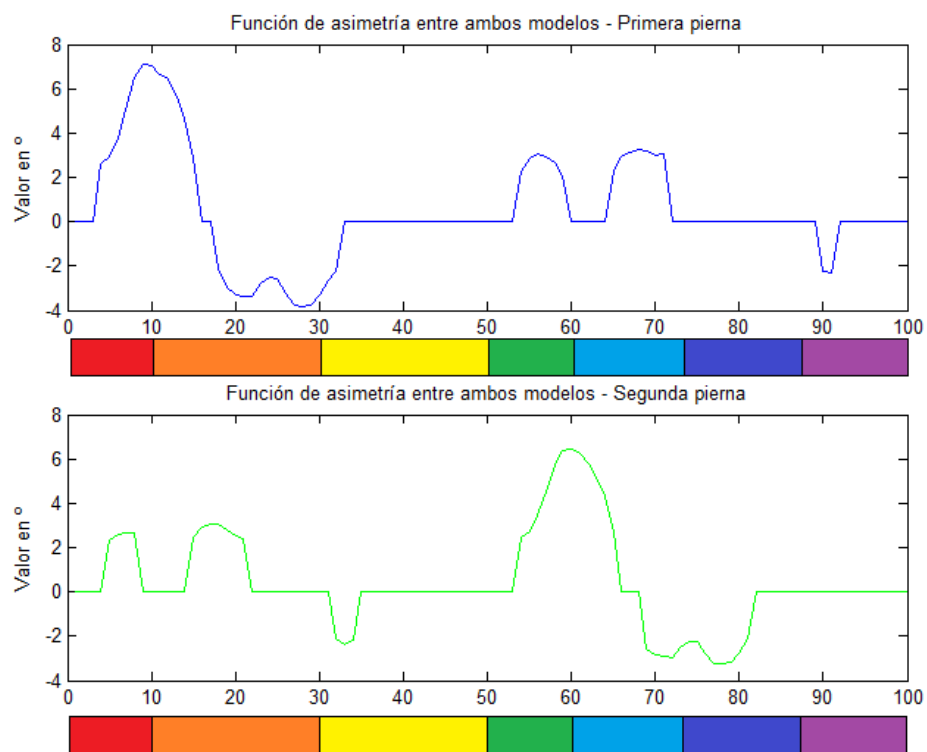


Figura 58: Ángulo α – Primera función

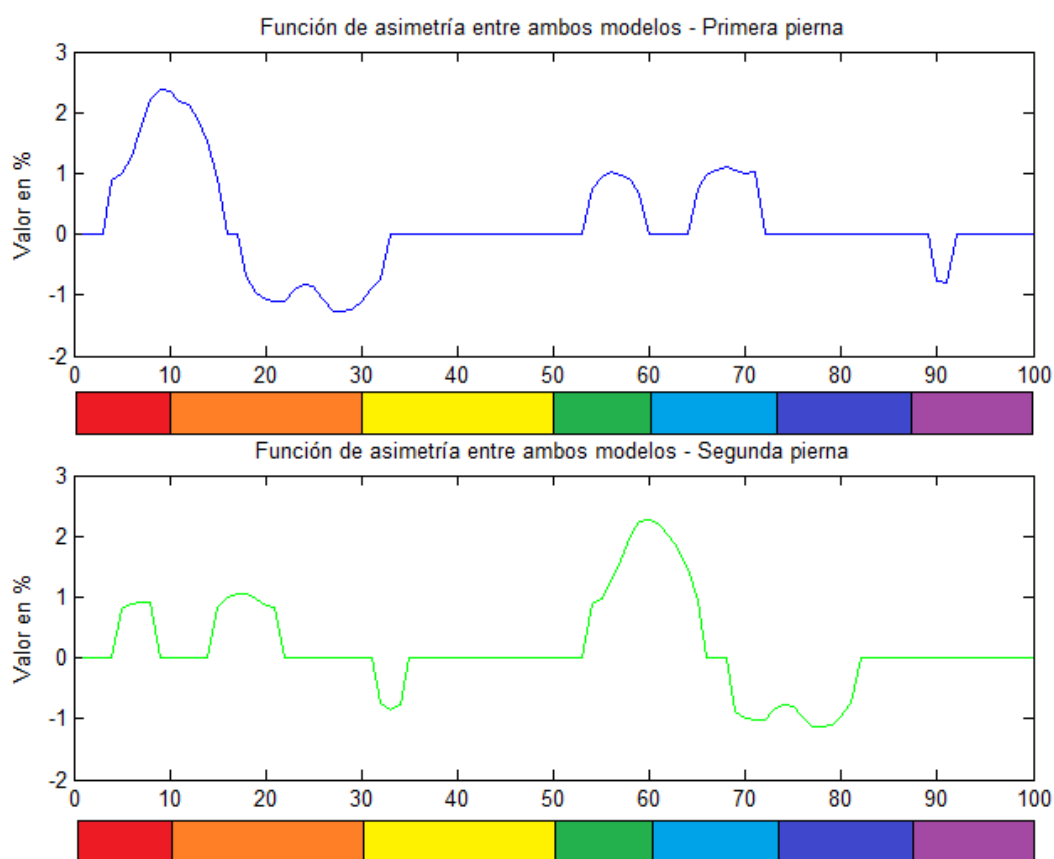


Figura 59: Ángulo alpha – Segunda función

En este ejemplo, si observamos por ejemplo la gráfica de la función correspondiente a la primera pierna, comprobaríamos que existen mayores asimetrías en las fases de “loading response” y “mid stance”, mientras que en las fases de “terminal stance” y “mid swing” no hay directamente asimetría.

Igualmente, la existencia de estas asimetrías en dichas fases de la marcha, no implicarían una anomalía. Por ello sería necesario el estudio intensivo de dichas fases para los distintos ángulos en numerosos pacientes y relacionar ciertos valores con diversas anomalías.

10. LÍNEAS FUTURAS:

A partir de las herramientas desarrolladas para este proyecto y de las conclusiones obtenidas, se podría trabajar en este campo a partir de este punto.

Por un lado, se podrían analizar diferentes pacientes y estudiar distintas patologías en función de las asimetrías presentadas en las diversas etapas.

De forma similar, podría adaptarse el contenido de este proyecto para el estudio de otros ángulos durante la marcha humana, de tal manera que se podrían analizar distintas patologías que no se han tenido en cuenta para este proyecto.

Otra posibilidad es simplemente definir nuevas funciones de asimetría, en función de distintos criterios como puede ser la sensibilidad.

Analizando distintos pacientes, también se podrían desarrollar unas cotas de asimetría, de tal manera que se pudiese diferenciar si un paciente presenta o no patología en función de las asimetrías en un ángulo y a su vez diferenciando entre las distintas fases según el criterio de Jacquelin Perry.

11. BIBLIOGRAFÍA:

[1] Asier Sancho Hidalgo: Estudiante de la Universidad de Valencia. Desarrollador del programa necesario para obtener los modelos de paso derecho y paso izquierdo.

[2] Matlab: Herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Es muy utilizado para el estudio de señales en el mundo de las telecomunicaciones.

[3] Laboratorio SATI: Grupo de investigación en el campo de “Sistemas de Adquisición y Tratamiento de Imágenes” en Valencia.

[4] Jacquelin Perry: Escritora del libro Gait Analysis: Normal and Pathological Function, en el que se describen las diversas fases en la marcha humana según avanzados estudios en este campo.

- Información proporcionada por el profesor Jacek Dusza.
- Presentación PowerPoint “Functional Analysis of Vertebral Spine” de Stereovision.
- Libro “Gait Analysis: Normal and Pathological Function” de Jacquelin Perry.
- Ficheros de Matlab que componen el proyecto de Asier Sanchez Hidalgo.
- www.draw.io
- www.wikipedia.com
- www.mathworks.com
- www.lawebdelprogramador.com

ANEXO I: Códigos del programa

Código de Principal:

```
%PRINCIPAL

%Carga de los ficheros
disp ('Seleccione el fichero con los datos referentes al movimiento de
pierna derecha')
disp (' ')
carga=input('Cargar archivo de entrada: ','s');
der=load(carga);

disp ('Seleccione el fichero con los datos referentes al movimiento de
pierna izquierda')
disp (' ')
carga=input('Cargar archivo de entrada: ','s');
izq=load(carga);

%Llamada al menu
Menu;
```

Código de Menu:

```
%MENU PRINCIPAL

%Seleccion de opciones entre distintos angulos
correcto = 0;
while (correcto == 0)

    disp(' ')
    disp('MENU')
    disp(' ')
    disp('1.- Angulo alpha')
    disp(' ')
    disp('2.- Angulo beta')
    disp(' ')
    disp('3.- Angulo gamma')
    disp(' ')
    disp('4.- Seleccionar nuevo paciente')
    disp(' ')
    disp('0.- Salir')

    opcion = input('Seleccione una opcion: ');

    switch opcion

        case 1,
            anguloAlpha;
            correcto = 1;
        case 2,
            anguloBeta;
            correcto = 1;
```

```

        case 3,
            anguloGamma;
            correcto = 1;
        case 4,
            Principal;
            clear all;
            correcto = 1;
        case 0,
            correcto = 1;
        otherwise,
            disp('No ha seleccionado una opcion correcta. Vuelva a
intentarlo')

    end

end

```

Código de anguloAlpha:

```

%ANGULO ALPHA

%Para el modelo de paso derecho calculamos los puntos de las dos
piernas
P6_der = der(:,16:18);
P7_der = der(:,19:21);
P8_der = der(:,22:24);
V76_der = P6_der - P7_der;
V78_der = P8_der - P7_der;

P24_der = der(:,70:72);
P25_der = der(:,73:75);
P26_der = der(:,76:78);
V2524_der = P24_der - P25_der;
V2625_der = P26_der - P25_der;

%Para el modelo de paso derecho calculamos los valores del angulo
for i=1:1:100

    coalpha_der1(i)=(V76_der(i,1)*V78_der(i,1)+V76_der(i,2)*V78_der(i,2)+V
76_der(i,3)*V78_der(i,3))/(sqrt(V76_der(i,1)^2+V76_der(i,2)^2+V76_der(
i,3)^2)*sqrt(V78_der(i,1)^2+V78_der(i,2)^2+V78_der(i,3)^2));
    alpha_der1(i)=acos(coalpha_der1(i))*(180/pi);
end

for j=1:1:100

    coalpha_der2(j)=(V2524_der(j,1)*V2625_der(j,1)+V2524_der(j,2)*V2625_de
r(j,2)+V2524_der(j,3)*V2625_der(j,3))/(sqrt(V2524_der(j,1)^2+V2524_der
(j,2)^2+V2524_der(j,3)^2)*sqrt(V2625_der(j,1)^2+V2625_der(j,2)^2+V2625
_der(j,3)^2));
    alpha_der2(j)=acos(coalpha_der2(j))*(180/pi);
end

%Para el modelo de paso izquierdo calculamos los puntos de las dos
piernas

```



```

P6_izq = izq(:,16:18);
P7_izq = izq(:,19:21);
P8_izq = izq(:,22:24);
V76_izq = P6_izq - P7_izq;
V78_izq = P8_izq - P7_izq;

P24_izq = izq(:,70:72);
P25_izq = izq(:,73:75);
P26_izq = izq(:,76:78);
V2524_izq = P24_izq - P25_izq;
V2625_izq = P26_izq - P25_izq;

%Para el modelo de paso izquierdo calculamos los valores del angulo
for a=1:1:100

coalpha_izq1(a)=(V76_izq(a,1)*V78_izq(a,1)+V76_izq(a,2)*V78_izq(a,2)+V
76_izq(a,3)*V78_izq(a,3))/(sqrt(V76_izq(a,1)^2+V76_izq(a,2)^2+V76_izq(
a,3)^2)*sqrt(V78_izq(a,1)^2+V78_izq(a,2)^2+V78_izq(a,3)^2));
alpha_izq1(a)=acos(coalpha_izq1(a))*(180/pi);
end

for b=1:1:100

coalpha_izq2(b)=(V2524_izq(b,1)*V2625_izq(b,1)+V2524_izq(b,2)*V2625_iz
q(b,2)+V2524_izq(b,3)*V2625_izq(b,3))/(sqrt(V2524_izq(b,1)^2+V2524_izq
(b,2)^2+V2524_izq(b,3)^2)*sqrt(V2625_izq(b,1)^2+V2625_izq(b,2)^2+V2625
_izq(b,3)^2));
alpha_izq2(b)=acos(coalpha_izq2(b))*(180/pi);
end

%Representamos los valores del angulo para ambos modelos
figure(1)
subplot(2,2,1);
plot(alpha_der1,'b');
hold on;
plot(alpha_der2,'g');
title('Modelo de pierna derecha, alpha1 y alpha2');

subplot(2,2,2);
plot(alpha_izq1,'r');
hold on;
plot(alpha_izq2,'y');
title('Modelo de pierna izquierda, alpha1 y alpha2');

subplot(2,2,3);
plot(alpha_der1,'b');
hold on;
plot(alpha_izq1,'r');
hold on;
title('Ambos modelos, alpha1');

subplot(2,2,4);
plot(alpha_der2,'g');
hold on;
plot(alpha_izq2,'y');
title('Ambos modelos, alpha2');

```

```

%Calculamos la funcion de asimetria

mostrar = 1;

while (mostrar == 1)

    opcion = input('Seleccione la función de asimetría que desea
    calcular (de 1 a 6): ');

    switch opcion

        case 1,
            for i=1:1:100
                if(abs(alpha_izq1(i)-alpha_der1(i))<2)
                    coas1(i)=0;
                else
                    coas1(i)=(alpha_izq1(i)-alpha_der1(i));
                end
            end

            for i=1:1:100
                if(abs(alpha_izq2(i)-alpha_der2(i))<2)
                    coas2(i)=0;
                else
                    coas2(i)=(alpha_izq2(i)-alpha_der2(i));
                end
            end

        case 2,
            for i=1:1:100
                if(abs(alpha_izq1(i)-alpha_der1(i))<2)
                    coas1(i)=0;
                else
                    coas1(i)=(alpha_izq1(i)-
alpha_der1(i))/(alpha_izq1(i)+alpha_der1(i));
                    coas1(i)=coas1(i)*100;
                end
            end

            for i=1:1:100
                if(abs(alpha_izq2(i)-alpha_der2(i))<2)
                    coas2(i)=0;
                else
                    coas2(i)=(alpha_izq2(i)-
alpha_der2(i))/(alpha_izq2(i)+alpha_der2(i));
                    coas2(i)=coas2(i)*100;
                end
            end

        case 3,
            for i=1:1:100
                if(abs(alpha_izq1(i)-alpha_der1(i))<2)
                    coas1(i)=0;
                else
                    coas1(i)=abs((alpha_der1(i)/alpha_izq1(i))-
(alpha_izq1(i)/alpha_der1(i)));
                    coas1(i)=coas1(i)*100;
                end
            end
    end
end

```

```

        end
    end

    for i=1:1:100
        if(abs(alpha_izq2(i)-alpha_der2(i))<2)
            coas2(i)=0;
        else
            coas2(i)=abs((alpha_der2(i)/alpha_der2(i))-
(alpha_izq2(i)/alpha_der2(i)));
            coas2(i)=coas2(i)*100;
        end
    end

    case 4,
        for i=1:1:100
            if(abs(alpha_izq1(i)-alpha_der1(i))<2)
                coas1(i)=0;
            else
coas1(i)=(alpha_der1(i)/alpha_izq1(i))+(alpha_izq1(i)/alpha_der1(i))-
2;
                coas1(i)=coas1(i)*100;
            end
        end

        for i=1:1:100
            if(abs(alpha_izq2(i)-alpha_der2(i))<2)
                coas2(i)=0;
            else
coas2(i)=(alpha_der2(i)/alpha_izq2(i))+(alpha_izq2(i)/alpha_der2(i))-
2;
                coas2(i)=coas2(i)*100;
            end
        end

    case 5,
        for i=1:1:100
            if(abs(alpha_izq1(i)-alpha_der1(i))<2)
                coas1(i)=0;
            else
coas1(i)=(alpha_der1(i)-
alpha_izq1(i))/max(alpha_der1(i),alpha_izq1(i));
                coas1(i)=coas1(i)*100;
            end
        end

        for i=1:1:100
            if(abs(alpha_izq2(i)-alpha_der2(i))<2)
                coas2(i)=0;
            else
coas2(i)=(alpha_der2(i)-
alpha_izq2(i))/max(alpha_der2(i),alpha_izq2(i));
                coas2(i)=coas2(i)*100;
            end
        end

    case 6,

```

```

        for i=1:1:100
            if(abs(alpha_izq1(i)-alpha_der1(i))<2)
                coas1(i)=0;
            else
                coas1(i)=abs((alpha_der1(i)-
alpha_izq1(i))/max(alpha_der1(i),alpha_izq1(i)));
                coas1(i)=coas1(i)*100;
            end
        end

        for i=1:1:100
            if(abs(alpha_izq2(i)-alpha_der2(i))<2)
                coas2(i)=0;
            else
                coas2(i)=abs((alpha_der2(i)-
alpha_izq2(i))/max(alpha_der2(i),alpha_izq2(i)));
                coas2(i)=coas2(i)*100;
            end
        end

        case 0,
            mostrar = 0;
            Menu;

        otherwise,
            disp('Opción incorrecta ')
            for i=1:1:100
                coas1(i)=0;
                coas2(i)=0;
            end

        end

        %Calculamos el coeficiente de asimetria
        media1=mean(coas1);
        media2=mean(coas2);
        for i=1:1:100
            resultado1(i)=(coas1(i)-media1)^2;
            resultado2(i)=(coas2(i)-media2)^2;
        end

        coeficiente1=sqrt(mean(resultado1));
        coeficiente2=sqrt(mean(resultado2));

        %Representamos el coeficiente de asimetria
        disp(' ');
        disp('El coeficiente de asimetría para la primera pierna es:
');disp(coeficiente1);
        disp(' ');
        disp('El coeficiente de asimetría para la segunda pierna es:
');disp(coeficiente2);

        %Representamos la funcion de asimetria
        if (opcion==1)
            figure(4)
            subplot(2,1,1);
            plot(coas1,'b');

```

```

        title('Función de asimetría entre ambos modelos - Primera
pierna')
        ylabel('Valor en °');

        subplot(2,1,2);
        plot(coas2,'g');
        title('Función de asimetría entre ambos modelos - Segunda
pierna');
        ylabel('Valor en °');
    else
        figure(4)
        subplot(2,1,1);
        plot(coas1,'b');
        title('Función de asimetría entre ambos modelos - Primera
pierna')
        ylabel('Valor en %');

        subplot(2,1,2);
        plot(coas2,'g');
        title('Función de asimetría entre ambos modelos - Segunda
pierna');
        ylabel('Valor en %');
    end

end

end

```

Código de anguloBeta:

```

%ANGULO ALPHA

%Para el modelo de paso derecho calculamos los puntos de las dos
piernas
P6_der = der(:,16:18);
P7_der = der(:,19:21);
P8_der = der(:,22:24);
V76_der = P6_der - P7_der;
V78_der = P8_der - P7_der;

P24_der = der(:,70:72);
P25_der = der(:,73:75);
P26_der = der(:,76:78);
V2524_der = P24_der - P25_der;
V2625_der = P26_der - P25_der;

%Para el modelo de paso derecho calculamos los valores del angulo
for i=1:1:100

    coalpha_der1(i)=(V76_der(i,1)*V78_der(i,1)+V76_der(i,2)*V78_der(i,2)+V
76_der(i,3)*V78_der(i,3))/(sqrt(V76_der(i,1)^2+V76_der(i,2)^2+V76_der(
i,3)^2)*sqrt(V78_der(i,1)^2+V78_der(i,2)^2+V78_der(i,3)^2));
    alpha_der1(i)=acos(coalpha_der1(i))*(180/pi);
end

for j=1:1:100

```

```

coalpha_der2(j)=(V2524_der(j,1)*V2625_der(j,1)+V2524_der(j,2)*V2625_der(j,2)+V2524_der(j,3)*V2625_der(j,3))/(sqrt(V2524_der(j,1)^2+V2524_der(j,2)^2+V2524_der(j,3)^2)*sqrt(V2625_der(j,1)^2+V2625_der(j,2)^2+V2625_der(j,3)^2));
alpha_der2(j)=acos(coalpha_der2(j))*(180/pi);
end

%Para el modelo de paso izquierdo calculamos los puntos de las dos
piernas
P6_izq = izq(:,16:18);
P7_izq = izq(:,19:21);
P8_izq = izq(:,22:24);
V76_izq = P6_izq - P7_izq;
V78_izq = P8_izq - P7_izq;

P24_izq = izq(:,70:72);
P25_izq = izq(:,73:75);
P26_izq= izq(:,76:78);
V2524_izq = P24_izq - P25_izq;
V2625_izq = P26_izq - P25_izq;

%Para el modelo de paso izquierdo calculamos los valores del angulo
for a=1:1:100

coalpha_izq1(a)=(V76_izq(a,1)*V78_izq(a,1)+V76_izq(a,2)*V78_izq(a,2)+V76_izq(a,3)*V78_izq(a,3))/(sqrt(V76_izq(a,1)^2+V76_izq(a,2)^2+V76_izq(a,3)^2)*sqrt(V78_izq(a,1)^2+V78_izq(a,2)^2+V78_izq(a,3)^2));
alpha_izq1(a)=acos(coalpha_izq1(a))*(180/pi);
end

for b=1:1:100

coalpha_izq2(b)=(V2524_izq(b,1)*V2625_izq(b,1)+V2524_izq(b,2)*V2625_izq(b,2)+V2524_izq(b,3)*V2625_izq(b,3))/(sqrt(V2524_izq(b,1)^2+V2524_izq(b,2)^2+V2524_izq(b,3)^2)*sqrt(V2625_izq(b,1)^2+V2625_izq(b,2)^2+V2625_izq(b,3)^2));
alpha_izq2(b)=acos(coalpha_izq2(b))*(180/pi);
end

%Representamos los valores del angulo para ambos modelos
figure(1)
subplot(2,2,1);
plot(alpha_der1,'b');
hold on;
plot(alpha_der2,'g');
title('Modelo de pierna derecha, alpha1 y alpha2');

subplot(2,2,2);
plot(alpha_izq1,'r');
hold on;
plot(alpha_izq2,'y');
title('Modelo de pierna izquierda, alpha1 y alpha2');

subplot(2,2,3);
plot(alpha_der1,'b');
hold on;

```

```

plot(alpha_izq1,'r');
hold on;
title('Ambos modelos, alpha1');

subplot(2,2,4);
plot(alpha_der2,'g');
hold on;
plot(alpha_izq2,'y');
title('Ambos modelos, alpha2');

%Calculamos la funcion de asimetria

mostrar = 1;

while (mostrar == 1)

    opcion = input('Seleccione la función de asimetría que desea
calcular (de 1 a 6): ');

    switch opcion

        case 1,
            for i=1:1:100
                if(abs(alpha_izq1(i)-alpha_der1(i))<2)
                    coas1(i)=0;
                else
                    coas1(i)=(alpha_izq1(i)-alpha_der1(i));
                end
            end

            for i=1:1:100
                if(abs(alpha_izq2(i)-alpha_der2(i))<2)
                    coas2(i)=0;
                else
                    coas2(i)=(alpha_izq2(i)-alpha_der2(i));
                end
            end

        case 2,
            for i=1:1:100
                if(abs(alpha_izq1(i)-alpha_der1(i))<2)
                    coas1(i)=0;
                else
                    coas1(i)=(alpha_izq1(i)-
alpha_der1(i))/(alpha_izq1(i)+alpha_der1(i));
                    coas1(i)=coas1(i)*100;
                end
            end

            for i=1:1:100
                if(abs(alpha_izq2(i)-alpha_der2(i))<2)
                    coas2(i)=0;
                else
                    coas2(i)=(alpha_izq2(i)-
alpha_der2(i))/(alpha_izq2(i)+alpha_der2(i));
                    coas2(i)=coas2(i)*100;
                end
            end
    end
end

```

```

end

case 3,
for i=1:1:100
if(abs(alpha_izq1(i)-alpha_der1(i))<2)
coas1(i)=0;
else
coas1(i)=abs((alpha_der1(i)/alpha_izq1(i))-
(alpha_izq1(i)/alpha_der1(i)));
coas1(i)=coas1(i)*100;
end
end

for i=1:1:100
if(abs(alpha_izq2(i)-alpha_der2(i))<2)
coas2(i)=0;
else
coas2(i)=abs((alpha_der2(i)/alpha_der2(i))-
(alpha_izq2(i)/alpha_der2(i)));
coas2(i)=coas2(i)*100;
end
end

case 4,
for i=1:1:100
if(abs(alpha_izq1(i)-alpha_der1(i))<2)
coas1(i)=0;
else
coas1(i)=(alpha_der1(i)/alpha_izq1(i))+(alpha_izq1(i)/alpha_der1(i))-
2;
coas1(i)=coas1(i)*100;
end
end

for i=1:1:100
if(abs(alpha_izq2(i)-alpha_der2(i))<2)
coas2(i)=0;
else
coas2(i)=(alpha_der2(i)/alpha_izq2(i))+(alpha_izq2(i)/alpha_der2(i))-
2;
coas2(i)=coas2(i)*100;
end
end

case 5,
for i=1:1:100
if(abs(alpha_izq1(i)-alpha_der1(i))<2)
coas1(i)=0;
else
coas1(i)=(alpha_der1(i)-
alpha_izq1(i))/max(alpha_der1(i),alpha_izq1(i));
coas1(i)=coas1(i)*100;
end
end

for i=1:1:100

```



```

        if(abs(alpha_izq2(i)-alpha_der2(i))<2)
            coas2(i)=0;
        else
            coas2(i)=(alpha_der2(i)-
alpha_izq2(i))/max(alpha_der2(i),alpha_izq2(i));
            coas2(i)=coas2(i)*100;
        end
    end

    case 6,
        for i=1:1:100
            if(abs(alpha_izq1(i)-alpha_der1(i))<2)
                coas1(i)=0;
            else
                coas1(i)=abs((alpha_der1(i)-
alpha_izq1(i))/max(alpha_der1(i),alpha_izq1(i)));
                coas1(i)=coas1(i)*100;
            end
        end

        for i=1:1:100
            if(abs(alpha_izq2(i)-alpha_der2(i))<2)
                coas2(i)=0;
            else
                coas2(i)=abs((alpha_der2(i)-
alpha_izq2(i))/max(alpha_der2(i),alpha_izq2(i)));
                coas2(i)=coas2(i)*100;
            end
        end

    case 0,
        mostrar = 0;
        Menu;

    otherwise,
        disp('Opción incorrecta ')
        for i=1:1:100
            coas1(i)=0;
            coas2(i)=0;
        end

end

%Calculamos el coeficiente de asimetria
media1=mean(coas1);
media2=mean(coas2);
for i=1:1:100
    resultado1(i)=(coas1(i)-media1)^2;
    resultado2(i)=(coas2(i)-media2)^2;
end

coeficiente1=sqrt(mean(resultado1));
coeficiente2=sqrt(mean(resultado2));

%Representamos el coeficiente de asimetria
disp(' ');

```

```

        disp('El coeficiente de asimetría para la primera pierna es:
');disp(coeficiente1);
        disp(' ');
        disp('El coeficiente de asimetría para la segunda pierna es:
');disp(coeficiente2);

        %Representamos la funcion de asimetria
        if (opcion==1)
            figure(4)
            subplot(2,1,1);
            plot(coas1,'b');
            title('Función de asimetría entre ambos modelos - Primera
pierna');
            ylabel('Valor en °');

            subplot(2,1,2);
            plot(coas2,'g');
            title('Función de asimetría entre ambos modelos - Segunda
pierna');
            ylabel('Valor en °');
        else
            figure(4)
            subplot(2,1,1);
            plot(coas1,'b');
            title('Función de asimetría entre ambos modelos - Primera
pierna');
            ylabel('Valor en %');

            subplot(2,1,2);
            plot(coas2,'g');
            title('Función de asimetría entre ambos modelos - Segunda
pierna');
            ylabel('Valor en %');
        end
    end
end

```

Código de anguloGamma:

```

%ANGULO GAMMA

%Para el modelo de paso derecho calculamos los puntos de las dos
piernas
P4_der = der(:,10:12);
P5_der = der(:,13:15);
P12_der = der(:,34:36);
P18_der = der(:,52:54);
V1218_der = P12_der - P18_der;
V54_der = P5_der - P4_der;

P22_der = der(:,64:66);
P23_der = der(:,67:69);
P12_der = der(:,34:36);
P18_der = der(:,52:54);
V1218_der = P12_der - P18_der;

```

```

V2322_der = P23_der - P22_der;

%Para el modelo de paso derecho calculamos los valores del angulo
for i=1:1:100

cogamma_der1(i)=(V1218_der(i,1)*V54_der(i,1)+V1218_der(i,2)*V54_der(i,
2)+V1218_der(i,3)*V54_der(i,3))/(sqrt(V1218_der(i,1)^2+V1218_der(i,2)^
2+V1218_der(i,3)^2)*sqrt(V54_der(i,1)^2+V54_der(i,2)^2+V54_der(i,3)^2)
);
    gamma_der1(i)=acos(cogamma_der1(i))*(180/pi);
end

for j=1:1:100

cogamma_der2(j)=(V1218_der(j,1)*V2322_der(j,1)+V1218_der(j,2)*V2322_de
r(j,2)+V1218_der(j,3)*V2322_der(j,3))/(sqrt(V1218_der(j,1)^2+V1218_der
(j,2)^2+V1218_der(j,3)^2)*sqrt(V2322_der(j,1)^2+V2322_der(j,2)^2+V2322
_der(j,3)^2));
    gamma_der2(j)=acos(cogamma_der2(j))*(180/pi);
end

%Para el modelo de paso derecho calculamos los puntos de las dos
piernas
P4_izq = izq(:,10:12);
P5_izq = izq(:,13:15);
P12_izq = izq(:,34:36);
P18_izq = izq(:,52:54);
V1218_izq = P12_izq - P18_izq;
V54_izq = P5_izq - P4_izq;

P22_izq = izq(:,64:66);
P23_izq = izq(:,67:69);
P12_izq = izq(:,34:36);
P18_izq = izq(:,52:54);
V1218_izq = P12_izq - P18_izq;
V2322_izq = P23_izq - P22_izq;

%Para el modelo de paso izquierdo calculamos los valores del angulo
for i=1:1:100

cogamma_izq1(i)=(V1218_izq(i,1)*V54_izq(i,1)+V1218_izq(i,2)*V54_izq(i,
2)+V1218_izq(i,3)*V54_izq(i,3))/(sqrt(V1218_izq(i,1)^2+V1218_izq(i,2)^
2+V1218_izq(i,3)^2)*sqrt(V54_izq(i,1)^2+V54_izq(i,2)^2+V54_izq(i,3)^2)
);
    gamma_izq1(i)=acos(cogamma_izq1(i))*(180/pi);
end

for j=1:1:100

cogamma_izq2(j)=(V1218_izq(j,1)*V2322_izq(j,1)+V1218_izq(j,2)*V2322_iz
q(j,2)+V1218_izq(j,3)*V2322_izq(j,3))/(sqrt(V1218_izq(j,1)^2+V1218_izq
(j,2)^2+V1218_izq(j,3)^2)*sqrt(V2322_izq(j,1)^2+V2322_izq(j,2)^2+V2322
_izq(j,3)^2));
    gamma_izq2(j)=acos(cogamma_izq2(j))*(180/pi);
end

%Representamos los valores del angulo para ambos modelos

```

```

figure(3)
subplot(2,2,1);
plot(gamma_der1,'b');
hold on;
plot(gamma_der2,'g');
title('Modelo de pierna derecha, gamma1 y gamma2');

subplot(2,2,2);
plot(gamma_izq1,'r');
hold on;
plot(gamma_izq2,'y');
title('Modelo de pierna izquierda, gamma1 y gamma2');

subplot(2,2,3);
plot(gamma_der1,'b');
hold on;
plot(gamma_izq1,'r');
hold on;
title('Ambos modelos, gamma1');

subplot(2,2,4);
plot(gamma_der2,'g');
hold on;
plot(gamma_izq2,'y');
title('Ambos modelos, gamma2');

%Calculamos la funcion de asimetria

mostrar = 1;

while (mostrar == 1)

    opcion = input('Seleccione la función de asimetría que desea
calcular (de 1 a 6): ');

    switch opcion

        case 1,
            for i=1:1:100
                if(abs(gamma_izq1(i)-gamma_der1(i))<2)
                    coas1(i)=0;
                else
                    coas1(i)=(gamma_izq1(i)-gamma_der1(i));
                end
            end

            for i=1:1:100
                if(abs(gamma_izq2(i)-gamma_der2(i))<2)
                    coas2(i)=0;
                else
                    coas2(i)=(gamma_izq2(i)-gamma_der2(i));
                end
            end

        case 2,
            for i=1:1:100
                if(abs(gamma_izq1(i)-gamma_der1(i))<2)

```

```

        coas1(i)=0;
    else
        coas1(i)=(gamma_izq1(i)-
gamma_der1(i))/(gamma_izq1(i)+gamma_der1(i));
        coas1(i)=coas1(i)*100;
    end
end

    for i=1:1:100
        if(abs(gamma_izq2(i)-gamma_der2(i))<2)
            coas2(i)=0;
        else
            coas2(i)=(gamma_izq2(i)-
gamma_der2(i))/(gamma_izq2(i)+gamma_der2(i));
            coas2(i)=coas2(i)*100;
        end
    end

    case 3,
        for i=1:1:100
            if(abs(gamma_izq1(i)-gamma_der1(i))<2)
                coas1(i)=0;
            else
                coas1(i)=abs((gamma_der1(i)/gamma_izq1(i))-
(gamma_izq1(i)/gamma_der1(i)));
                coas1(i)=coas1(i)*100;
            end
        end

        for i=1:1:100
            if(abs(gamma_izq2(i)-gamma_der2(i))<2)
                coas2(i)=0;
            else
                coas2(i)=abs((gamma_der2(i)/gamma_der2(i))-
(gamma_izq2(i)/gamma_der2(i)));
                coas2(i)=coas2(i)*100;
            end
        end

    case 4,
        for i=1:1:100
            if(abs(gamma_izq1(i)-gamma_der1(i))<2)
                coas1(i)=0;
            else
                coas1(i)=(gamma_der1(i)/gamma_izq1(i)+(gamma_izq1(i)/gamma_der1(i))-
2;
                coas1(i)=coas1(i)*100;
            end
        end

        for i=1:1:100
            if(abs(gamma_izq2(i)-gamma_der2(i))<2)
                coas2(i)=0;
            else
                coas2(i)=(gamma_der2(i)/gamma_izq2(i)+(gamma_izq2(i)/gamma_der2(i))-
2;

```

```

        coas2(i)=coas2(i)*100;
    end
end

case 5,
    for i=1:1:100
        if(abs(gamma_izq1(i)-gamma_der1(i))<2)
            coas1(i)=0;
        else
            coas1(i)=(gamma_der1(i)-
gamma_izq1(i))/max(gamma_der1(i),gamma_izq1(i));
            coas1(i)=coas1(i)*100;
        end
    end

    for i=1:1:100
        if(abs(gamma_izq2(i)-gamma_der2(i))<2)
            coas2(i)=0;
        else
            coas2(i)=(gamma_der2(i)-
gamma_izq2(i))/max(gamma_der2(i),gamma_izq2(i));
            coas2(i)=coas2(i)*100;
        end
    end

case 6,
    for i=1:1:100
        if(abs(gamma_izq1(i)-gamma_der1(i))<2)
            coas1(i)=0;
        else
            coas1(i)=abs((gamma_der1(i)-
gamma_izq1(i))/max(gamma_der1(i),gamma_izq1(i)));
            coas1(i)=coas1(i)*100;
        end
    end

    for i=1:1:100
        if(abs(gamma_izq2(i)-gamma_der2(i))<2)
            coas2(i)=0;
        else
            coas2(i)=abs((gamma_der2(i)-
gamma_izq2(i))/max(gamma_der2(i),gamma_izq2(i)));
            coas2(i)=coas2(i)*100;
        end
    end

case 0,
    mostrar = 0;
    Menu;

otherwise,
    disp ('Opción incorrecta ')
    for i=1:1:100
        coas1(i)=0;
        coas2(i)=0;
    end
end

```

```

end

%Calculamos el coeficiente de asimetria
medial=mean(coas1);
media2=mean(coas2);
for i=1:1:100
    resultado1(i)=(coas1(i)-medial)^2;
    resultado2(i)=(coas2(i)-media2)^2;
end

coeficiente1=sqrt(mean(resultado1));
coeficiente2=sqrt(mean(resultado2));

%Representamos el coeficiente de asimetria
disp(' ');
disp('El coeficiente de asimetría para la primera pierna es:');disp(coeficiente1);
disp(' ');
disp('El coeficiente de asimetría para la segunda pierna es:');disp(coeficiente2);

%Representamos la funcion de asimetria
if (opcion==1)
    figure(4)
    subplot(2,1,1);
    plot(coas1,'b');
    title('Función de asimetría entre ambos modelos - Primera
pierna')
    ylabel('Valor en °');

    subplot(2,1,2);
    plot(coas2,'g');
    title('Función de asimetría entre ambos modelos - Segunda
pierna');
    ylabel('Valor en °');
else
    figure(4)
    subplot(2,1,1);
    plot(coas1,'b');
    title('Función de asimetría entre ambos modelos - Primera
pierna')
    ylabel('Valor en %');

    subplot(2,1,2);
    plot(coas2,'g');
    title('Función de asimetría entre ambos modelos - Segunda
pierna');
    ylabel('Valor en %');
end

end

```

Código de graf3D:

```
%REPRESENTACION 3D

function graf3D(a,b)

n=size(a,1);
cont=0;
for i=1:1:3

    while(cont<200)
        for q=1:n
            j=1:26;
            r=j-1;
            plot3((a(q,3*j)), (a(q,3*j-2)), (-a(q,3*j-1)), 'ro');
            hold on
            plot3((b(q,3*j)), (b(q,3*j-2)), (-b(q,3*j-1)), '.');
            axis([-1000 1000 -1000 1000 -1000 1000]);
            grid
            legend('Modelo', 'Original', 'Location', 'Best')
            hold off
            drawnow
            pause((30/1000));
            cont=cont+1;
        end
    end

end
end
```

Código de graf2D suelo:

```
%REPRESENTACION 2D VISTA DESDE EL SUELO

function graf2D_suelo(a,b)

n=size(a,1);
for i=1:1:3

    cont=0;

    while(cont<200)
        for q=1:n
            j=1:26;
            r=j-1;
            plot((a(q,3*j)), (a(q,3*j-2)), 'ro');
            hold on
            plot((b(q,3*j)), (b(q,3*j-2)), '.');
            axis([-1000 1000 -1000 1000]);
            grid
            legend('Modelo', 'Original', 'Location', 'Best')
            hold off
            drawnow
            pause((30/1000));
        end
    end

end
```



```

        cont=cont+1;
    end
    end
end

```

```

end

```

Código de graf2D costado:

```

%REPRESENTACION 2D VISTA DESDE EL COSTADO

```

```

function graf2D_costado(a,b)

n=size(a,1);
for i=1:1:3

    cont=0;

    while(cont<200)
        for q=1:n
            j=1:26;
            r=j-1;
            plot((a(q,3*j)), (-a(q,3*j-1)), 'ro');
            hold on
            plot((b(q,3*j)), (-b(q,3*j-1)), '.');
            axis([-1000 1000 -1000 1000]);
            grid
            legend('Modelo', 'Original', 'Location', 'Best')
            hold off
            drawnow
            pause((30/1000));
            cont=cont+1;
        end
    end
end

end

```

Código de graf2D frente:

```

%REPRESENTACION 2D VISTA DESDE EL COSTADO

```

```

function graf2D_costado(a,b)

n=size(a,1);
for i=1:1:3

    cont=0;

```

```

while(cont<200)
for q=1:n
j=1:26;
r=j-1;
plot((a(q,3*j)),(-a(q,3*j-1)),'ro');
hold on
plot((b(q,3*j)),(-b(q,3*j-1)),'. ');
axis([-1000 1000 -1000 1000]);
grid
legend('Modelo', 'Original','Location','Best')
hold off
drawnow
pause((30/1000));
cont=cont+1;
end
end
end

end

```

ANEXO II: Artículo publicado

Asymmetry functions of joint angles in the human gait

José Carlos Lara Gómez^a, Jacek Dusza^{b,c,*}

^aETS de Ingeniería Informática y de Telecomunicaciones, Granada, Spain, ^bWarsaw University of Technology, Institute of Electronics Systems, ^cMedical University of Warsaw

ABSTRACT

The idea of this project is to study the asymmetry during human walking on some important angles. For this purpose, asymmetry functions have been developed. Until this moment, the asymmetry on these angles has been studied like something global with the asymmetry coefficients (for example, just taking the best value of asymmetry for every angle). With the asymmetry functions it is possible to study this asymmetry along the whole. Knowing this information, it could be possible to determine different anomalies depending on the angle and on the moment of the asymmetry. The ultimate goal is to obtain a computer program to perform the asymmetry functions at different angles from data obtained from patients.

Keywords: Asymmetry, asymmetry functions, asymmetry coefficients, joint angle, middle double step.

1. INTRODUCTION

At first sight, the walking process seems to be periodical and symmetrical, but neither of these are true. However, the differences between following gait periods and the difference between the right and left step are so insignificant, that gait is assumed as quasi-periodical and quasi-symmetrical. On the other hand, if someone has huge walking asymmetry or deviation of periodicity of the gait, it usually means that he is suffering from a disease. For a thorough analysis of gait asymmetry, it is worth to develop two of its symmetric models, one based on the left step and the second model, based on the right step.

1.1 Data capture:

The study of the asymmetry functions will be conducted using Matlab¹. First, it is necessary to obtain patient data while he is walking. To do this, the patient is walking on a treadmill with infrared reflectors placed on the body at the points shown in the Figure 1. Data capture is carried out using infrared cameras. Location of the markers is recorded with 4 cameras and then 3D reconstruction is performed. There is thus obtained a signal matrix where each column describes the corresponding coordinate for the selected marker.

1.2 Double half-step models:

Once data are captured, they are adapted to the double half step model². Using the techniques of double half step model makes it possible to control and characterize the kinematics of the subject's body by comparing the data obtained with the data of a normal step in each time step.

Once data are normalized, the separation of the data concerning the right and the left legs is performed. Having done that, two models that simulate the complete walking based on each of the legs are created. These models are nothing but arrays of the same dimension data, so that any point can be compared in each of the legs at the same instant of time. These are the data which are split for the study of the asymmetry. The program that allows us to obtain these models was created by Asier Sancho Hidalgo³.

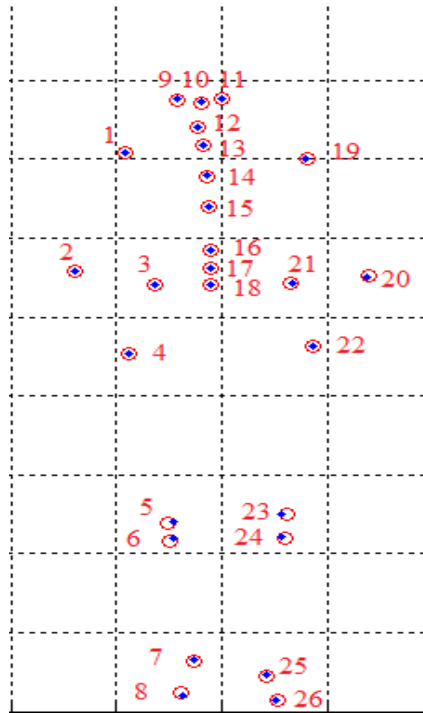


Figure 1. Location of markers on the human body in frontal plane.

2. JOINT ANGLES

For this project we have chosen three angles that are of great importance for the study of asymmetry in human walking. These angles are shown in the Figure 2:

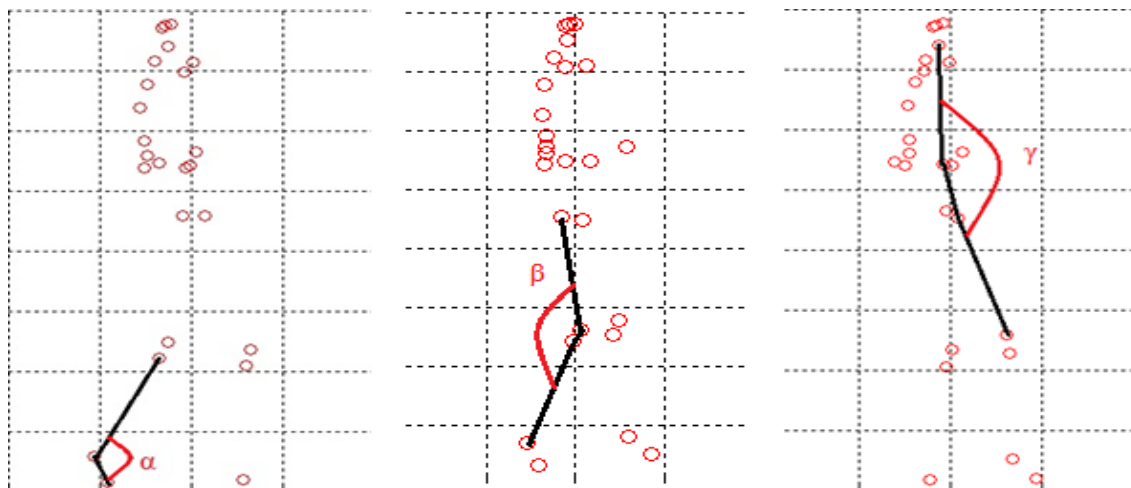


Figure 2. Presentation of the joint angles in the ankle, knee and hip

From data of left and right step models, it is possible to calculate the vectors that define the different angles and from them the own angles. The program will allow selecting between these angles and calculating the different asymmetry functions.

3. ASYMMETRY FUNCTIONS

After selecting one angle, the temporal function that represents the value of this angle along the time is created for both models (right step and left step). Given these functions, it is possible to compute a new function that compares the asymmetry between the two models for each of the time instants.

In the case of this program, it is possible to calculate up to six different asymmetry functions, with different properties. Definitions of asymmetry functions are presented below.

$$AF_1(i) = LeftAng(i) - RightAng(i) [\%] \quad (1)$$

$$AF_2(i) = \frac{LeftAng(i) - RightAng(i)}{LeftAng(i) + RightAng(i)} [\%] \quad (2)$$

$$AF_3(i) = \left| \frac{RightAng(i)}{LeftAng(i)} - \frac{LeftAng(i)}{RightAng(i)} \right| [\%] \quad (3)$$

$$AF_4(i) = \left(\frac{RightAng(i)}{LeftAng(i)} + \frac{LeftAng(i)}{RightAng(i)} - 2 \right) [\%] \quad (4)$$

$$AF_5(i) = \frac{RightAng(i) - LeftAng(i)}{\max(RightAng(i), LeftAng(i))} [\%] \quad (5)$$

$$AF_6(i) = \left| \frac{RightAng(i) - LeftAng(i)}{\max(RightAng(i), LeftAng(i))} \right| [\%] \quad (6)$$

Where AF_n is asymmetry function, $RightAng(i)$ and $LeftAng(i)$ are respective angles defined for the i -th image.

From the asymmetry function, it is possible to obtain the value of the asymmetry coefficient, which measures the degree of asymmetry of the distribution with respect to the arithmetic mean of the values obtained in different moments of time in the asymmetry function.

$$AC = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

4. OPERATION OF THE PROGRAM

When the program is executed, the operation is given in the form of menus to interact with. To boot, the data concerning the right step and left step models are needed. The choices are represented in the Figure 3.

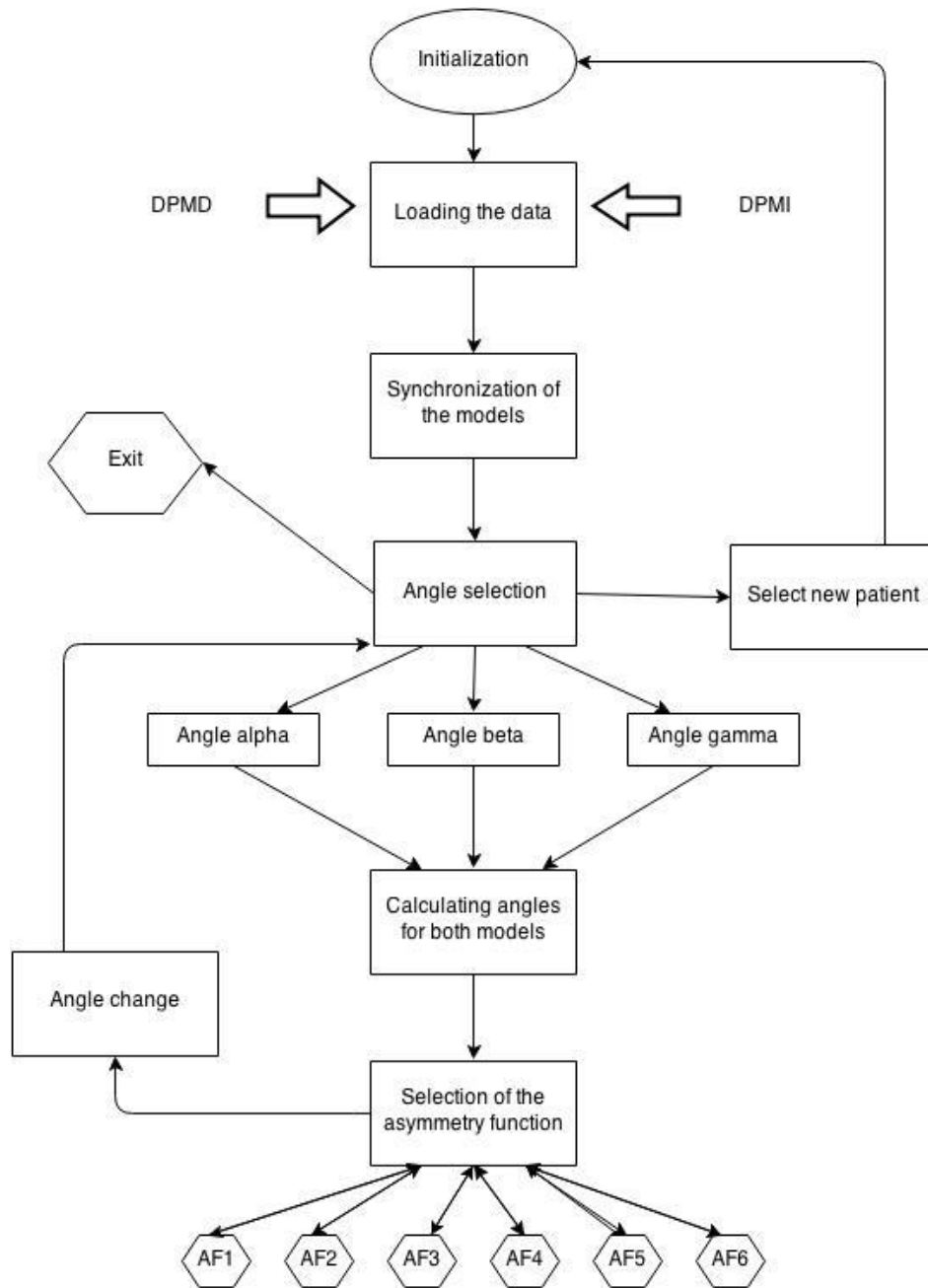


Figure 3. flow diagram of presented program

5. RESULTS

Below, it is presented the result of the execution of the program selecting one angle (in this case, alpha angle).

Representation of the angle: The first image shows the representation of the alpha angle for both legs in the right step model. The second image shows the representation of the same angle in the left step model. The third image shows the representation of the comparison of one leg in both models. The last picture shows the comparison of the other leg in both models.

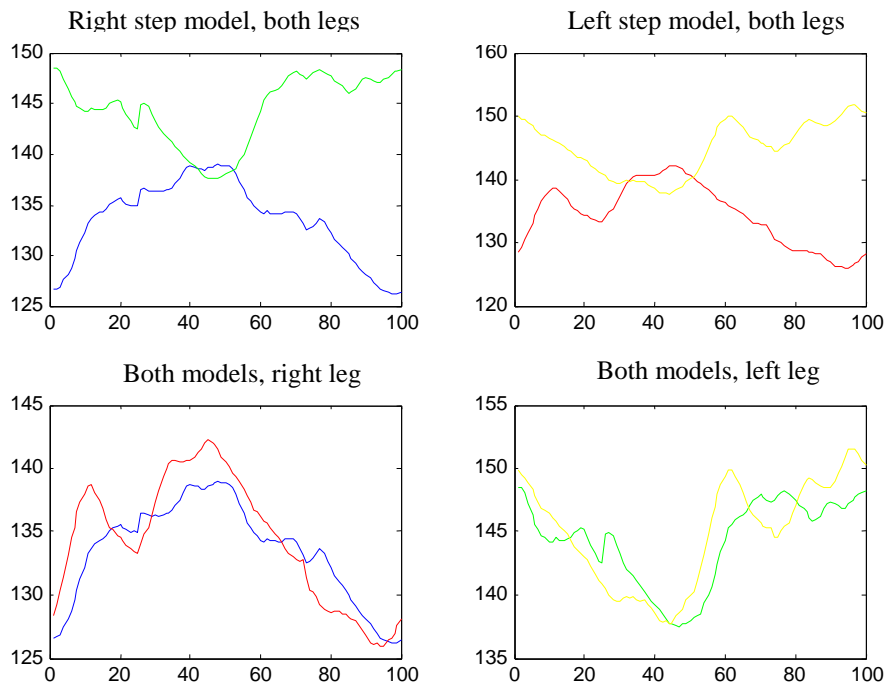


Figure 4. First asymmetry function where it is simply shown the difference in degrees between the two models for each of the time intervals.

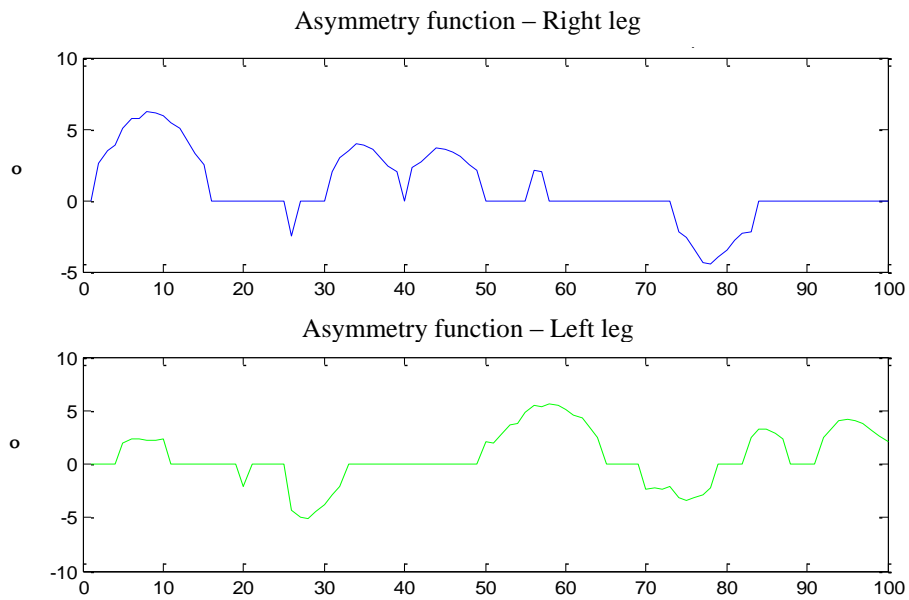


Figure 5. First asymmetry function where it is simply shown the difference in degrees between the two models for each of the time intervals.

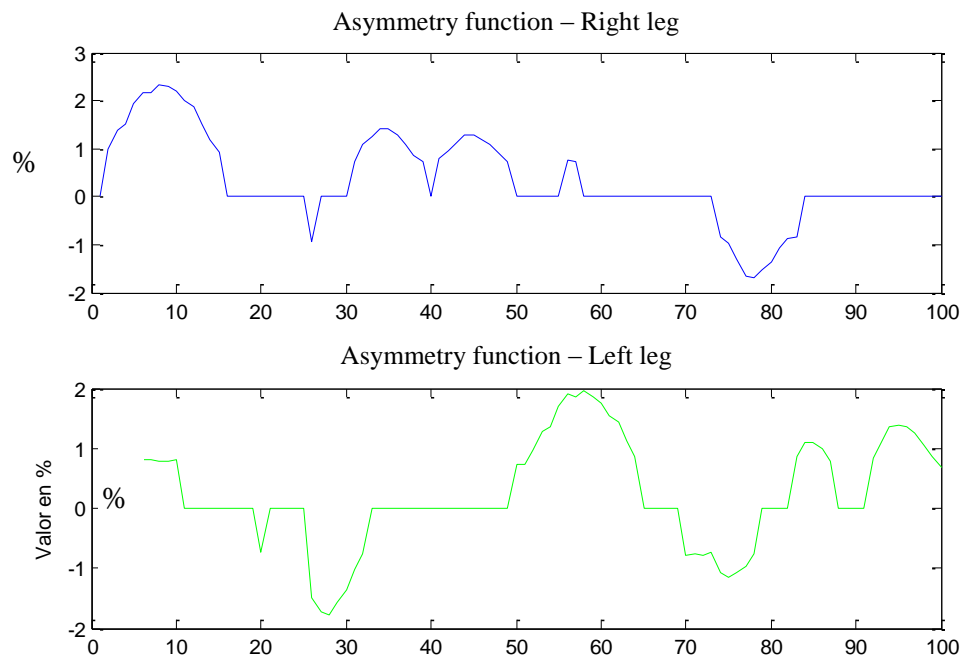


Figure 6. Second asymmetry function. This function seems similar to the previous function, but given percentage.

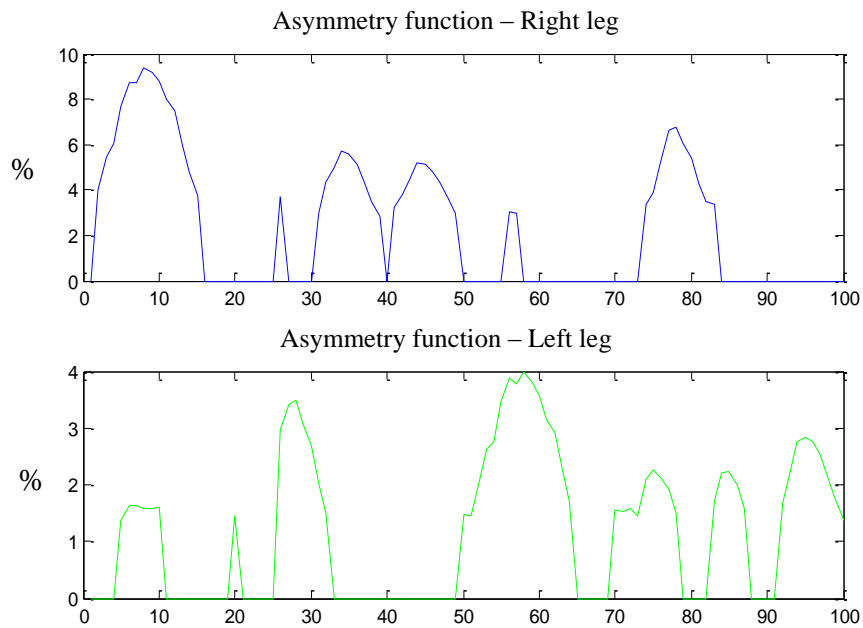


Figure 7. Third asymmetry function is very sensitive to changes. It has large leaps as differences in different time instants are larger.

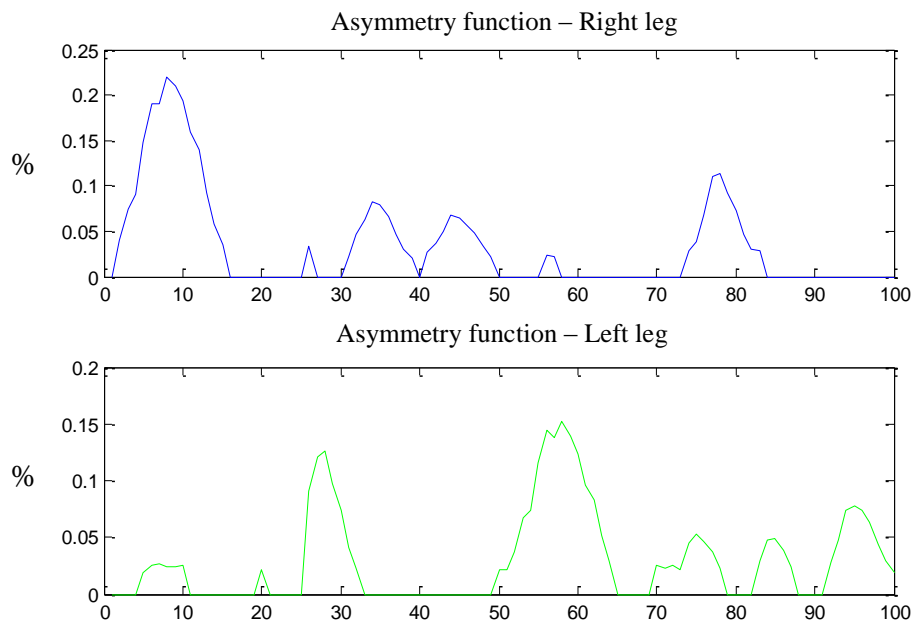


Figure 8. Fourth asymmetry function: Like the one above, it is also very sensitive to changes, but represents a much smaller scale (0 to 1%).

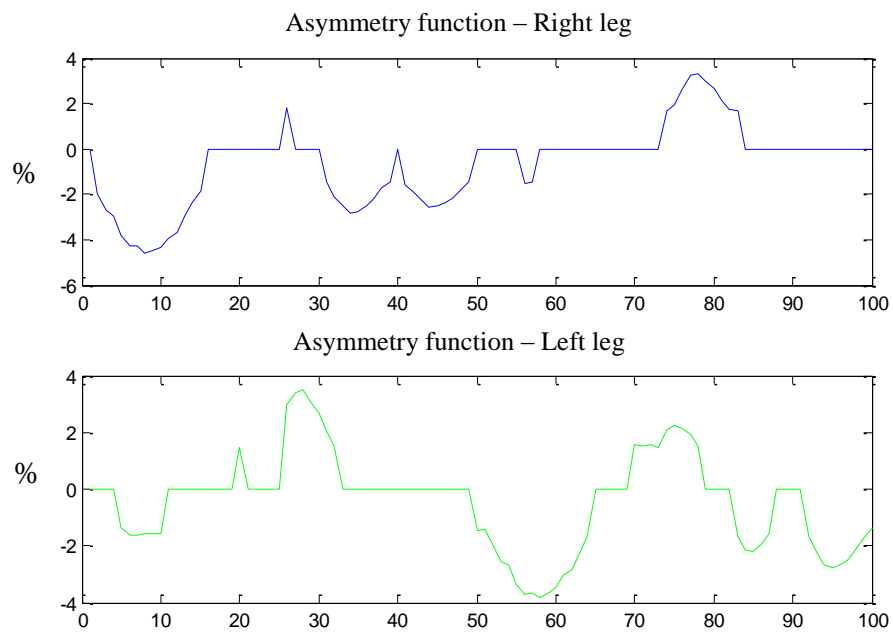


Figure 9. Fifth asymmetry function: This function is more sensitive in each of the time intervals to the angle that presents a greater value in each case.

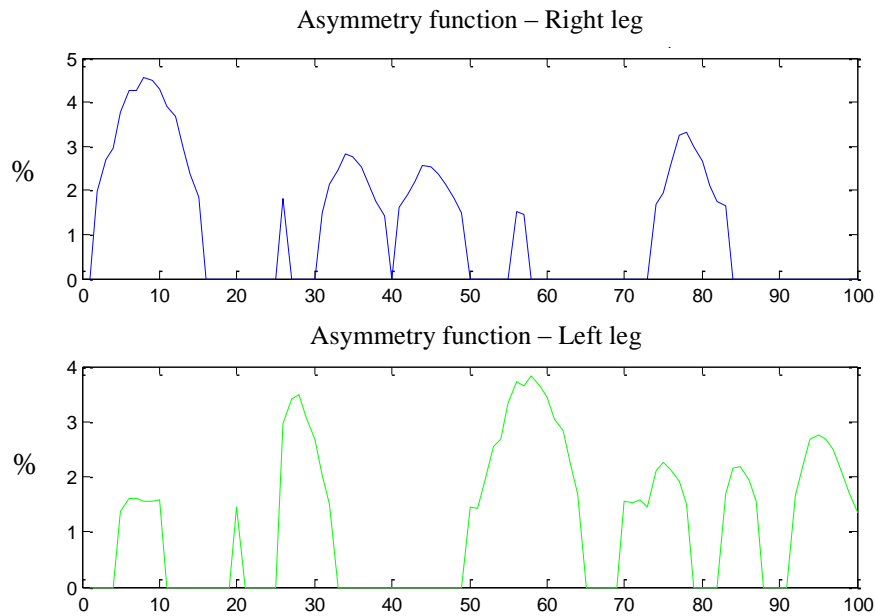


Figure 10. Sixth asymmetry function: Very similar to the function mentioned above, but only in a positive range due to the absolute value. It does not take into account which leg has a higher value of the angle at each time instant

6. CONCLUSIONS

6.1 Categorize a patient:

If a study about asymmetry coefficients is performed, it may be the case that a patient would show a large asymmetry in any angle, but after observing the asymmetry function of the patient in that angle it does not present large asymmetries. This may be because the function at any instant of time presents a great leap, which would increase the asymmetry coefficient. Thus, we arrive at the conclusion that the most optimal way to classify a patient is studying the asymmetry function.

Thus, it should be done a division in phases for the full walking, to study the asymmetry in each of the angles during different phases. Then, some levels could be created, so that they could differentiate different pathologies depending on the asymmetry value, the phase at which they occur and the angle at which they occur.

6.2 Division of the asymmetry function in phases:

A possible division into phases is suggested by Jacquelin Perry⁴. This can be seen in the Figure 11. Thus, taking this division, the different phases on angle representation and function asymmetry:

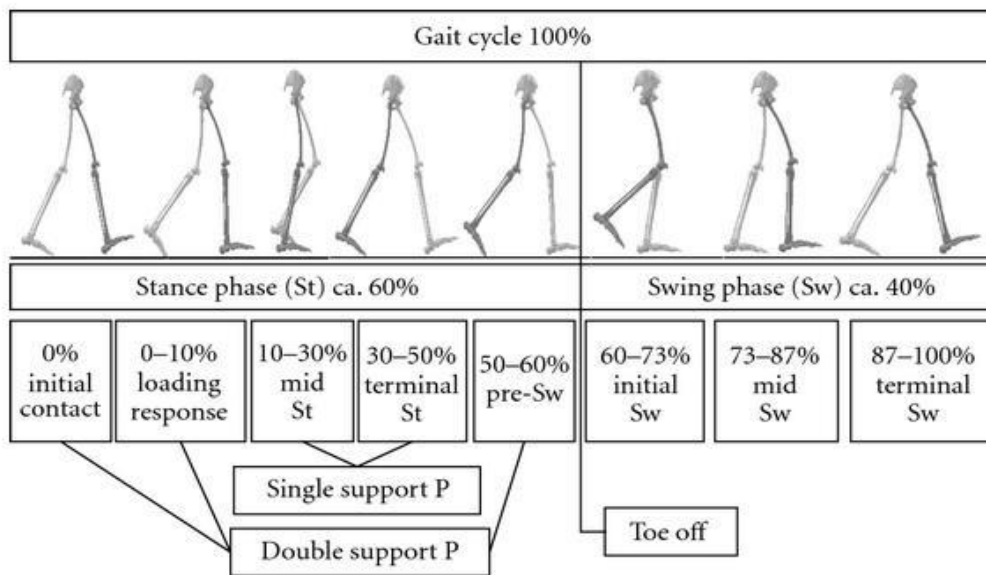


Figure 11. Gait phases suggested by Jacquelin Perry.

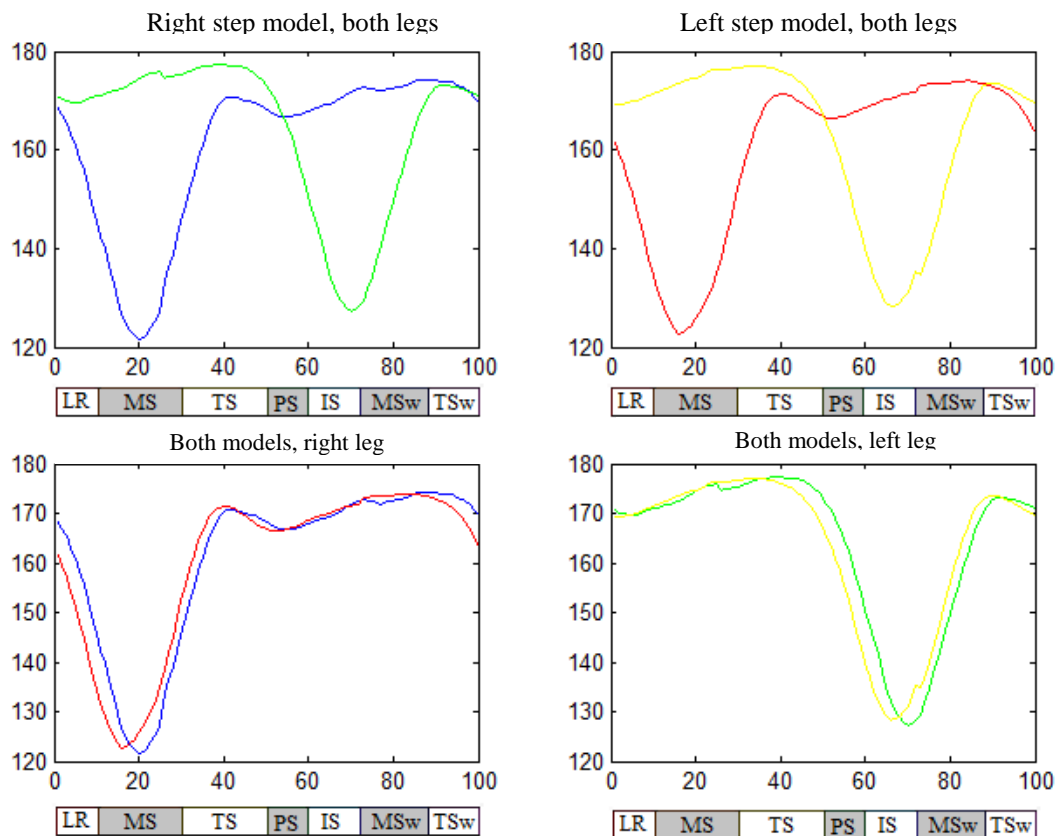


Figure 12. Joint angles correlated with gait phases

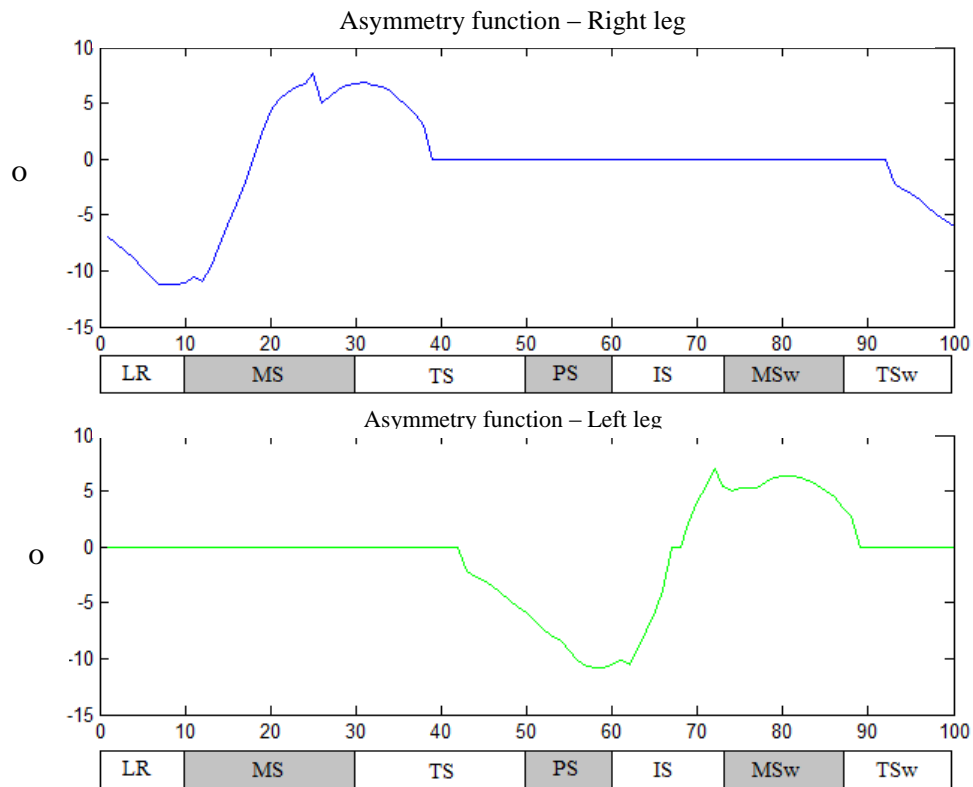


Figure 13. Changes of the asymmetry function in subsequent phases of the gait

In this specific example it is possible to observe that for the asymmetry function of the right leg there are big asymmetries in the “mid st” and “terminal st” phases. However, in the “pre-sw”, “initial sw” and “mid sw” phases there are not asymmetries.

By studying the different phases is easier to differentiate between different pathologies according to the angle.

It is necessary to clarify that the existence of asymmetry in certain phases according to the angle does not imply the existence of one anomaly. It would be necessary to study different cases with a significant variety of patients to get conclusions about the current asymmetries.

REFERENCES

- [1] MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.
- [2] J. J. Dusza, Z. M. Wawrzyniak C. F. Mugarra González.: Simplified model of mean double step (MDS) in human body movement, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2012, edited by Ryszard S. Romaniuk, Proc. of SPIE Vol. 8454, 84541N · © 2012 SPIE CCC code: 0277-786/12/\$18 · doi: 10.1117/12.200005
- [3] Asier Sancho Hidalgo: Student of the University of Valencia. Developer of the needed program for obtain the left step and right step models.
- [4] Jacquelin Perry: “Gait Analysis: Normal and Pathological Function” book writer, which describes the various stages in human walking.