Desarrollo de una guía metodológica para el registro y análisis de marcha protésica en usuarios de prótesis de miembro inferior utilizando OpenSim

María Paulina Tobón Palacio

Bioingeniera

Modalidad de Práctica
Trabajo de Grado

Bioingeniería
Juliana Uribe Pérez, PhD

Universidad de Antioquia
Facultad de Ingeniería
Bioingeniería
Medellín
2025

Cita	Tobón Palacio [1]
Referencia Estilo IEEE (2020)	[1] Tobón Palacio, “Desarrollo de una guía metodológica para el registro y análisis de marcha protésica en usuarios de prótesis de miembro inferior utilizando OpenSim” Trabajo de grado, Bioingeniería, Universidad de Antioquia, Medellín, 2025







Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

TABLA DE CONTENIDO

RESUMEN..... 9
ABSTRACT..... 10
I. INTRODUCCIÓN..... 11
II. OBJETIVOS..... 13
A. Objetivo general..... 13
B. Objetivos específicos..... 13
III. MARCO TEÓRICO..... 14
Prótesis transfemorales..... 14
Ciclo de marcha..... 15
OpenSim..... 17
Cinemática y cinemática inversa..... 19
Dinámica y dinámica inversa..... 20
Algoritmo de reducción residual..... 20
Optimización estática..... 21
Control muscular computacional..... 22
IV. METODOLOGÍA..... 23
Protocolo de registro e importación de datos de marcha protésica en OpenSim..... 23
Flujo de trabajo para el análisis de marcha protésica en OpenSim..... 24
Evaluación de la calidad de los protocolos propuestos..... 25
V. ANÁLISIS DE RESULTADOS..... 26
Protocolo de registro e importación de datos de marcha protésica en OpenSim..... 26
Flujo de trabajo para el análisis de marcha protésica en OpenSim..... 31
Evaluación de la calidad de los protocolos propuestos..... 39
VI. CONCLUSIONES Y RECOMENDACIONES..... 40
REFERENCIAS..... 41
ANEXOS..... 44

LISTA DE TABLAS

TABLA I
EVALUACIÓN DE ESCALADO 32
TABLA II
RESULTADOS DE RRA 34

LISTA DE FIGURAS

Fig. 1 Prótesis transfemoral 14
Fig. 2 Ciclo de marcha 15
Fig. 3 Partes del ciclo de marcha 16
Fig. 4 Ventana principal de OpenSim 17
Fig. 5 Gráficas de OpenSim para el momento del ángulo de la articulación del talón 17
Fig. 6 Metodología 22
Fig. 7 Flujo de trabajo de OpenSim 26
Fig. 8 Trazado del vector de fuerza de reacción del suelo. 28
Fig. 9 Trazado del vector de fuerza de reacción del suelo obtenido con MOKKA 29
Fig. 10 Vector de fuerza de reacción del suelo del sujeto de prueba 30
Fig. 11 Detalles del flujo de trabajo de OpenSim 31
Fig. 12 Modelo de amputación transfemoral izquierda 37

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

ID	Dinámica inversa
IK	Cinemática inversa
OS	Optimización estática
CMC	Control muscular computarizado
RRA	Algoritmo de reducción residual
EMG	Electromiografía
RMS	Raíz cuadrática media

RESUMEN

El objetivo de este trabajo fue desarrollar una metodología para el registro de la marcha en usuarios de prótesis de miembro inferior y el análisis de la marcha protésica utilizando OpenSim, para ello se partió de una búsqueda bibliográfica que permitió comprender los aspectos relacionados con el registro de marcha, la preparación de archivos, el uso de las herramientas de escalado, cinemática inversa, dinámica inversa, algoritmo de reducción residual, optimización estática y control muscular computarizado de OpenSim, así como el proceso requerido para simular una amputación transfemoral en este software. Los detalles asociados al uso y validación de cada herramienta se consignaron en una guía metodológica que describe cada proceso y apoya al usuario por medio de imágenes. Por otro lado, se puso a prueba la claridad del material por medio de un proceso de evaluación que permitió efectuar correcciones en la guía metodológica y medir su comprensibilidad, alcanzando un puntaje de 4.78 sobre 5. Además, se realizó un video donde se ejecutan los pasos de la guía para mejorar la calidad del material. Por lo tanto, se concluye que fue posible establecer una metodología detallada para el registro y el análisis de la marcha protésica en usuarios de prótesis de miembro inferior, así como la simulación de amputación transfemoral, utilizando OpenSim.

Palabras clave **— Amputación, marcha, OpenSim, simulación, cinemática, dinámica.**

ABSTRACT

The objective of this project was to develop a methodology for gait recording in users of lower limb prostheses and the analysis of prosthetic gait using OpenSim. To achieve this, a literature review was conducted to understand aspects related to gait recording, file preparation, and the use of OpenSim tools such as scaling, inverse kinematics, inverse dynamics, residual reduction algorithm, static optimization, and computed muscle control, as well as the process required to simulate a transfemoral amputation in this software. The details regarding the use and validation of each tool were compiled into a methodological guide that describes each process and supports the user through images. Additionally, the clarity of the material was tested through an evaluation process that allowed for corrections in the methodological guide and assessed its comprehensibility, achieving a score of 4.78 out of 5. Furthermore, a video was created demonstrating the steps outlined in the guide to enhance the quality of the material. Therefore, it is concluded that it was possible to establish a detailed methodology for gait recording and prosthetic gait analysis in users of lower limb prostheses, as well as for the simulation of transfemoral amputation using OpenSim.

Keywords **— Amputation, gait, OpenSim, simulation, kinematics, dynamics.**

I. INTRODUCCIÓN

La pérdida parcial o total de una extremidad inferior es un trastorno que afecta la calidad de vida de las persona que lo sufren, dado que dificulta la movilidad e independencia de la misma, en 2023 [1] publicó una investigación en la que se analizó la carga mundial en 204 países para determinar la incidencia y prevalencia de amputaciones debido a eventos traumáticos, según sus resultados, para 2019, las amputaciones unilaterales y bilaterales de miembros inferiores presentaban una incidencia de $1.126{,}51x10^3$ y $217{,}31x10^3$, respectivamente, mientras que la prevalencia era de $25.637{,}64x10^3$ y $5.960{,}95x10^3$ respectivamente. Estas cifras indican el gran número de personas cuya movilidad se ve afectada debido a la pérdida de una extremidad; sin embargo, este estudio no contempla todas las causas de amputación, puesto que la pérdida de extremidades también puede darse por complicaciones asociadas a otras causas como diabetes, infecciones, cáncer o alteraciones congénitas. Es por ello que se sobreentiende el interés de la comunidad científica en comprender cómo abordar la problemática y desarrollar herramientas que permitan que los pacientes mejoren su calidad de vida. En el marco Colombiano, según las cifras del DANE, entre el 4.3 y el 6.3% de la población total, mayor de 5 años, presenta alguna discapacidad, es decir, entre 1.7 y 3.1 millones de personas, de las cuales, el 31% tiene una condición que le dificulta mover el cuerpo, caminar o subir y bajar escaleras [2]. En este sentido, se infiere la necesidad de desarrollar estrategias que permitan facilitar el estudio de esta condición; por lo cual, se propone como objetivo realizar una guía metodológica que explique el proceso relacionado con el registro y el análisis de la marcha de usuarios de prótesis de miembro inferior utilizando OpenSim.

Teniendo en cuenta lo anterior, diferentes autores han llevado a cabo investigaciones enfocadas a comprender los patrones de marcha de los usuarios de prótesis de miembro inferior, dado que estos difieren de los patrones analizados y descritos para las personas con ambos miembros inferiores sanos, en la mayoría de estudios se destaca el uso de herramientas de simulación que permiten no solo reproducir la marcha de una persona con discapacidad, sino que también ayuda a entender cómo la variación de diferentes parámetros altera las características de su marcha y en qué medida es comparable con los ciclos de marcha de las personas sin amputación. Una de las herramientas más empleadas es OpenSim, dado que este software libre

permite analizar la biomecánica de diferentes modelos musculoesqueléticos así como reproducir movimientos obtenidos de registros de pacientes.

Entre las aplicaciones que se le han dado a este software se encuentran estudios como el realizado por [3], cuyo objetivo consistió en adaptar un modelo biomecánico que permitiera analizar la respuesta muscular de amputados transfemorales para ver la influencia del uso de prótesis. Por otro lado [4] emplea un enfoque con el que buscaban desarrollar una herramienta que permitiera analizar la biomecánica de la marcha y comprender tanto las contribuciones de los actuadores protésicos, como musculares, logrando el desarrollo de un modelo de amputación transfemoral unilateral con prótesis osteointegrada. Finalmente, otro ejemplo del uso de OpenSim es el desarrollado por [5] donde emplean diferentes técnicas y restricciones para realizar un modelo que permite estimar cómo funciona la interfaz entre el encaje y el miembro residual, así como adaptarse tanto a marcha normal como marcha protésica, permitiendo profundizar en la comprensión biomecánica de la marcha protésica y sugiriendo que puede ser empleado para facilitar el desarrollo de protocolos de rehabilitación o mejorar el diseño de dispositivos de asistencia. Ahora bien, aunque todos estos estudios tienen tanto un software como una temática en común, no permiten identificar cuáles son los pasos exactos ni cómo llevarlos a cabo para realizar simulaciones que permitan encaminar los estudios de manera diferente, es por ello que se propone la creación de un protocolo que permita guiar a los investigadores sobre las consideraciones necesarias para implementar simulaciones de marcha protésica, desde la recolección de datos de registro de marcha, hasta la modificación geométrica y osteomuscular del modelo para que coincida con una amputación específica y se agregue un diseño protésico determinado.

II. OBJETIVOS

A. Objetivo general

Desarrollar una metodología para el registro de la marcha en usuarios de prótesis de miembro inferior y el análisis de la marcha protésica utilizando OpenSim.

B. Objetivos específicos

- Implementar un protocolo estandarizado del registro e importación de datos de marcha protésica en el software OpenSim.
- Desarrollar un flujo de trabajo para el análisis de marcha protésica en OpenSim.
- Evaluar la calidad de los protocolos propuestos a través de la retroalimentación de usuarios con conocimientos en el manejo de OpenSim.

III. MARCO TEÓRICO

Con el propósito de comprender el presente trabajo, es necesario desarrollar conceptos teóricos fundamentales relacionados con las características y particularidades de la marcha en usuarios de prótesis de miembro inferior. Las prótesis de miembro inferior son dispositivos diseñados para sustituir extremidades amputadas, permitiendo la bipedestación, es decir, permanecer de pie, y la locomoción. La configuración de estas varía según el tipo de amputación del paciente, que puede clasificarse según la articulación afectada. El presente trabajo se enfocará en amputación transfemoral, es decir, aquellas que involucran la pérdida de la parte inferior de la pierna con desarticulación por encima del nivel de la rodilla [6, 7].

Prótesis transfemorales

Las prótesis transfemorales constan de cuatro componentes principales: el encaje, donde se inserta el muñón; la rodilla protésica, que puede tener diferentes grados de movilidad y número de ejes, según el estilo de vida del usuario; el vástago, que conecta la rodilla con la parte final de la prótesis; y el pie protésico, cuya número de grados de libertad varía en función de las necesidades y preferencias del usuario (**Fig. 1)**. Algunas de las características de estos componentes se seleccionan específicamente para mejorar la adaptación del usuario, como el tipo de rodilla (mecánica, hidráulica o electromecánica), la personalización del vástago para un ajuste cómodo y seguro, y un pie que ofrezca mayor absorción de impactos o flexibilidad. De esta manera, cada prótesis no solo tiene un peso diferente, sino también características particulares que mejoran la funcionalidad y calidad de vida del usuario [8, 9]



Fig. 1 Prótesis transfemoral

Nota. Tomado y adaptado de https://www.ottobock.com/en-ca/product/4R11

Ciclo de marcha

El ciclo de marcha se considera un semiperiodo de carga y descarga, el cual se divide en dos partes, la fase de apoyo y la fase de balanceo, donde la primera se produce cuando la extremidad inferior está en contacto con el suelo, mientras que la segunda se presenta cuando la extremidad está en el aire. Ambas fases se superponen, lo que significa que la fase de balanceo de un pie se traslapa con la mitad de la fase de apoyo del otro. El inicio de un ciclo de marcha está marcado por el contacto inicial, es decir, cuando se produce el apoyo del talón de una extremidad y termina cuando ese mismo talón se vuelve a apoyar, lo que también se considera el inicio del siguiente ciclo [8, 10].

La fase de apoyo tiene una duración aproximada del 62% del ciclo de marcha y consta de cuatro subetapas cuando se consideran ambas extremidades inferiores, iniciando con el primer apoyo doble, es decir, cuando ambas extremidades están en contacto con el suelo durante el inicio del ciclo, posteriormente se da el primer contacto simple, cuando hay una extremidad en apoyo y otra en balanceo, acto seguido se produce un segundo soporte doble, el cual da lugar a un segundo soporte simple pero con la extremidad contraria apoyada en el suelo, como se observa en la **Fig. 2.** Sin embargo, cuando la división se hace teniendo en cuenta una única extremidad se emplean cinco subetapas, la primera es el contacto inicial o primer contacto el cual da inicio al

ciclo de marcha, después se produce el apoyo plantar, durante el cual el pie toca el piso con toda la planta, lo que también se conoce como respuesta a la carga, puesto que la extremidad absorbe el impacto inicial, permitiendo tanto la estabilidad como la progresión del apoyo, es en este punto donde aumenta la magnitud de la fuerza de reacción del suelo. Posteriormente, se presenta el apoyo o soporte medio, durante el cual la extremidad que está en balanceo cruza a la extremidad que está apoyada en el suelo, por consiguiente se presenta la elevación del talón, por lo que este deja de tocar el suelo y da paso al despegue del pie, es ahí donde termina la fase de apoyo de esa extremidad y comienza su fase de balanceo [8, 10]**.**

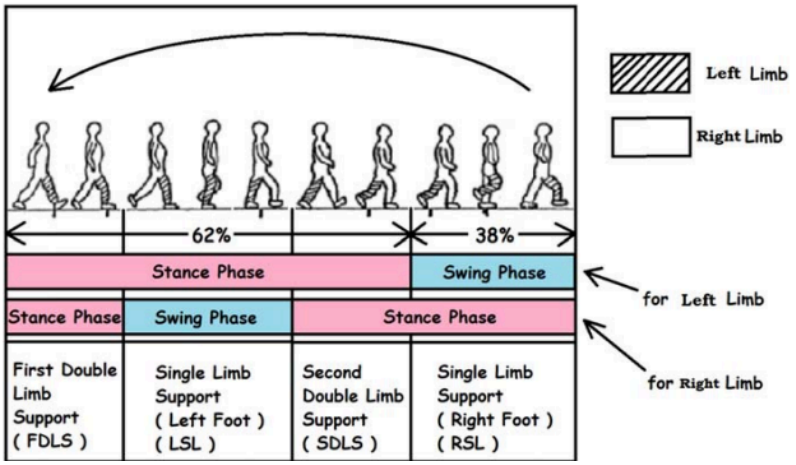


Fig. 2 Ciclo de marcha

Nota. Fuente Modern Methods for Affordable Clinical Gait Analysis: Theories and Applications in Healthcare Systems [10]

La fase de balanceo, por consiguiente, presenta una duración aproximada del 38% del ciclo de marcha y se divide en tres subetapas cuando se considera una sola extremidad (**Fig. 3**), iniciando con la aceleración o balanceo inicial, esta se presenta justo después de la elevación del pie al final de la fase de apoyo, con el fin de permitir el movimiento hacia adelante, posteriormente se produce un balanceo medio en el cual el pie en balanceo cruza al pie en apoyo, para adelantarlo, finalmente se produce la desaceleración o balanceo terminal, justo antes de dar inicio a un nuevo ciclo de marcha, en este punto se desea estabilizar de nuevo el cuerpo, por lo que se produce una desaceleración de los músculos de la extremidad [10].

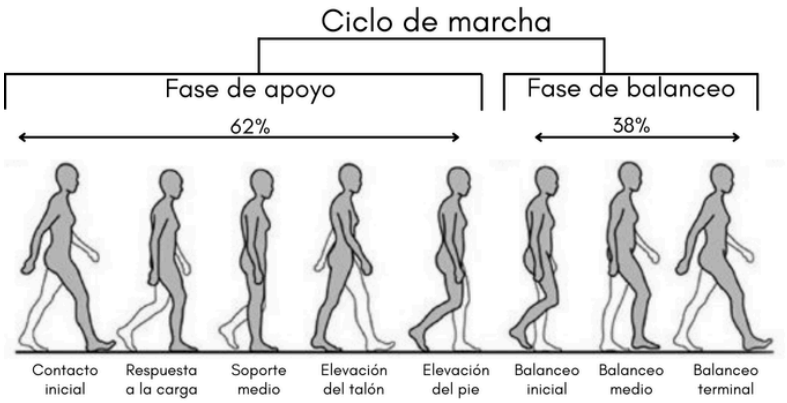


Fig. 3 Partes del ciclo de marcha

Nota. Tomado y adaptado de https://es.pinterest.com/pin/597219600619851329/

En personas con extremidades completas, estas fases y subetapas están ampliamente estudiadas en términos de duración, activación muscular, ángulos de flexión y patrones de movimiento. Sin embargo, en la marcha protésica, estas características se ven alteradas por las compensaciones particulares que los usuarios desarrollan debido a la pérdida de una extremidad, haciendo su análisis más complejo [11, 12].

OpenSim

Estudiar la biomecánica del cuerpo humano permite entender cómo se mueven y se comportan las diferentes estructuras de este bajo la aplicación de diversas fuerzas y condiciones, estos estudios son clave en diferentes campos, como la medicina, el deporte y la ingeniería, dado que permiten mejorar la eficiencia del movimiento, prevenir lesiones, diseñar dispositivos ortopédicos, mejorar el rendimiento deportivo, entre otros [13].

Actualmente, existen algunos softwares que permiten realizar este tipo de simulaciones o manipular archivos de movimiento obtenidos a partir de laboratorios de biomecánica, entre ellos se destaca el uso de OpenSim, el cual es un software de código abierto utilizado para la simulación y análisis del movimiento humano, **Fig. 4**. En el contexto de este trabajo, OpenSim desempeña un papel crucial al permitir la creación de modelos musculoesqueléticos personalizados para analizar el ciclo de marcha y evaluar la alineación protésica. El software ofrece herramientas avanzadas para realizar cinemática y dinámica inversa, lo que permite

calcular el movimiento de las articulaciones y las fuerzas internas a partir de datos experimentales, **Fig. 5**. Su capacidad para simular y analizar los movimientos del cuerpo humano en entornos controlados lo convierte en una herramienta útil para realizar estudios biomecánicos relacionados tanto con la rehabilitación, como con el uso de prótesis u órtesis [14].

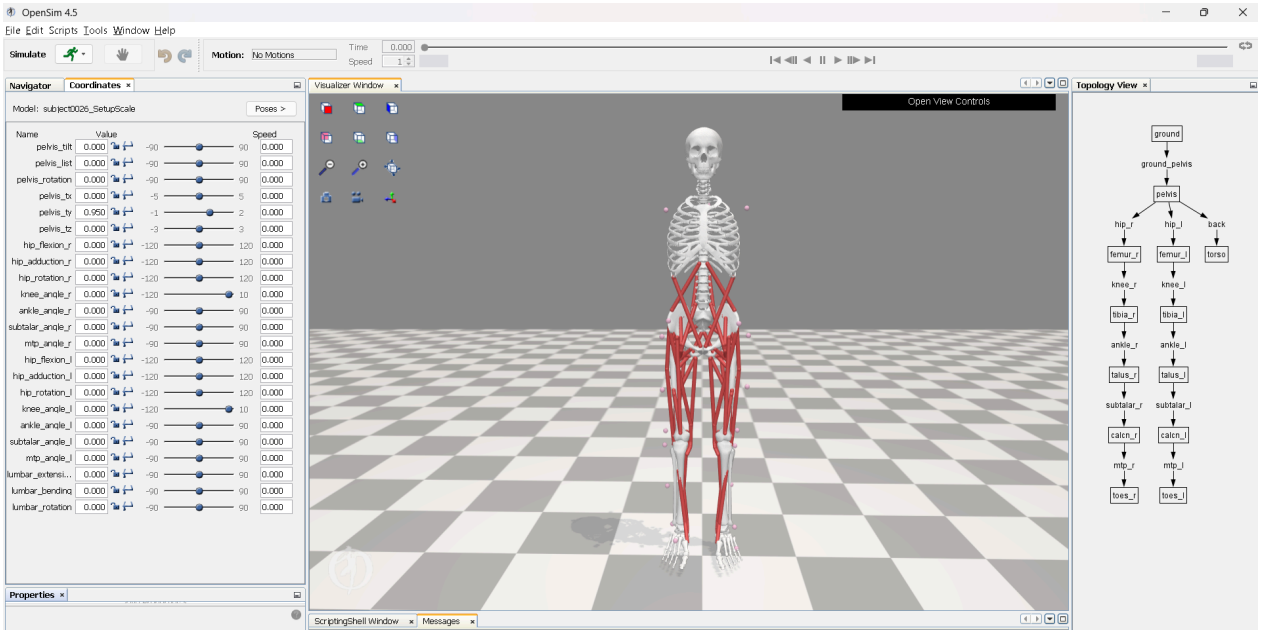


Fig. 4 Ventana principal de OpenSim

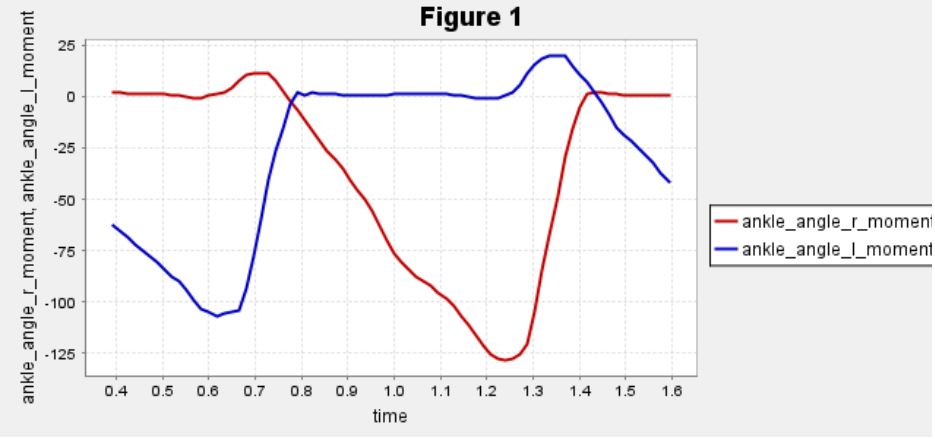


Fig. 5 Gráficas de OpenSim para el momento del ángulo de la articulación del talón

Como se ve en la **Fig. 4,** la interfaz de OpenSim permite visualizar el objeto de estudio, en este caso un modelo musculoesquelético del cuerpo humano que no contempla las extremidades

superiores y solo cuenta con 54 músculos, la complejidad de este depende del objetivo de cada investigación.

En la **Fig. 4**, se muestra la ventana principal, donde también se presentan los marcadores de dicho modelo, estos se simbolizan como pequeñas esferas rosadas ubicadas en puntos específicos, cada marcador debe tener una correspondencia con un marcador reflectivo ubicado sobre el cuerpo del sujeto que será empleado para registrar el movimiento que se intenta reproducir en la simulación, por lo que se denominan marcadores experimentales a los que están sobre la persona real y marcadores virtuales a los que están en el software. Por otro lado, en la parte izquierda de la ventana principal se presentan las coordenadas del cuerpo, cada una hace referencia a un movimiento que puede ser efectuado, como las traslaciones de la pelvis o la rotación de la cadera. Finalmente al lado izquierdo se encuentra una vista topológica que contempla una simplificación de la simulación, permitiendo comprender la jerarquía del modelo y cómo se relacionan los cuerpos entre sí, cada cajón representa un cuerpo y cada palabra entre las líneas hace relación a una articulación entre esos cuerpos.

Cinemática y cinemática inversa

La cinemática es una rama de la mecánica que se encarga de describir el movimiento de los objetos sin tener en cuenta las causas de dicho movimiento, esta se centra en el análisis de variables como la posición, la velocidad, la aceleración y el tiempo [10, 13]. En términos biomecánicos, la evaluación de las características de la cinemática se aplican sobre el cuerpo humano, como los ángulos articulares, los cuales son fundamentales dentro de un análisis de marcha dado que permiten identificar si una persona presenta un patrón de marcha normal o patológico [15].

Ahora bien, OpenSim hace uso de una herramienta denominada cinemática inversa (IK por sus siglas en inglés), donde se busca determinar cuáles son los movimientos necesarios de las partes del cuerpo de un modelo para alcanzar una posición o realizar un movimiento determinado, para esto, el software procesa para cada instante de tiempo de un registro de marcha o de movimiento la posición espacial de los marcadores y calcula las coordenadas generalizadas

que colocan al modelo en la pose que mejor se ajusta a los datos experimentales en ese instante. Para llevarlo a cabo se utiliza una ecuación de mínimos cuadrados ponderados, donde cada marcador tiene un peso que determina qué tanto se debe minimizar el error de este, es decir, la distancia entre el marcador experimental, obtenido del registro de movimiento, y el marcador del modelo, una vez se definen estos parámetros el solucionador de OpenSim emplea un criterio de convergencia de 0,0001 y un límite de 1000 iteraciones para resolver el problema y como resultado se obtiene un archivo que contiene las trayectorias de las coordenadas tanto de los ángulos articulares como de las traslaciones que fueron calculadas [16].

Dinámica y dinámica inversa

La dinámica es una disciplina que estudia no solo el movimiento y las fuerzas que lo producen sino también las interacciones entre los objetos en movimiento. Cuando se aplica en biomecánica se emplea para comprender el efecto de la fuerza y el torque que producen los músculos, huesos y ligamentos sobre el movimiento corporal, por lo que ayuda a predecir un movimiento a partir de un conjunto de fuerzas y condiciones iniciales [17].

Por otro lado, la dinámica inversa (ID por sus siglas en inglés), se suele emplear en conjunto con la cinemática para calcular las fuerzas musculares netas, las fuerzas de reacción articular y los momentos que producen un movimiento conocido. OpenSim permite realizar estudios de este tipo partiendo del movimiento obtenido durante la solución de cinemática inversa y agregando información de la cinética, es decir, los datos de cargas externas aplicadas sobre el cuerpo, los cuales son recolectados durante los registros de movimientos por diferentes sensores como plataformas de fuerza, con las que es posible determinar la magnitud y la dirección del vector de fuerza de reacción del suelo, que es fundamental para realizar análisis de marcha, pues indica la fuerza que actúa sobre el segmento del cuerpo que está en contacto con una superficie, a partir de estos archivos de entrada se calculan las fuerzas que producen el movimiento de cada articulación dentro del modelo [10, 13, 18]

Algoritmo de reducción residual

El algoritmo de reducción Residual (RRA por sus siglas en inglés), es una herramienta utilizada por OpenSim en el análisis de marcha, cuyo objetivo es minimizar las inconsistencias dinámicas en los modelos de simulación al corregir los errores que surgen de los modelos de datos, como los errores de modelado y los datos ruidosos de los marcadores. Estos errores pueden generar grandes fuerzas no físicas, conocidas como fuerzas residuales, las cuales afectan la precisión de la simulación [19].

El proceso de RRA ajusta el centro de masa del torso del modelo, permitiendo que la cinemática de este, derivada de la cinemática inversa, se alinee de manera más coherente con las fuerzas de reacción del suelo, para lo cual se realiza una simulación en pasos pequeños, donde en cada uno se calculan las fuerzas necesarias para que el modelo se desplace desde una configuración inicial hasta la posición deseada, minimizando una función objetivo. Posteriormente, los valores promedio de las fuerzas residuales se utilizan para ajustar el centro de masa del torso y corregir cualquier inclinación excesiva causada por errores en la distribución de masa y geometría. En la siguiente etapa, la cinemática ajustada se repite, con restricciones adicionales sobre las fuerzas residuales para asegurar que el movimiento generado sea realista y que las fuerzas se asignen correctamente a los músculos, lo que mejora la precisión de los resultados en el control muscular computarizado (CMC por sus siglas en inglés). Estas restricciones buscan minimizar el uso de las fuerzas residuales, permitiendo que el movimiento del modelo sea generado principalmente por los momentos articulares internos [19]..

Optimización estática

La optimización estática (OS por sus siglas en inglés), es una herramienta de OpenSim que complementa la dinámica inversa al estimar las activaciones y fuerzas musculares necesarias para reproducir un movimiento dado, asegurando el equilibrio entre posiciones, velocidades, aceleraciones y fuerzas externas en cada instante de tiempo. Se denomina "estática" porque los cálculos se realizan cuadro a cuadro, sin integrar las ecuaciones de movimiento a lo largo del tiempo. Utiliza criterios como la minimización de la suma de las activaciones musculares al cuadrado para distribuir la carga entre los músculos, lo que la hace rápida y computacionalmente eficiente, aunque sin considerar la dinámica de activación ni la elasticidad del tendón [20, 21].

Por lo cual, es una herramienta clave en el análisis de marcha y el estudio de movimientos patológicos, como en usuarios de prótesis, donde las alteraciones musculoesqueléticas pueden afectar la estabilidad y eficiencia del movimiento. A partir de datos experimentales de cinemática, cinética y electromiografía (EMG), permite estimar las fuerzas musculares involucradas en cada fase del ciclo de la marcha y analizar la compensación biomecánica [22].

Dado que el sistema musculoesquelético cuenta con más músculos de los necesarios para generar un mismo momento articular, la distribución de fuerzas musculares no es única y se obtiene mediante un proceso de optimización matemática, generalmente minimizando el esfuerzo muscular o maximizando la eficiencia mecánica. Para validar los resultados, se comparan las fuerzas optimizadas con señales EMG procesadas, las cuales proporcionan un perfil de activación relacionado con la fuerza máxima isométrica de cada músculo. Estudios previos han mostrado una correlación razonable entre ambas estimaciones, respaldando la optimización estática como un método complementario en el análisis biomecánico. Sin embargo, esta técnica no modela la dinámica de activación muscular ni la elasticidad del tendón, lo que puede generar discrepancias con los datos EMG en ciertos músculos [20, 23].

Control muscular computacional

El control muscular computacional es una herramienta utilizada en simulaciones de modelos musculoesqueléticos cuyo objetivo es calcular los niveles de excitación muscular necesarios para que las coordenadas generalizadas de un modelo sigan una trayectoria específica esto se logra mediante una combinación de control proporcional-derivativo y optimización estática, donde el primero determina las aceleraciones necesarias que llevarán las coordenadas del modelo hacia las coordenadas experimentales deseadas y el segundo distribuye las cargas entre los actuadores para lograr las aceleraciones deseadas, minimizando errores y costos energéticos. Adicionalmente, emplea una simulación de dinámica directa para aplicar los controles calculados, permitiendo avanzar en el tiempo y simular el movimiento [24].

En el análisis de la marcha, CMC es una herramienta clave para evaluar la función muscular y comprender los patrones biomecánicos. Su aplicación permite estimar las fuerzas musculares involucradas en cada fase de la marcha, analizar estrategias de compensación en sujetos con alteraciones en la locomoción y estudiar el impacto de intervenciones como el uso de

prótesis o dispositivos de asistencia. Además, su capacidad para proporcionar una estimación detallada de las activaciones musculares lo convierte en un recurso valioso para la rehabilitación y el diseño de tratamientos personalizados. Al integrar CMC en estudios de marcha, es posible mejorar la precisión del análisis biomecánico y optimizar estrategias terapéuticas para poblaciones con necesidades específicas [25].

IV. METODOLOGÍA

Para llevar a cabo el presente trabajo, se partió de una búsqueda bibliográfica en diversas fuentes, cuyo objetivo principal fue comprender tanto el funcionamiento como los requerimientos del software OpenSim. Para ello, se consultó la documentación oficial del programa, así como archivos de simulación disponibles en la base de datos SimTK, especializada en modelos biomecánicos, además, se exploraron webinars oficiales de OpenSim y artículos académicos. Esta investigación tuvo como propósito abordar aspectos clave relacionados con el flujo de trabajo del software, los formatos de archivo compatibles, las condiciones necesarias para la simulación, los antecedentes existentes en el campo y las características específicas para dar cumplimiento a cada objetivo del proyecto. La búsqueda bibliográfica se desarrolló como un proceso iterativo, dado que una vez recopilada suficiente información, se procedió a verificar el funcionamiento práctico de las simulaciones encontradas, la validez de la documentación y la eficacia de los programas recomendados. Esto permitió seleccionar los pasos más reproducibles y efectivos para ser incluidos en la guía metodológica, cómo se presenta en la **Fig. 6.**

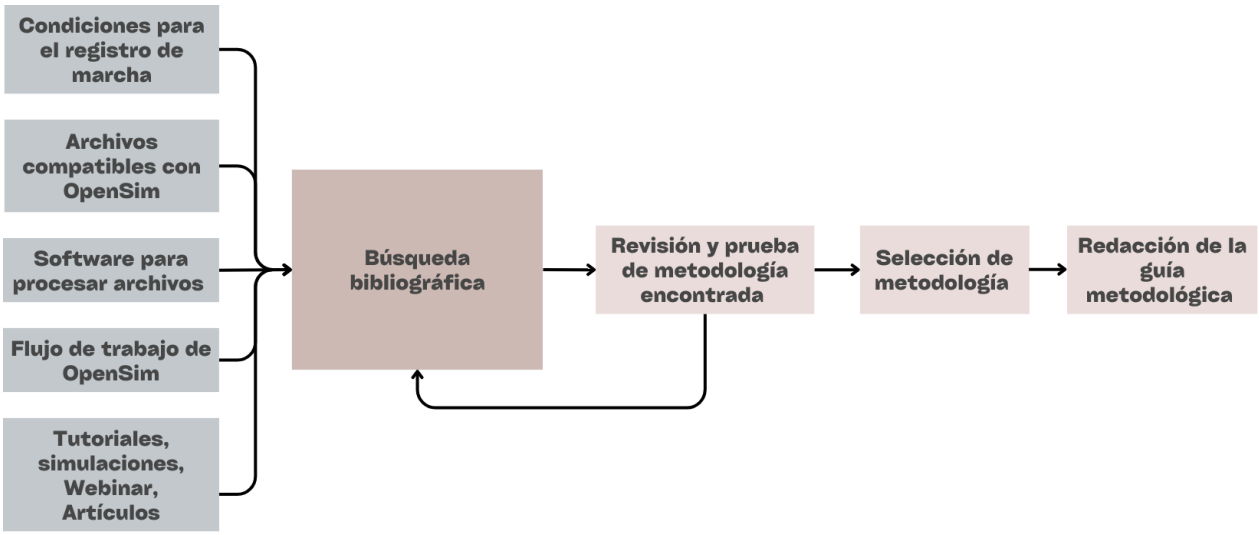


Fig. 6 Metodología

Protocolo de registro e importación de datos de marcha protésica en OpenSim.

Una vez se contó con suficiente información bibliográfica sobre las consideraciones necesarias para realizar registros de marcha orientados a simulaciones en OpenSim, fueron

condensadas en el primer apartado de la guía metodológica. No obstante, dentro del alcance de este trabajo no se contempló la realización de nuevos registros de marcha, dado que se emplearon aquellos desarrollados previamente por [26], por lo que también fue necesario revisar la información disponible relacionada con el sujeto de prueba, como datos antropométricos y ubicación de los marcadores.

Posteriormente, se estudió el flujo de trabajo del software y se verificaron las condiciones requeridas para la importación de datos de marcha en este, para ello, se probaron diversas aplicaciones y recomendaciones identificadas durante la búsqueda bibliográfica. Estas herramientas y métodos estaban orientados a la conversión de formatos de los archivos de registro a extensiones compatibles con OpenSim, así como a la extracción de información clave contenida en los archivos, necesaria para que el software pueda realizar simulaciones de manera adecuada. Por consiguiente, se condensó la información encontrada en la guía metodológica y se continuó con la especificación de los criterios que deben considerarse antes de cargar los archivos en OpenSim, entre estos se incluyeron instrucciones sobre la modificación de los registros, como el ajuste de la duración de los mismos y la edición de marcadores experimentales. Además, se añadieron apartados sobre la selección y edición del modelo base en OpenSim, abarcando procesos como la creación, edición, eliminación y exportación de conjuntos de marcadores virtuales.

Flujo de trabajo para el análisis de marcha protésica en OpenSim.

Para esta sección se desarrolló una búsqueda bibliográfica que permitiera comprender cómo ejecutar diversas simulaciones a partir de las cuales se puede realizar el análisis de marcha de un sujeto, teniendo en cuenta el flujo de trabajo de OpenSim. Posteriormente, se pusieron a prueba los métodos encontrados empleando tanto los ejemplos de OpenSim, como los datos del laboratorio de marcha para verificar que las metodologías encontradas pudieran aplicarse a archivos de diversas procedencias, una vez se verificó el funcionamiento de las herramienta de escalado, IK, ID, RRA, OS y CMC con los archivos existentes, se procedió a describir un proceso detallado sobre cómo ejecutar estás simulaciones, agregando apartados que comprenden desde la descripción general del proceso, pasando por la creación y edición de los archivos de configuración de cada uno, la ejecución de la herramienta y la validación de los resultados.

Posteriormente, la búsqueda de información se centró en cómo ejecutar simulaciones relacionadas con amputación de extremidad inferior, se revisaron los códigos de simulaciones realizadas en estudios que emplean OpenSim para comprender cómo se deben de modificar los archivos que componen un modelo. Adicionalmente se realizó el diseño simplificado de una prótesis transfemoral y se puso a prueba la información encontrada, una vez se encontró la metodología adecuada para llevar a cabo los cambios requeridos para hacer esta simulación se agregó a la guía metodológica y se incluyeron todos los aspectos clave relacionados como modificaciones óseas, musculares y articulares, además de importación de geometría externa. Adicionalmente, para verificar el funcionamiento del nuevo modelo se implementaron nuevamente las herramientas de escalado, IK, ID, RRA, OS y CMC con los mismos archivos aplicados en el modelo sin amputación.

Evaluación de la calidad de los protocolos propuestos

Una vez se consideró suficiente la información contenida en la guía metodológica se realizó un cuestionario que permitiera evaluar la calidad de la misma, el cuál constaba de 69 preguntas, distribuidas en secciones correspondientes con la organización de la guía, este cuestionario contempla preguntas tanto cerradas como abiertas y escalas de valoración, además de proporcionar un espacio para realizar observaciones a cada una de ellas y comentarios generales sobre la sección que se está evaluando. El proceso de evaluación se llevó a cabo por una estudiante de último semestre de Bioingeniería de la Universidad de Antioquia, a la cual se le entregaron los archivos necesarios para la realización del proceso, la carpeta contenía la geometría protésica, los archivos de la prueba estática y el registro de marcha, la documentación con la información antropométrica del sujeto de prueba, la guía metodológica y el formulario de evaluación.

V. ANÁLISIS DE RESULTADOS

Protocolo de registro e importación de datos de marcha protésica en OpenSim.

El primer apartado de la guía metodológica reúne un conjunto de consideraciones y recomendaciones básicas a tener en cuenta antes y durante un registro de marcha, las cuales fueron recopiladas a partir de la revisión bibliográfica llevada a cabo a lo largo del proyecto. Si bien no se realizó una verificación práctica de este protocolo inicial, su importancia quedó evidenciada al enfrentar dificultades en el manejo de los archivos debido a la ausencia de estas consideraciones en los datos previamente recolectados, resaltando la relevancia de incluir pautas claras para el registro cuando el propósito contempla realizar simulaciones en OpenSim. Ahora bien, la ausencia del proceso de verificación se debe a que el enfoque del proyecto no pretendía realizar nuevas grabaciones, sino hacer uso de archivos existentes, permitiendo identificar los retos asociados al manejo y análisis de datos obtenidos en estudios externos.

Posteriormente, se identificó el flujo de trabajo del software, el cual está definido para realizar simulaciones a partir de archivos que contengan trayectorias de marcadores y plataformas de fuerza, **Fig. 7**, este comienza con la recolección de registros de movimiento que pueden provenir de un laboratorio o de bases de datos, generalmente contiene la ubicación espacial y la trayectoria de una cantidad específica de marcadores experimentales ubicados sobre un sujeto de prueba en una posición de interés. Luego se debe realizar un proceso de escalado, el cual permite ajustar las medidas de un modelo para hacer coincidir los parámetros antropométricos de este con los de un sujeto de estudio [27], después de este proceso se realiza IK e ID, donde el primero permite posicionar los marcadores para cada instante de tiempo del registro sobre el modelo y calcula las coordenadas generalizadas que colocan al modelo en la posición que mejor se ajusta a los datos experimentales en ese instante [16], mientras que el segundo permite obtener las fuerzas netas, así como los torques que producen movimiento en cada articulación [18]. No obstante, otra forma de realizar dinámica inversa es ejecutar RRA esta herramienta mejora la precisión de la simulación al reducir los residuos dinámicos de fuerzas y momentos no fisiológicos, es decir, busca ajustar el centro de masa del torso permitiendo variar la cinemática del modelo a partir de los datos de cinemática inversa, para que este sea dinámicamente coherente con los datos de fuerza de reacción del suelo [19]. Seguidamente, se debe llevar a cabo un proceso de

optimización estática, el cual permite ampliar el enfoque de la dinámica inversa, al calcular cómo los momentos articulares netos se distribuyen entre las fuerzas de los músculos individuales en cada instante de tiempo [20]. Finalmente se realiza una simulación de CMC para estimar las activaciones musculares requeridas para reproducir un movimiento determinado, resolviendo la redundancia neuromuscular al distribuir las fuerzas entre los músculos de manera eficiente, respetando las restricciones fisiológicas del sistema musculoesquelético [25].

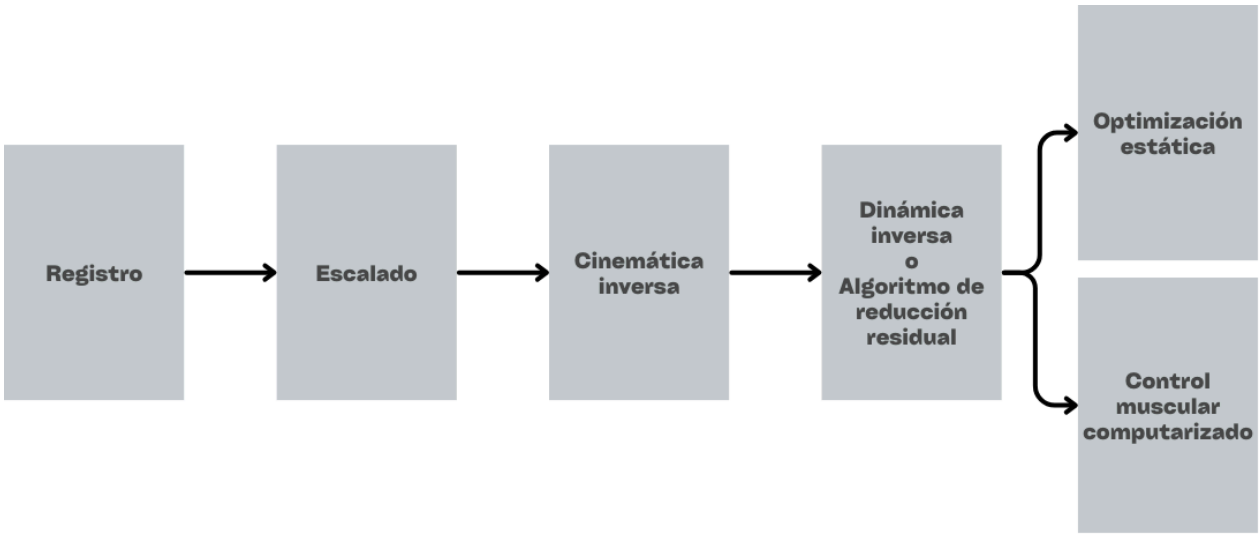


Fig. 7 Flujo de trabajo de OpenSim

Durante el análisis del flujo de trabajo de OpenSim se halló que este software trabaja principalmente con dos tipos de archivos externos, los archivos de rastreo (.trc) y los archivos de movimiento (.mot), en los primeros se almacenan las coordenadas de los marcadores experimentales durante una captura de movimiento, dado que contienen las posiciones de estos marcadores en X, Y y Z a lo largo del tiempo, mientras que los segundos poseen los datos de movimiento o fuerzas aplicadas en función del tiempo, como ángulos articulares, velocidades, aceleraciones, fuerzas o momentos. Por lo que fue necesario comprender cómo obtener estos archivos puesto que generalmente los repositorios de registros de movimiento trabajan con otros formatos.

Como se mencionó anteriormente, los registro de marcha protésica se obtuvieron de un estudio previo, el cual pretendía desarrollar un protocolo para apoyar al protesista en la alineación

estática y dinámica de la prótesis, vinculando las variables biomecánicas del uso de prótesis transfemorales con rodilla poliaxial [26], por lo cual era necesario revisar la información con la que se contaba, es decir, tanto los registros como los datos del sujeto de prueba, dado que este estudio no contemplaba el uso de los archivos para realizar simulaciones en OpenSim, al ejecutar este proceso se identificaron diferentes dificultades, primero, los archivos debían ser convertidos del formato original (.tdf) a un formato que pudiera ser interpretado por OpenSim (.trc y .mot), sin embargo, esta conversión no puede realizarse de modo directo, sino que primero deben pasar por un formato intermedio (.c3d), el cual es un formato usado en análisis biomecánicos para capturar información de movimiento 3D, plataformas de fuerza, electromiografía y diferentes sensores [28], por lo que se empleó MOKKA, el cual es un software libre que permite analizar diferentes tipos de archivos que contienen datos biomecánicos en 3D y 2D, como trayectorias de marcadores, señales de EMG, plataformas de fuerza, entre otros [29], para pasar del formato original al formato intermedio y luego se emplearon diferentes software (Matlab, MOKKA y C3D Tools) para intentar extraer los archivos soportados por OpenSim a partir del formato intermedio, sin embargo, no fue posible extraer adecuadamente el archivo que contenía la información de la plataforma de fuerza (.mot).

Inicialmente se puso a prueba una librería de OpenSim para Matlab en la cual hay dos archivos que permiten procesar la información de archivos .c3d para obtener tanto los datos de trayectorias de marcadores como de plataformas de fuerza. No obstante, al emplear ambos códigos estos no transcriben adecuadamente la información de la plataforma de fuerza. En la **Fig. 8a,** se ve como debería ser el trazado de los vectores, un punto de origen diferente para cada magnitud registrada por la plataforma, este trazado se obtuvo de un archivo de registro del repositorio de C3D.org, que no estaba relacionado con marcha protésica y que se procesó con C3D Tools para obtener el archivo de la plataforma de fuerza, mientras que **Fig. 8b**, se presenta el resultado de procesar este mismo archivo con Matlab, como se observa, este redujo todas las componentes a un único punto de origen y una única magnitud. Por otro lado, la **Fig. 8c,** presenta el resultado de Matlab para los archivos del sujeto de prueba de interés, aunque no redujo todas las componentes en un único vector, si les asignó el mismo punto de origen y adicionalmente rotó las coordenadas, por lo que no es posible usar este archivo pues los resultados que produciría serían erróneos. En segundo lugar, se consideró usar MOKKA, pero, a pesar de que este software

permitió ver el trazado de los vectores de la plataforma de fuerza, **Fig. 9,** no permitió extraer el archivo correspondiente, por lo cual se descartó su uso para esta finalidad, no obstante, si se empleó para pasar del formato .tdf a .c3d y de este extraer los archivos tanto del registro de la prueba estática como de marcha. Finalmente, se empleó C3D Tools, el cual es un servicio digital y gratuito que permite no solo visualizar archivos en formato .c3d sino que también brinda la posibilidad de extraer los archivos de trayectorias y plataformas de fuerza en los formatos requeridos por OpenSim, .trc y .mot respectivamente, sin embargo, a pesar de que funcionó con archivos extraídos de un repositorio digital, no fue posible obtener los datos de la plataforma de fuerza del archivo en el formato intermedio (.c3d), previamente extraído de MOKKA pues indicaba que este no cumplía con las condiciones necesarias para extraer la información [30]. Teniendo en cuenta lo anterior, de los archivos proporcionados por el laboratorio solo se pudo obtener la trayectoria de los marcadores tanto de la prueba estática como del registro de marcha.

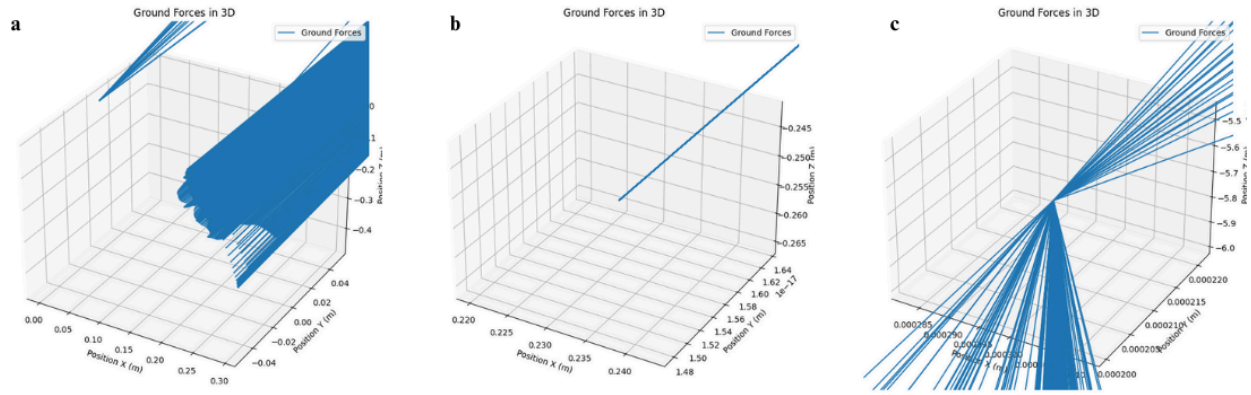


Fig. 8 Trazado del vector de fuerza de reacción del suelo.

a) Trazado correctamente. b) Vector obtenido de emplear Matlab para el archivo del vector a. c)Vector obtenido de emplear Matlab para el archivo del sujeto de prueba

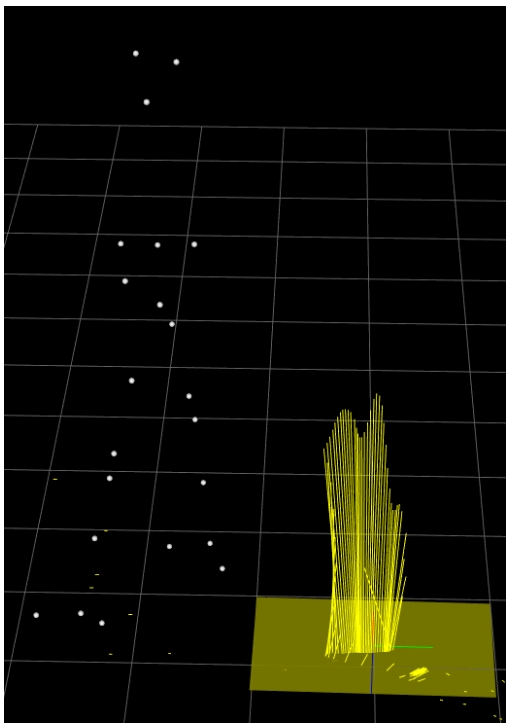


Fig. 9 Trazado del vector de fuerza de reacción del suelo obtenido con MOKKA

Nota. Las líneas amarillas son el trazado del vector

Por lo cual, fue necesario exportar directamente desde el laboratorio de registro de marcha los archivos en formato .c3d y no en el formato propio del software del laboratorio (.tdf), una vez se contó con estos, se empleó C3D Tools para obtener los archivos de movimiento del registro de marcha y de la trayectoria de los marcadores experimentales del registro de la prueba estática y de marcha. Posteriormente se verificó que el trazado del vector fuese adecuado en términos de orientación y componentes, como se presenta en la **Fig. 10.**

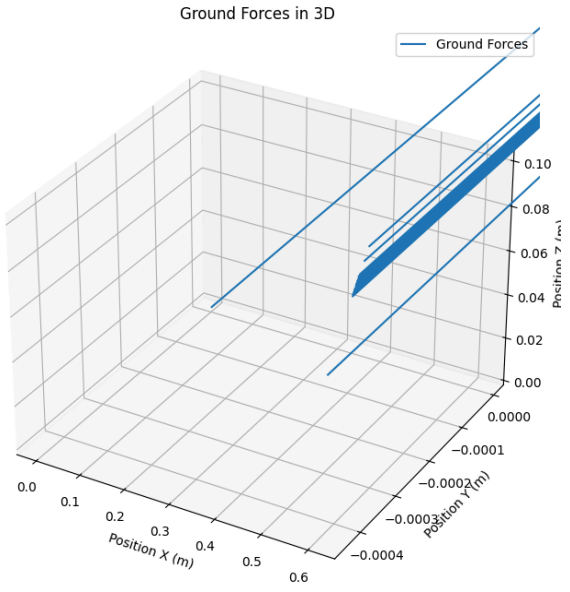


Fig. 10 Vector de fuerza de reacción del suelo del sujeto de prueba

Finalmente, una vez se lograron obtener los archivos en los formatos adecuados y verificar el funcionamiento de las diferentes opciones halladas en la literatura, se condensó la información en la guía metodológica, especificando los procesos necesarios para llevar a cabo las conversiones de formato.

Flujo de trabajo para el análisis de marcha protésica en OpenSim.

Las herramientas que ofrece OpenSim (Escalado, IK, ID, RRA, SO y CMC), requieren el uso no solo de archivos externos, sino también de archivos de configuración específicos para cada una, como se observa en la **Fig. 11**, en la cual se presentan los datos de entrada y salida de cada una y se ilustra cómo a partir de los resultados de un estudio es posible continuar con el siguiente, es por ello que, en aras de completar el segundo objetivo, se analizó a profundidad cómo emplear cada una de ellas.

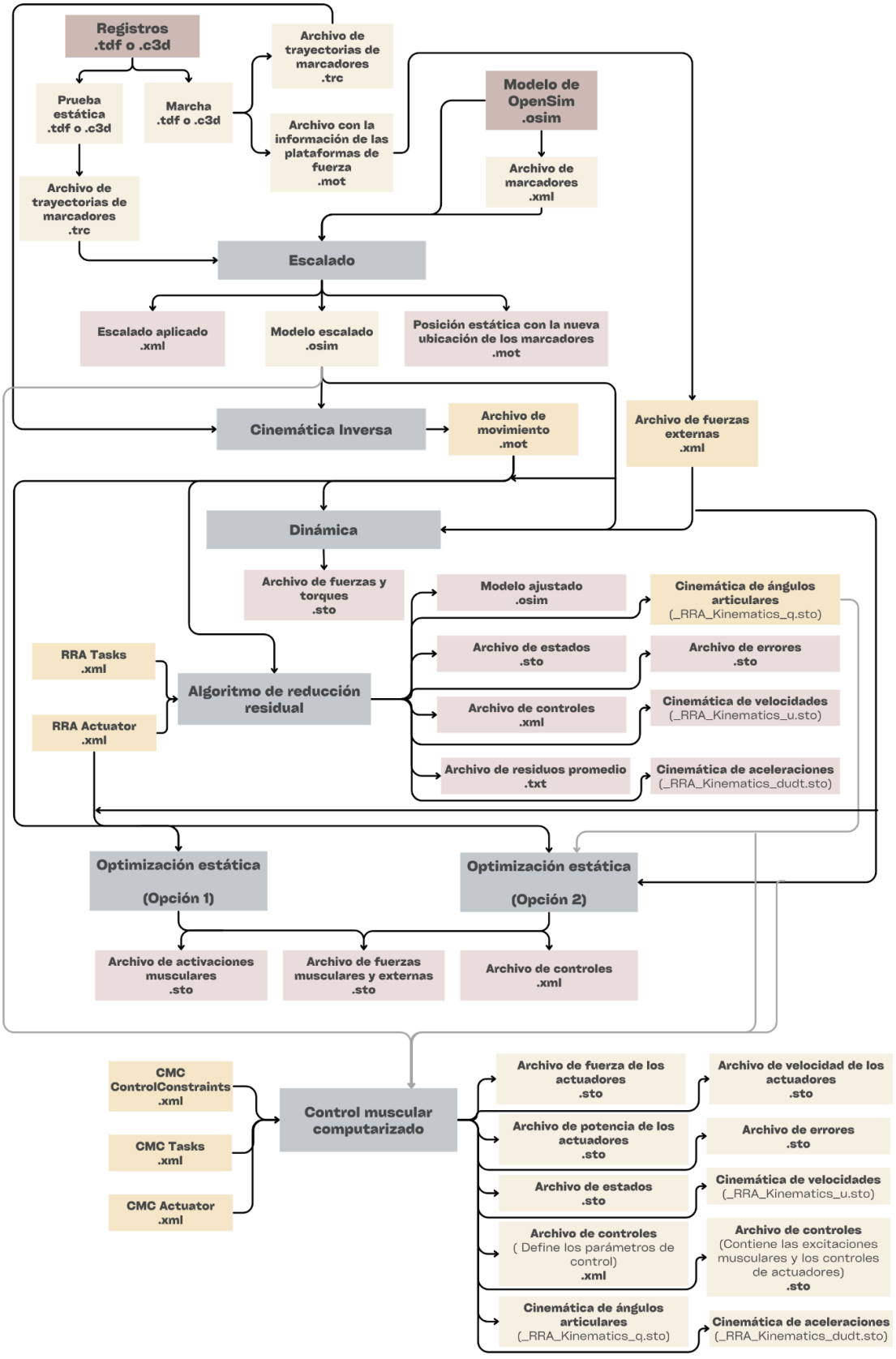


Fig. 11 Detalles del flujo de trabajo de OpenSim

Al estudiar los requisitos para ejecutar las herramientas de OpenSim, se verificó si los archivos extraídos en la sección anterior cumplían con estos criterios y se exploró cómo modificarlos según fuera necesario. Por ello, se añadieron apartados a la guía metodológica con instrucciones sobre la edición de los registros de marcha, incluyendo la duración del registro, el renombrado de marcadores experimentales, la selección y edición del modelo, así como la gestión de marcadores virtuales (creación, edición, eliminación y exportación). Esta revisión resaltó la importancia de los aspectos abordados en la sección de registro de marcha, dado que se requirieron múltiples ajustes tanto en el modelo base de OpenSim como en los archivos de captura de movimiento, debido a que el estudio del cual provenían los datos utilizados en este proyecto no fue realizado considerando las especificaciones de OpenSim, puesto que su propósito era diferente.

Una vez que los archivos de trayectorias de los marcadores experimentales cumplían con las condiciones necesarias para su uso, se procedió a realizar las simulaciones de escalado y cinemática inversa para el modelo. El proceso de escalado se verificó realizando mediciones sobre el modelo de OpenSim, las cuales se compararon con los datos antropométricos del sujeto de prueba, en la **TABLA I**, se presentan las medidas reales y las del modelo, así como el porcentaje de error entre estos.

TABLA I
EVALUACIÓN DE ESCALADO

Modelo	**Referencia antropométrica**	**Medida del sujeto (cm)**	**Medida del modelo (cm)**	**Error (%)**
Modelo sin amputación	Ancho de la pelvis	21.3	24	12.7%
	Longitud de la pierna	90	84	6.6%
	Altura del sujeto	166	167	0.6%
Modelo con amputación	Ancho de la pelvis	21.3	25	17.3%
	Longitud de la pierna	90	86	4.4%
	Altura del sujeto	166	167	0.6%

El mayor porcentaje de error en ambos modelos se presentó en el ancho de la pelvis, el cual tuvo una diferencia de 2.7cm y 3.7cm para el modelo sin amputación y el modelo con amputación respectivamente, posiblemente por la ubicación de los marcadores en el modelo original, dado que la posición de estos se realizó empleando imágenes de las posiciones de los marcadores en el sujeto de prueba, las cuales no permiten identificar adecuadamente la posición del marcador en la zona pélvica, más allá de conocer que estaban sobre alguna parte de la cresta iliaca. Cabe destacar que este porcentaje de error se puede reducir si se usan factores de escala manual para las zonas que más se le dificultan al software, sin embargo, el propósito de la guía era describir el proceso de forma sencilla, por lo que el escalado manual no fue descrito en la misma, sino sólo el proceso automático.

Por otro lado, para el proceso de cinemática inversa, los resultados se evaluaron verificando que los marcadores no tuvieran un valor de RMS (Raíz cuadrática media) mayor a 2cm, ni un error de marcador (Error promedio por marcador) superior a 4 cm, que es lo recomendado por OpenSim, estos valores son arrojados por el Software para cada uno de los fotogramas de la simulación, indicando cuál ha sido el marcador que ha presentado mayor error en cada iteración [31].

Posterior a la verificación de escalado y cinemática inversa se agregaron las secciones correspondientes en la guía metodológica, incluyendo información sobre cómo crear los archivos de configuración, cómo ejecutar la herramienta y cómo validar los resultados, además se añadieron imágenes que guían al lector para facilitar la comprensión del proceso.

Siguiendo el flujo de trabajo del software, el paso siguiente consistía en emplear la herramienta de dinámica inversa, por lo que se probó la documentación encontrada respecto a cómo ejecutarla, este proceso permitió comprender por qué es importante considerar el propósito con el que se van a emplear estudios de captura de movimiento, antes de desarrollarlos, dado que durante la validación de la información encontrada se presentaron diferentes dificultades relacionadas con los datos proporcionados por la plataforma de fuerza, puesto que es necesario considerar desde un inicio que para hacer análisis de marcha con OpenSim no es suficiente con una única plataforma de fuerza, sino que se recomienda el uso de varias, con el fin de capturar las componentes de la fuerza de reacción del suelo de ambas extremidades y mejorar el cálculo del programa [32, 33], sin embargo, los datos empleados para esta simulación sólo contenían la

información de una plataforma de fuerza, por lo que solo se capturaban las componentes del vector de una extremidad. Como resultado de dinámica inversa se produjo un archivo que contenía los pares y fuerzas de las articulaciones netas que actúan para producir el movimiento experimental medido y las fuerzas externas aplicadas, a partir de los cuáles es posible realizar gráficas de Momento Vs Tiempo de cada una de las articulaciones del modelo, no obstante, al no contar con la información de ambas extremidades el programa no puede calcular adecuadamente y por tanto no es posible emplear estos resultados para estudiar los momentos del sujeto de prueba.

Ahora bien, OpenSim no cuenta con una sola herramienta para realizar dinámica inversa, sino con dos, la que se describió previamente y RRA, esta permite calcular de manera más precisa la dinámica del modelo, no obstante, se debe de contar con un archivo que capture el vector de fuerza de reacción del suelo de cada extremidad, al igual que en ID, pero como se explicó previamente, los registros se tomaron de otro estudio y éste sólo consideró el registro de una extremidad, puesto que era suficiente para el objetivo del mismo, por lo cual, a pesar de que fue posible emplear la simulación, no se obtuvieron resultados adecuados, dado que las fuerzas residuales eran superiores a lo que se reporta en la guía de buenas prácticas de OpenSim, como se presenta en la **TABLA II,** [19].

TABLA II
RESULTADOS DE RRA

Parámetros	**Valor de referencia Bueno**	**Valor de referencia Aceptable**	**Valor de referencia Malo**	**Valores obtenidos (XYZ)**
Fuerza residual máxima (N)	0 - 10	10 - 20	> 25	19, 291, 13
Momento residual máximo (Nm)	0 - 50	50 - 75	>75	-5, -11, -83

Sin embargo, a pesar de los resultados obtenidos tanto en RRA como en ID, se condensó en la guía metodológica la información necesaria para ejecutar ambas herramientas, incluyendo información relevante a partir de las complicaciones encontradas durante el proceso de implementación de las mismas, dado que el presente trabajo no pretende analizar la marcha del

sujeto como tal, sino proporcionar una guía que permitan comprender cómo ejecutar las herramientas de OpenSim, es por esto que se documentó todo el procedimiento de creación de archivos, ejecución y validación de resultados para ambas.

Por otro lado, a partir de los resultados obtenidos en dinámica inversa, es posible realizar una simulación de optimización estática, en la cual se usa el movimiento del modelo para resolver las ecuaciones de movimiento para las fuerzas generalizadas desconocidas, como pares de torsión de las articulaciones. De este estudio se producen archivos que permiten graficar las activaciones musculares a partir de la solución proporcionada por el software empleando los resultados de RRA o de ID para calcularlos. La revisión bibliográfica realizada para comprender el funcionamiento de la herramienta, así como la prueba del funcionamiento de esta empleando los archivos del sujeto y los resultados obtenidos en las simulaciones previas permitieron describir el proceso de ejecución y verificación de la misma dentro de la guía metodológica.

Finalmente, los análisis de marcha en OpenSim concluyen con el uso de la herramienta de CMC, la cual emplea los resultados de cinemática que se obtienen a partir de RRA. A pesar de que la solución de OpenSim para el sujeto de prueba empleado en este estudio no es adecuada, si permite analizar los parámetros necesarios para ejecutar CMC, por lo que se describió en la guía el proceso de modificación y creación de archivos, verificación de resultados y corrección de activaciones musculares para mejorar la solución de CMC, con el fin de que el lector pueda emplear la herramienta y analizar la calidad de los resultados obtenidos.

Por otro lado, uno de los propósitos del presente trabajo era explorar la posibilidad de simular una amputación transfemoral en OpenSim y documentar la metodología para llevarla a cabo. Tras una revisión bibliográfica, se identificaron diversos estudios que desarrollan modelos de amputación, tanto transtibial como transfemoral.

Por ejemplo, en [5], se buscó estimar el movimiento y la carga entre el encaje y el miembro residual como parte del protocolo estándar de análisis de marcha protésica transtibial. Mientras que en [34], se planteó el desarrollo de una prótesis transtibial biarticular y cuasipasiva, cuya validación se realizó en OpenSim mediante la creación de un modelo genérico de amputación, el cual se ofrece como recurso libre para simulaciones de análisis de marcha.

Asimismo, en [4], se diseñó un modelo genérico en OpenSim para personas con amputación transfemoral y prótesis osteointegrada, con el fin de analizar la biomecánica de la marcha.

Si bien estos estudios fueron desarrollados en OpenSim y algunos proporcionan los archivos de simulación, en los artículos publicados no se encuentra una descripción detallada sobre cómo generar un modelo propio. En su lugar, sólo incluyen información general sobre aspectos clave como la eliminación y modificación de grupos musculares, la reubicación del centro de masa y del momento de inercia de los nuevos segmentos corporales, los movimientos articulares permitidos, los grados de libertad y las consideraciones necesarias para la colocación del grupo de marcadores de registro y escalado de sus propios modelos. Además, estos estudios señalan que para realizar simulaciones en OpenSim con herramientas como ID, RRA, OS y CMC, es imprescindible registrar las fuerzas de reacción del suelo de ambas extremidades.

A partir de la información recopilada, se estudiaron las simulaciones disponibles para comprender las modificaciones necesarias al desarrollar un modelo de amputación transfemoral con prótesis no osteointegrada. Como resultado, se registró un protocolo detallado que documenta cada paso del proceso. Adicionalmente, se incluyeron explicaciones sobre las libertades que el usuario puede tomar para personalizar la simulación de acuerdo con sus objetivos. Finalmente, se obtuvo un modelo de OpenSim que contiene tres cuerpos externos pertenecientes a una simplificación de una prótesis transfemoral (encaje, vástago y pie), no se incluyó un elemento para la rodilla dado que no se pretendía probar ni diseñar una prótesis comercial, por lo que el movimiento articular se describe y el peso y geometría del componente se simplifican en el vástago. Además, se lograron describir cuáles son las consideraciones y modificaciones necesarias para eliminar no solo los componentes óseos sino también los musculares, además de realizar la reasignación de articulaciones y marcadores en el modelo, guiando al lector por medio de imágenes y demás recursos que le ayudan a comprender cómo realizar el proceso en OpenSim.

En la **Fig. 12,** se presenta el modelo diseñado, donde se aprecia la amputación transfemoral izquierda y la geometría protésica, además de los movimientos de flexoextensión tanto de rodilla como de tobillo.

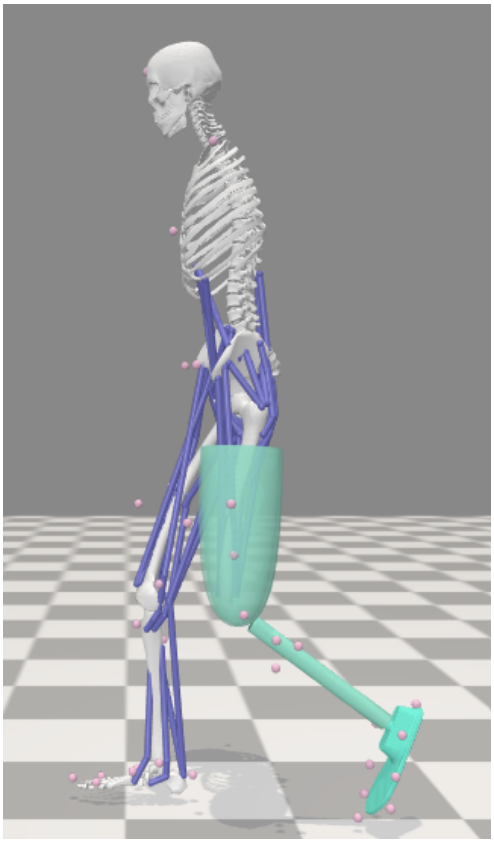


Fig. 12 Modelo de amputación transfemoral izquierda

Con el modelo amputado se procedió a realizar las simulaciones de escalado y cinemática inversa efectuando las modificaciones necesarias para considerar la nueva geometría y se obtuvo el porcentaje de error del proceso de escalado (**TABLA I)**, además se verificó que la cinemática inversa cumpliera los parámetros previamente mencionados.

Adicionalmente, se realizó ID y RRA, a pesar de no contar con el vector de fuerza de reacción del suelo para ambas extremidades, no con el propósito de evaluar la calidad de los resultados obtenidos en términos de análisis de marcha, sino para verificar que se pudieran emplear las herramientas sin inconvenientes debidos a las modificaciones realizadas en el modelo. Cabe destacar que fue necesario modificar los archivos en función de incluir todos los componentes nuevos y eliminar aquellos que ya no pertenecían al modelo.

Evaluación de la calidad de los protocolos propuestos

La guía metodológica desarrollada consta de 9 secciones que comprenden las recomendaciones para el registro de marcha, la preparación de archivos, el uso de las herramientas de escalado, cinemática inversa, dinámica inversa, algoritmo de reducción residual, optimización estática y control muscular computacional, además de la última sección dedicada a la simulación de amputación transfemoral, en cada uno de estos apartados se presenta la descripción del mismo, los archivos de configuración en caso de ser necesarios y las instrucciones para ejecutar la herramienta y analizar los resultados.

Como último objetivo se propuso evaluar la calidad de los protocolos mediante la retroalimentación de usuarios con conocimiento en el manejo de OpenSim, tras el período de evaluación, se recolectaron y analizaron las respuestas del formulario de retroalimentación. Cabe mencionar que se excluyó la evaluación práctica de la sección inicial relacionada con las recomendaciones para realizar los registros de marcha, debido a la falta de condiciones para su implementación; por ello, esta sección se evaluó únicamente en términos de claridad en la descripción.

El análisis del formulario de evaluación reflejó un puntaje promedio de 4.78 sobre 5 en nivel de claridad para el total de secciones, donde 5 correspondía a "muy claro" y 1 a "poco claro". Adicionalmente, el tiempo requerido por el evaluador para completar todas las secciones de la guía fue de aproximadamente 11 horas, lo que quiere decir que a pesar de que este contaba con conocimientos previos del software, el proceso de aprendizaje y aplicación al análisis de marcha implica una curva de aprendizaje considerable. No obstante, el tiempo necesario para ejecutar las simulaciones a partir de la información proporcionada en la guía, es menor en comparación con el tiempo que tomaría buscar información en diversas fuentes sobre el uso de las herramientas y el análisis de los resultados. Esto resalta la utilidad de la guía al consolidar en un solo documento el proceso completo de simulación y análisis.

Con base en las sugerencias obtenidas en la evaluación, se realizaron ajustes a la guía metodológica, como la inclusión de una sección que lista los softwares a emplear durante la ejecución de la misma, la clarificación de algunas secciones y el anexo de más imágenes que mejoren la descripción del proceso. Adicionalmente se desarrollaron videos complementarios para cada sección, con el objetivo de mejorar la claridad del contenido y reforzar el apoyo visual.

VI. CONCLUSIONES Y RECOMENDACIONES

- Se logró establecer una metodología detallada para el registro y el análisis de la marcha protésica en usuarios de prótesis de miembro inferior, así como la simulación de amputación transfemoral, utilizando OpenSim. Esta metodología permite un flujo de trabajo claro desde la adquisición de datos hasta la simulación y el análisis.
- Se observó que los modelos generados en OpenSim son sensibles a la calidad de los datos utilizados. Por lo tanto, la precisión de los resultados depende en gran medida de la correcta captura y procesamiento de la información experimental.
- A pesar de la utilidad de OpenSim para el análisis biomecánico, se identificaron limitaciones, como la necesidad de ajustes manuales en el escalado de los modelos y la dependencia de datos experimentales de alta calidad. Además, la curva de aprendizaje del software puede representar un desafío para nuevos usuarios.
- La metodología desarrollada tiene el potencial de mejorar la evaluación y comprensión de la biomecánica en usuarios de prótesis de miembro inferior, proporcionando una base para futuras investigaciones con distintos enfoques y sirviendo como material educativo para la formación de profesionales en el área.
- Como trabajo futuro se propone poner en práctica la guía desarrollada empleando registros de marcha cuya adquisición parta de seguir las recomendaciones descritas en la guía podría mejorar el proceso de validación de esta..

REFERENCIAS

[1] B. Yuan, D. Hu, S. Gu, S. Xiao y F. Song, "The global burden of traumatic amputation in 204 countries and territories." *Frontiers in public health,* vol. 11, no. 1258853. 2023, doi: https://doi.org/10.3389/fpubh.2023.1258853

[2] DANE, "Estado actual de la medición de la discapacidad en Colombia." 2022

[3] C. A. Silva Castellanos, *Modelamiento de la marcha humana con prótesis de miembro inferior mediante herramientas de simulación dinámica ("una aplicación en OPENSIM").* [tesis de maestría]. Bogotá (Colombia): Universidad Nacional de Colombia, 2015.

[4] V. Raveendranathan, V. G. M. Kooiman y R. Carloni R, "Musculoskeletal model of osseointegrated transfemoral amputees in OpenSim." *PLoS ONE*, vol. 18, no.9, 2023, doi: https://doi.org/10.1371/journal.pone.0288864

[5] A. K. LaPrè, M. A. Price, R. D. Wedge, B. R. Umberger y F. C. Sup, "Approach for gait analysis in persons with limb loss including residuum and prosthesis socket dynamics." *International journal for numerical methods in biomedical engineering,* vol. 34, no. 4, 2018, doi: https://doi.org/10.1002/cnm.2936

[6] A. I. Vásquez Hincapié, "Dispositivo de alineación para prótesis transfemorales," Universidad de Antioquia, Medellín, 2022

[7] M. L. Ocampo, L. M Henao y L. Vásquez, "Amputación de miembro inferior," *Documentos de investigación. Facultad de Rehabilitación y Desarrollo Humano*. no. 42, 2010

[8] T. Dewees, "Transtibial Prosthetics," *en Orthotics and Prosthetics in rehabilitation,* 4ta edición, 2019, pp. 605-634

[9] J. E. Edelstein y K. K. Chui, "Transfemoral Prostheses," *en Orthotics and Prosthetics in rehabilitation,* 4ta edición, 2019, pp. 635-653

[10] A. Nandy, S. Chakraborty, J. Chakraborty y G. Venture, "Introduction," en *Modern Methods for Affordable Clinical Gait Analysis : Theories and Applications in Healthcare Systems*, G. Capille, Eds. Academic Press, 2021, pp.1-3.

[11] C. A. Silva Castellanos, "Modelamiento de la marcha humana con prótesis de miembro inferior mediante herramientas de simulación dinámica ("Una aplicación en OpenSim")," Universidad Nacional de Colombia, Bogotá, 2015

[12] J. Perry, "Normal and pathologic gait," 2016. [En línea] Disponible: https://musculoskeletalkey.com/normal-and-pathologic-gait/

[13] P. Roriz y A. B. Lobo Ribeiro "Fiber optical sensors in biomechanics" en *Opto-Mechanical Fiber Optic Sensors*, H. Alemohammad, Eds. Fondo Editorial CIB, 2018, pp. 263–300.

[14] National Center for Simulation in Rehabilitation Research (NCSRR), "OpenSim, See the work," [En línea] Disponible: https://opensim.stanford.edu/work/index.html

[15] A. Nandy, S. Chakraborty, J. Chakraborty y G. Venture, "Validation study of low-cost sensors." en *Modern Methods for Affordable Clinical Gait Analysis: Theories and Applications in Healthcare Systems.*" G. Capille, Eds. Academic Press, 2021, pp.45-55.

[16] J. Hicks, "How Inverse Kinematics Works." 2018. Disponible en: https://shorturl.at/L1ucB

[17] T. K. Uchida y S. L. Delp, "1: First steps" en "*Biomechanics of Movement : The Science of Sports, Robotics, and Rehabilitation."* Eds. The MIT Press, 2021, pp. 1-22.

[18] J. Hicks, "Getting Started with Inverse Dynamics." 2018. Disponible en: https://shorturl.at/scPzd

[19] J. Hicks, "How RRA Works." 2018. Disponible en: https://shorturl.at/1tUvx

[20] J. Hicks, "Getting Started with Static Optimization." 2018. Disponible en: https://shorturl.at/qosby

[21] J. Hicks, "Working with Static Optimization." 2018. Disponible en: https://shorturl.at/m08dR

[22] S. Heitnz y E. M. Gutierrez Farewik, "Static optimization of muscle forces during gait in comparison to EMG-to-force processing approach," *Gait & Posture*, vol. 26, no. 2, pp. 279-288, 2007, doi: https://doi.org/10.1016/j.gaitpost.2006.09.074

[23] S. Heitnz, *Muscular Forces from Static Optimization* [Reporte técnico]. Estocolmo (Suecia): Royal Institute of Technology, 2006.

[24] J. Hicks, "Getting Started with CMC" 2018. Disponible en: https://acortar.link/LkOGXJ

[25] D. G. Thelen y F. c. Anderson, "Using computed muscle control to generate forward dynamic simulations of human walking from experimental data" *Journal of Biomechanics*, vol. 36, no. 6, pp.1107-1115, 2006, doi: https://doi.org/10.1016/j.jbiomech.2005.02.010

[26] A. M. Cárdenas Torres, *Design and Validation of a Polyaxial Knee Mechanical Prosthesis Alignment Protocol, Based on a Multivariate Biomechanical Model* [tesis de doctorado]. Medellín (Colombia): Universidad de Antioquia, 2022.

[27] J. Hicks, “Scaling.” 2018.Disponible en: https://acortar.link/NN9yhx

[28] C3D org, “The 3D Standar” [En línea]. Disponible en: https://www.c3d.org/

[29] Mokka, “Motion Kinematic and kinetic analyzer.” [En línea]. Disponible en: https://biomechanical-toolkit.github.io/mokka/

[30] S. Bagheri, “C3D Tools” [En línea]. Disponible en: https://c3dtools.com/

[31] J. Hicks, “_Inverse Kinematics Best Practices”, 2012. Disponible en: https://acortar.link/IEFc34

[32] F. Vaverka, M. Elfmark, Z. Svoboda y M. Janura, “System of gait analysis based on ground reaction force assessment.” *Acta Gymnica,* vol. 45, no. 4, 2015, [En línea]. https://shorturl.at/a10Bs

[33] Vald Performance, “Uniaxial vs. Triaxial Force Plates: How many axes do I need?.” 2023 [En línea]. Disponible en: https://bit.ly/3mfvktD.

[34] A. Willson, *A Quasi-Passive Biarticular Prosthesis and Novel Musculoskeletal Model for Transtibial Amputees* [tesis de maestría]. Washington (Estados Unidos): University of Washington, 2017.

ANEXOS

Guía metodológica
Simulación de marcha y amputación transfemoral con OpenSim

María Paulina Tobón Palacio

Índice

Introducción **4**
Softwares y recursos digitales **4**
1. Registro de Marcha **4**
1.1. Número de cámaras 5
1.2. Archivos 5
1.3. Fotos y videos 5
1.4. Marcadores 5
1.5. Registros 7
1.6. Gestión de información 8
2. Preparación de archivos **8**
2.1. Conversión de archivos 9
2.2. Revisar y depurar los archivos 12
2.2.1. Archivos de movimiento y plataformas .mot 12
2.2.2. Archivos de trayectoria de marcadores .trc 12
2.3. Edición de marcadores 12
2.4. Recortar tiempo del registro 14
2.5. Exportar el archivo de trayectorias .trc 15
2.6. Edición del set de marcadores de un modelo de OpenSim 15
2.6.1. Eliminar marcadores 17
2.6.2. Agregar un nuevo marcador 18
2.6.3. Editar un marcador 19
2.6.4. Exportar el set de marcadores 20
3. Escalado **22**
3.1. Archivo de configuración 22
3.1.1. Edición de información general 24
3.1.2. ModelScaler 24
3.1.3. MarkerPlace 30
3.1.4. Edición de información final 31
3.2. Ejecutar la herramienta de escalado 31
3.3. Evaluar el resultado del escalado 33
4. Cinemática inversa **35**
4.1. Archivo de configuración de cinemática inversa 38
4.1.1. Creación y edición del archivo de cinemática inversa 38
4.2. Ejecutar la herramienta de cinemática inversa 40
4.3. Evaluar el resultado de cinemática inversa 42
5. Dinámica inversa **43**
5.1. Archivo de configuración dinámica inversa 44

5.1.1. Archivo de fuerzas externas 46
5.2. Ejecutar la herramiento de dinámica inversa 49
5.3. Evaluar el resultado de dinámica inversa 50
6. Algoritmo de reducción residual 54
6.1. Archivo de configuración de RRA 55
6.1.1. RRA Tasks 58
6.1.2. RRA Actuators 60
6.2. Ejecutar la herramienta de RRA 61
6.3. Evaluar el resultado de RRA 62
7. Optimización estática 64
7.1. Opción 1 64
7.1.1. Ejecutar la herramienta de Optimización estática 65
7.2. Opción 2 67
7.2.1. Ejecutar la herramienta de Optimización estática 67
7.3. Evaluar el resultado de optimización estática 67
8. Control muscular computacional 68
8.1.1. CMC ControlConstraints 72
8.1.2. CMC Tasks 74
8.1.3. CMC Actuators 76
8.2. Ejecutar la herramienta de CMC 78
8.3. Evaluar el resultado de CMC 79
8.4. Corregir CMC 80
9. Simulación de amputación transfemoral 84
9.1. Eliminación de componentes 84
9.1.1. Componentes óseos 86
9.1.2. Componentes musculares 88
9.1.3. Marcadores 95
9.2. Modificación femoral 96
9.2.1. ParaView 96
9.2.2. MeshLab 100
9.2.3. Software CAD 101
9.2.4. Reubicación en OpenSim 103
9.3. Modificación muscular 105
9.4. Carga de geometría protésica 108
9.4.1. Ubicación espacial de componentes 108
9.4.2. Creación de cuerpos 109
9.4.3. Creación de articulaciones 111
Referencias 121

Introducción

Esta guía metodológica presenta las consideraciones esenciales para la simulación de marcha en OpenSim, abarcando desde las condiciones previas a la ejecución de la simulación hasta el uso detallado de las herramientas del software. Se incluyen aspectos clave como la adquisición y preparación de los datos, la selección y modificación del modelo en OpenSim, así como el empleo de herramientas fundamentales, desde el escalado hasta la dinámica inversa. Además, se explica la creación y modificación de los archivos de configuración necesarios y los criterios para la validación de los resultados obtenidos.

Más allá de proporcionar instrucciones sobre cómo analizar archivos de captura de movimiento en OpenSim, la guía también detalla el procedimiento para modificar un modelo y simular una amputación transfemoral. Asimismo, se presentan herramientas complementarias que facilitan el procesamiento y la visualización de los datos, optimizando el flujo de trabajo en conjunto con OpenSim.

Si bien este trabajo se enfoca en la simulación de la marcha protésica de un sujeto con amputación transfemoral, las metodologías descritas pueden adaptarse a otros análisis biomecánicos, permitiendo su aplicación en estudios con objetivos distintos. El propósito principal de este documento es establecer una metodología clara y estructurada que facilite el uso de OpenSim, reduciendo el tiempo de búsqueda y aprendizaje necesario para comprender y ejecutar cada una de sus herramientas.

Softwares y recursos digitales

Para completar los pasos de esta guía, es necesario contar con diversas herramientas digitales y programas, cuya descarga se recomienda realizar previamente. El uso de cada uno de estos recursos se explicará en las secciones correspondientes, según sea necesario. A continuación, se presenta la lista de recursos requeridos:

- OpenSim
- C3D Tools
- MOKKA
- Editor de texto (Notepad++, Visual Studio Code u otro)
- ParaView
- MeshLab
- Software CAD (Fusion 360, SolidWorks u otro)

1. Registro de Marcha

Lo primero que se debe responder cuando se desea realizar un estudio en OpenSim a partir de registros de marcha de pacientes es qué se desea analizar y cuál es el punto de partida, es decir, se debe delimitar el objeto de interés, bien sea, estudiar patrones de marcha, evaluar diseños, realizar simulaciones o modelos predictivos, adquirir información, entre otros. Para esto es necesario revisar un poco los modelos básicos que ofrece OpenSim, así como algunos que se pueden encontrar en SimTK, el cual es un repositorio gratuito que contiene diferentes modelos biomecánicos [1]. Cuando se revisan estos modelos hay que tener en cuenta ciertas consideraciones como, número y ubicación de marcadores, tipo de modelo muscular, segmentos y cuerpos del modelo; esto, con el fin de realizar los registros basados no sólo en lo que se desea estudiar sino también en la información disponible y necesaria para tomar como base un modelo ya establecido.

1.1. Número de cámaras

El laboratorio de captura de movimiento donde se llevarán a cabo los registros debe de contar como mínimo con un conjunto de 6 cámaras, ubicadas adecuadamente para poder reproducir un objeto 3D [2].

1.2. Archivos

Es necesario conocer cómo se exportan los archivos del laboratorio; es decir, el formato (.c3d, .tdf), la calidad y la información que contienen.

1.3. Fotos y videos

Es de gran utilidad tomar fotos y videos de los sujetos desde diferentes vistas, como frontal y sagital, tanto antes como después de ubicar los marcadores experimentales, con el fin de facilitar la reconstrucción del modelo en OpenSim [3].

1.4. Marcadores

Partiendo del enfoque de esta guía, algunos modelos útiles de OpenSim son Gait2354, Gait2392 y Rajagopal, los cuales tienen archivos con conjuntos de marcadores virtuales posicionados adecuadamente. Una vez haya definido su objeto de estudio y el modelo de OpenSim en el que se va a basar, es necesario establecer cuántos y dónde se van a ubicar los marcadores experimentales. Para esto se debe tener en cuenta que debe tener suficientes marcadores como para escalar al sujeto, es decir, al menos tres por segmento corporal. Además, se deben ubicar no solo en lugares donde el movimiento tanto muscular como de piel sea mínimo, sino que correspondan con puntos anatómicos

fácilmente identificables en otros sujetos, para ello se recomienda hacer una revisión bibliográfica, usar los marcadores ya establecidos por el modelo de OpenSim seleccionado (En la sección 2.3 se explica cómo realizar ajustes al set de marcadores) o revisar la **TABLA I** donde se muestra un conjunto de 50 marcadores para una simulación de cuerpo completo, de la cual puede extraer su conjunto de marcadores según su aplicación, considerando las posiciones allí descritas.

TABLA I

Conjunto de marcadores para simulación de cuerpo completo [3].

Segmento Corporal	Marcadores
Right Upper Arm	R.Bicep.Front, R.Bicep.Upper, R.Bicep.Lower, R.Elbow.Lateral, R.Elbow.Medial
Right Lower Arm	R.Forearm, R.Wrist.Lateral, R.Wrist.Medial
Left Upper Arm	L.Bicep.Front, L.Bicep.Upper, L.Bicep.Lower, L.Elbow.Lateral, L.Elbow.Medial
Left Lower Arm	L.Forearm, L.Wrist.Lateral, L.Wrist.Medial
Torso	R.Acromion, L.Acromion, Clavicle, C7
Pelvis	R.ASIS, L.ASIS, R.PSIS, L.PSIS
Right Thigh	R.Thigh.Front, R.Thigh.Upper, R.Thigh.Lower
Left Thigh	L.Thigh.Front, L.Thigh.Upper, L.Thigh.Lower
Right Shank	R.Shank.Front, R.Shank.Upper, R.Shank.Lower
Left Shank	L.Shank.Front, L.Shank.Upper, L.Shank.Lower
Right Foot	R.Heel, R.MTP1 (big toe), R.MTP5 (little toe)
Left Foot	L.Heel, L.MTP1 (big toe), L.MTP5 (little toe)
Static Trial Only	
Right Knee	R.Knee.Lateral, R.Knee.Medial
Left Knee	L.Knee.Lateral, L.Knee.Medial
Right Ankle	R.Ankle.Lateral, R.Ankle.Medial
Left Ankle	L.Ankle.Lateral, L.Ankle.Medial

Los modelos suelen tener más marcadores de los empleados durante los registros de movimiento, dado que OpenSim sugiere usar un conjunto de marcadores mayor para aumentar la precisión del escalado de los modelos. Por ejemplo, para escalar los modelos Gait2354 y Gait2392, solo es necesario usar 12 marcadores, sin embargo, cada modelo cuenta con un conjunto de 39 marcadores (**Fig. 1**), usados para describir el movimiento

del sujeto, por lo cual, al momento de escoger su conjunto de marcadores, debe considerar tanto los que usará para el escalado como los que requiere para rastrear el movimiento del paciente.

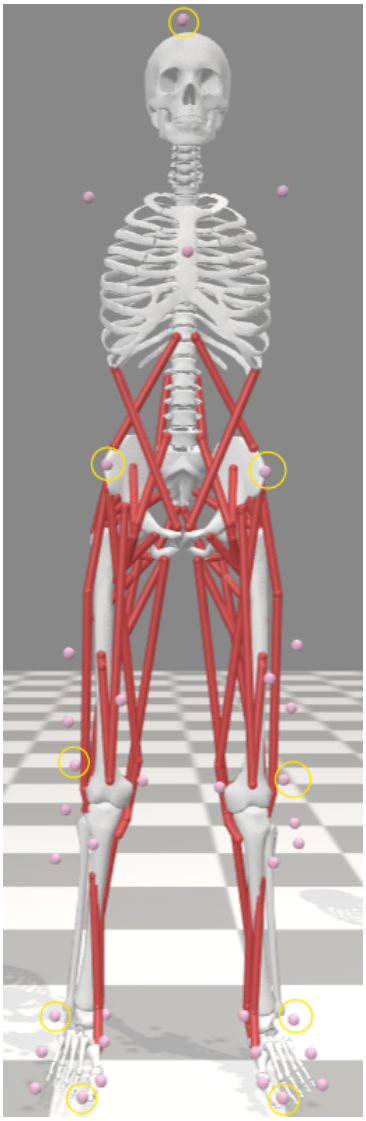


Fig. 1 Marcadores del modelo Gait2354.

Nota. En amarillo se presentan 9 de los 12 marcadores usados para el escalado, dado que los marcadores de los talones y el sacro están en la zona posterior.

En caso de que esté realizando el registro de movimiento usted mismo y ya haya definidos su conjunto de marcadores, proceda a etiquetarlos en el software de registro de marcha que use su laboratorio, tal cual se sugiere en su modelo base de OpenSim, esto facilitará el trabajo, posteriormente coloque los marcadores en el sujeto de prueba teniendo en cuenta los sitios anatómicos, tome fotos y videos de las ubicaciones y guarde el nombre de los

marcadores adecuadamente en el archivo del laboratorio, para proceder con los registros del sujeto.

1.5. Registros

Proceda a grabar tanto una prueba estática, con el sujeto en posición anatómica, **Fig. 2,** así como un mínimo de tres registros de marcha de cada sujeto evaluado. Mediante la prueba estática es posible realizar el escalado del sujeto, dado que proporciona la ubicación de todos los marcadores experimentales sin movimiento de estos, por tanto el software es capaz de efectuar mediciones entre pares de marcadores y calcular un factor de escala para cada segmento corporal. Por otro lado, con los registros de marcha se obtienen las trayectorias de los marcadores, lo que permite recrear el movimiento del sujeto en OpenSim y obtener información a partir de este, así como realizar estudios de cinemática y dinámica inversa. Las pruebas de marcha se deben realizar a una velocidad cómoda para el sujeto y deben incluir por lo menos 3 ciclos de marcha, en caso de contar con plataformas de fuerza, es necesario garantizar que el sujeto sólo la pise con el pie de interés y no con ambos.



Fig. 2 Posición anatómica, vistas frontal y posterior

Nota. Fuente Freepik.es

1.6. Gestión de información

Ser ordenado con la información es clave, nombrar los marcadores adecuadamente, guardar la identificación del usuario (nombre o código), recopilar información relevante como duración del registro o número de frames y unificar esta con la información descrita en el numeral anterior. Con el fin de reconocer con qué se está trabajando considere crear una carpeta para cada sujeto de estudio la cual contenga fotos y videos del mismo, así como los resultados del registro.

2. Preparación de archivos

Para ejecutar simulaciones en OpenSim es necesario contar con una serie de archivos que se emplearán durante el uso de las diferentes herramientas del software, como se muestra en la **Fig. 3**, por lo cual se debe realizar la preparación de estos.

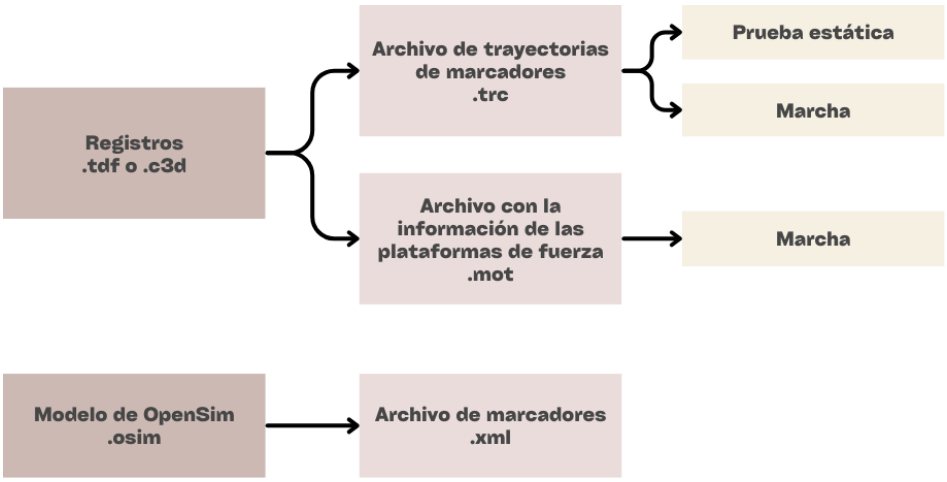


Fig. 3 Archivos iniciales para simular en OpenSim

2.1. Conversión de archivos

OpenSim aún no permite trabajar directamente con archivos .c3d, ni .tdf; sino que emplea otro tipo de formatos, para cargar archivos externos que contengan los datos de plataformas de fuerzas o movimiento emplea la extensión .mot, mientras que para utilizar la información de las trayectorias de los marcadores experimentales se requiere un .trc.

Cuando se parte desde un archivo .c3d, que es el formato más común, se recomienda usar la herramienta C3D Tools, este un servicio digital y gratuito que permite no solo visualizar archivos en este formato sino que también brinda la posibilidad de extraer los archivos .trc y .mot requeridos por OpenSim, para usarlo es necesario ingresar a la página web y crear una cuenta, una vez se haya registrado o iniciado sesión, debe navegar al apartado de *Export to OpenSim*, el cual se encuentra en el panel izquierdo, como se ve en la **Fig. 4,** allí encontrará la opción para cargar un archivo en formato .c3d, como se muestra en la **Fig. 5,** cuando el sistema termine de procesar la información, podrá desplazarse a la parte inferior para encontrar botones que le permiten descargar los nuevos archivos .trc y .mot, **Fig. 6.** Cabe mencionar que el archivo .mot sólo se requiere del registro de marcha, dado que es en este en el que se usan plataformas de fuerza.

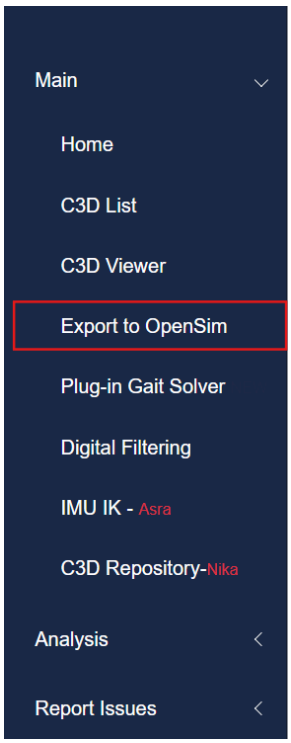


Fig. 4 Menú de C3D Tools

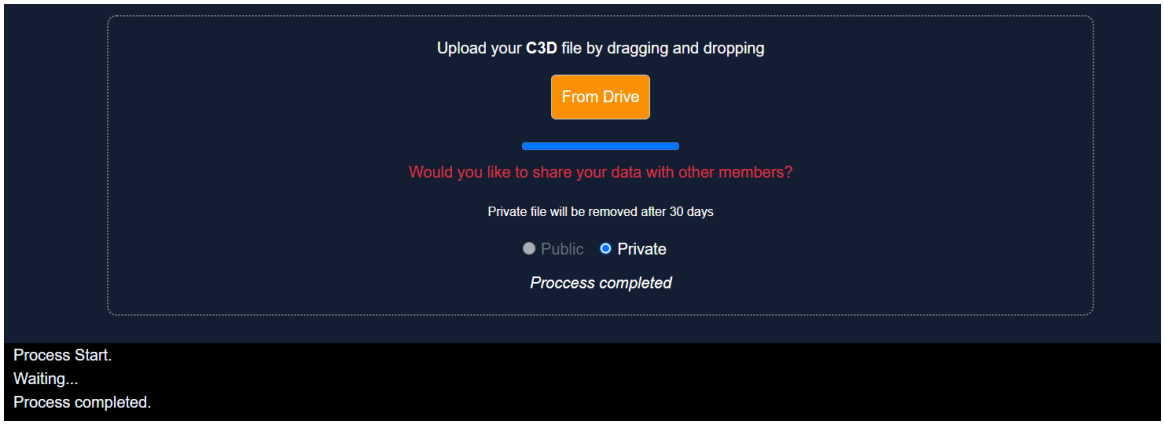


Fig. 5

Donde cargar el archivo en C3D Tools

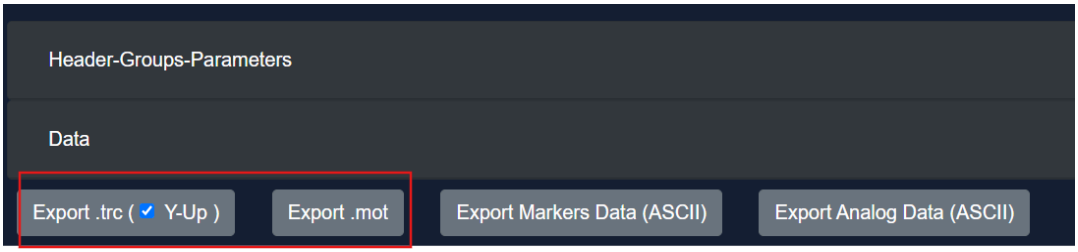


Fig. 6 Donde descargar los archivos en C3D Tools

Por otro lado, si se parte desde un archivo de tipo .tdf, que son los formatos originales de los laboratorios BTS, es posible leer el contenido de este a partir de una aplicación

llamada MOKKA, este software libre permite analizar diferentes tipos de archivos que contienen datos biomecánicos en 3D y 2D, como trayectorias de marcadores, señales de electromiografía (EMG), plataformas de fuerza, entre otros. Con este software podrá visualizar y editar los registros, revisar la nomenclatura de los marcadores y editarla, así como ver el movimiento registrado y recortar la duración de este para mantener solo la fracción de interés [4].

Partiendo de que tiene instalado MOKKA comience por abrir el archivo ejecutable (*Mokka.exe*) que contiene el software de MOKKA y presione *File → Open…* Como se muestra en la **Fig. 7,** esto abrirá el explorador de archivos de su dispositivo, donde podrá buscar los registros del sujeto y abrirlos.

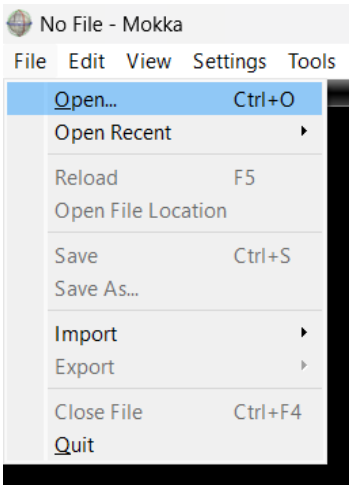


Fig. 7 Abrir un archivo en MOKKA

Para obtener el archivo .c3d desde MOKKA, a partir de los registros de marcha y de la prueba estática, basta con seguir la siguiente ruta: *File → Export → C3D file…*, como se muestra en la **Fig. 8**. Sin embargo, se recomienda tener precaución en este paso y exportar directamente los registros de movimiento en formato .c3d desde el laboratorio de captura de movimiento, dado que la conversión de formatos puede dañar los archivos disminuyendo la capacidad de C3D Tools para extraer los archivos .mot y .trc desde el .c3d.

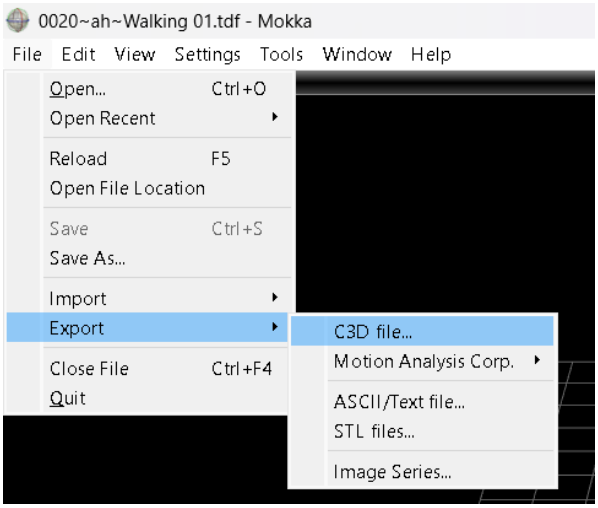


Fig. 8 Exportar archivos de MOKKA

2.2. Revisar y depurar los archivos

Una vez tenga los archivos del registro tanto de marcha, como de la prueba estática en los formatos aceptados por OpenSim, es necesario revisar la calidad de los mismos.

2.2.1. Archivos de movimiento y plataformas .mot

Empleando un editor de texto, como Visual Studio Code es posible revisar la estructura de estos archivos, la cual debería ser similar a la presentada en la **Fig. 9**, donde se observa que contiene un encabezado con información general del documento, seguido de los indicadores de magnitud y ubicación del vector de fuerza de reacción del piso y del torque producido por este, como se marca en rojo, además del tiempo de registro.

walkingC3Dtools.mot

```
0026~aa~Walking 01_79a.mot
version = 1
nRows= 729
nColumns= 10
endheader
time    ground_force_vx ground_force_vy ground_force_vz ground_force_px ground_force_py ground_force_pz ground_torque_x ground_torque_y ground_torque_z
5.950000    11.717051   134.596025  -5.856692   0.611835    -0.000439   0.069975    18.941176   0.629569    42.351511
5.966667    -45.783277  400.150244  -11.423564  0.558513    -0.000287   0.086835    49.541485   12.620631   104.521199
5.983333    -83.795151  549.821024  -26.429328  0.574004    -0.000322   0.085031    69.061983   21.749502   152.129164
6.000000    -80.628392  595.754635  -21.547274  0.576480    -0.000327   0.084992    74.855107   18.519058   166.313255
```

Fig. 9 Archivo de movimiento y plataformas de fuerza (.mot)

Nota. En rojo se marcan los identificadores de la plataforma de fuerza y el tiempo del registro

2.2.2. Archivos de trayectoria de marcadores .trc

La estructura de este archivo es similar a la que se presenta en la **Fig. 10,** contiene un encabezado con información general del registro y posteriormente los nombres de los

marcadores, los cuales se pueden editar desde MOKKA o usando un editor de código como Visual Studio Code. Debajo de los nombres se encuentran las coordenadas en XYZ de cada marcador.

walking0026_mokka.trc

```
1   PathFileType    4   (X/Y/Z) C:/Users/Pauli/Desktop/Paulina/UdeA/Trabajo_de_grado/Subject0026_OpenSim/walking0026_mokka.trc
2   DataRate    CameraRate  NumFrames   NumMarkers  Units   OrigDataRate    OrigDataStartFrame  OrigNumFrames
3   60.00   60.00   287 71  m   60.00   347 287
4   Frame#  Time    V.Sacral            R.ASIS          R.Thigh.Upper           R.Thigh.Rear            R.Knee.Lat          R.Shank.Upper
5           X1  Y1  Z1  X2  Y2  Z2  X3  Y3  Z3  X4  Y4  Z4  X5  Y5  Z5  X6  Y6  Z6  X7  Y7  Z7  X8  Y8  Z8  X9  Y9  Z9  X10 Y10 Z10 X11 Y11 Z11 X1
6
7   1   0.000   1.17389 0.95030 0.13658 1.00071 0.94365 0.00042 1.08705 0.91780 -0.05944    1.01127 0.69369 -0.09515    0.94771 0.50527 -0.09725
8   2   0.017   1.16170 0.94855 0.13356 0.98767 0.94119 -0.00075    1.07452 0.91533 -0.06104    1.00373 0.68888 -0.09501    0.94133 0.50000 -0.09
9   3   0.033   1.15058 0.94766 0.13039 0.97528 0.93829 -0.00166    1.06108 0.91235 -0.06297    0.99769 0.68376 -0.09510    0.93452 0.49471 -0.09
10  4   0.050   1.14065 0.94759 0.12677 0.96062 0.93529 -0.00319    1.04754 0.90915 -0.06490    0.99151 0.67847 -0.09499    0.92696 0.49011 -0.08
11  5   0.067   1.12684 0.94568 0.12283 0.94588 0.93168 -0.00494    1.03296 0.90516 -0.06707    0.98391 0.67360 -0.09511    0.91819 0.48605 -0.08
12  6   0.083   1.11298 0.94372 0.11890 0.93165 0.92765 -0.00654    1.01905 0.90122 -0.06926    0.97291 0.66935 -0.09542    0.90712 0.48200 -0.08
```

Fig.10 Archivo de registro de marcha (.trc)

Nota. En rojo se señalan los marcadores experimentales registrados durante la prueba

2.3. Edición de marcadores

Una ventana similar a la que se muestra en la **Fig. 11** es lo que usted verá cuando abra un archivo en MOKKA, donde cada punto es un marcador experimental, cuyo nombre aparece en el lado derecho de la pantalla. Puede reproducir el registro desde el panel inferior izquierdo para ver el movimiento de los marcadores.

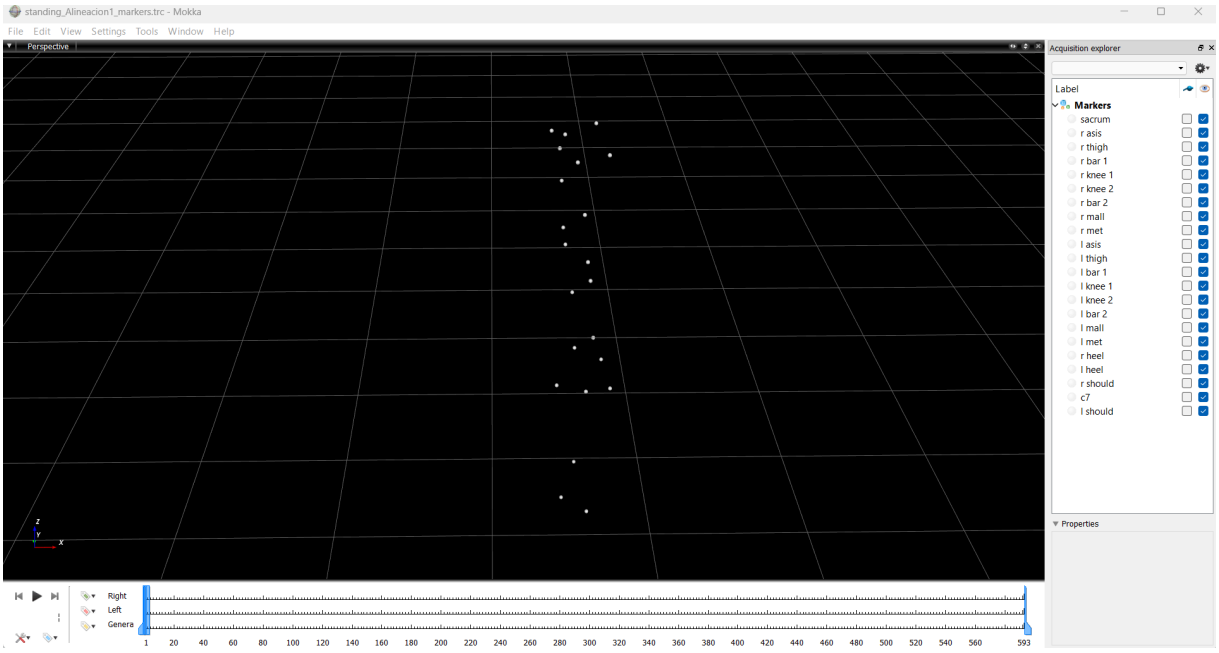


Fig. 11 Pantalla principal de MOKKA

Para renombrar un marcador basta con seleccionarlo y escribir el nuevo nombre en la pestaña de *Properties* como se muestra en la **Fig. 12**

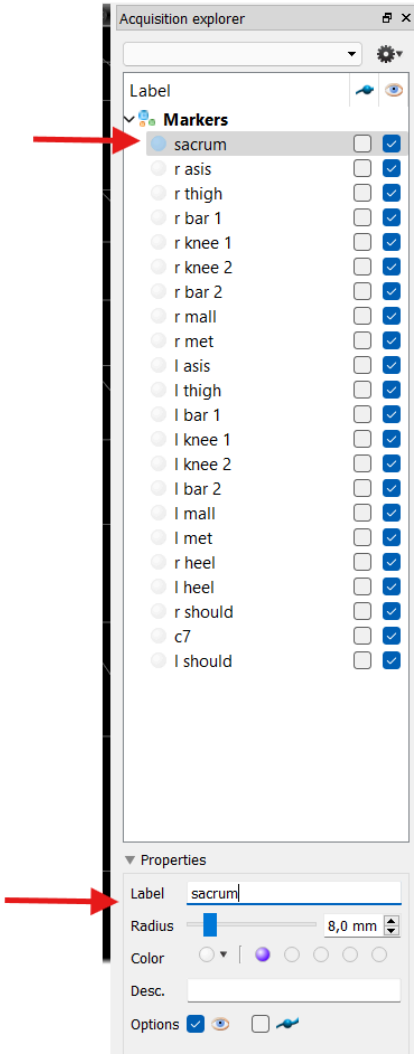


Fig. 12 Lista de marcadores en MOKKA.

Nota. En rojo se señala un marcador seleccionado para editar

2.4. Recortar tiempo del registro

A veces, los registros contienen espacios de tiempo no son útiles para la simulación, bien sea por tiempos sin movimiento, ruido, recorrido fuera de las plataformas de fuerza o demás, por lo que es útil delimitar la porción exacta del video que se va a usar. Para recortar el segmento de interés, use las barras laterales que están en el panel inferior y delimite con ellas la zona que desea mantener, como se muestra en la **Fig. 13**, una vez tenga la duración deseada presiona en la barra superior *Tools* → *Acquisition* → *Crop Frames of Interest* como se muestra en la **Fig. 14.**

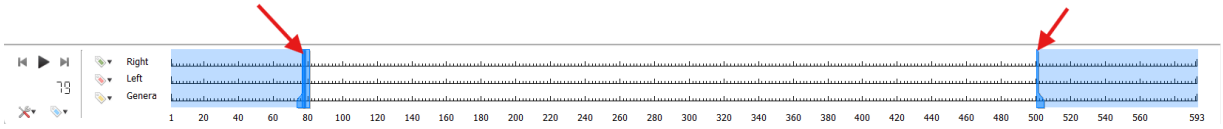


Fig. 13 Vista de la sección para recortar la duración de un registro en MOKKA

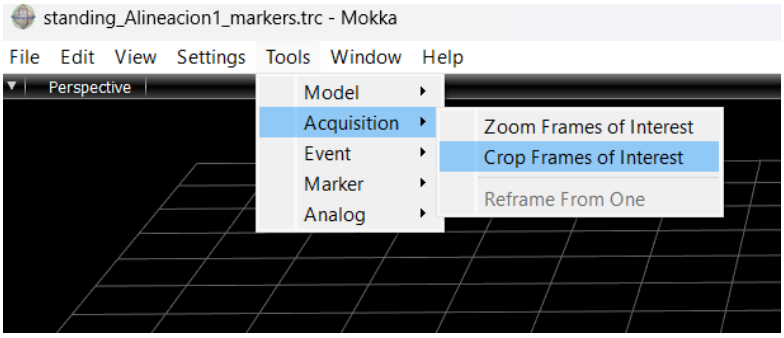


Fig. 14 Herramienta para recortar un registro con MOKKA

MOKKA tiene más funciones que puede explorar; sin embargo, no son objeto de estudio de esta guía.

2.5. Exportar el archivo de trayectorias .trc

Una vez terminado de editar el archivo de trayectoria, nombre de los marcadores o la duración del registro, basta con seguir la siguiente ruta: *File → Export → Motion Analysis Corp. → .TRC file…*, como se muestra en la **Fig. 15,** así podrá obtener el archivo que será usado en las simulaciones de OpenSim.

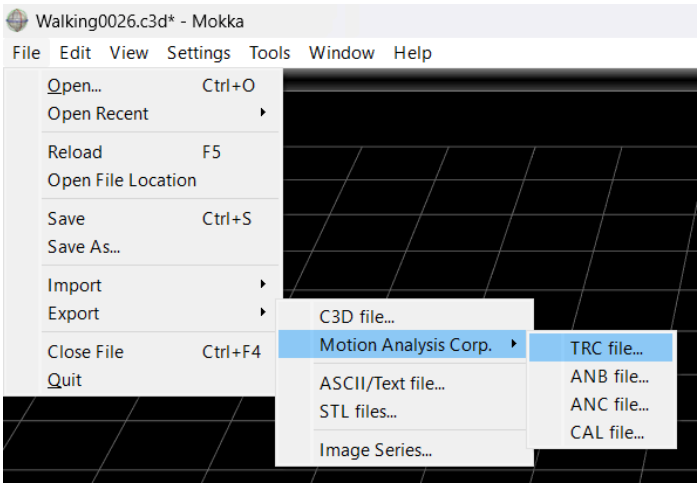
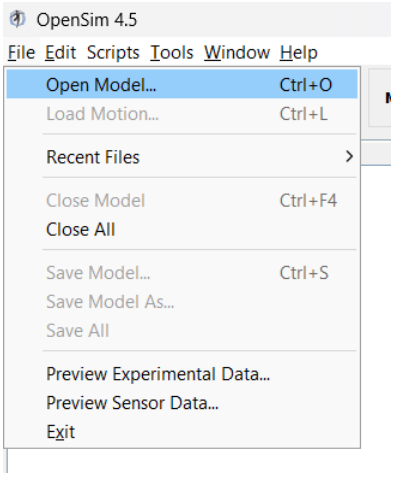


Fig. 15 Como exportar archivos desde MOKKA

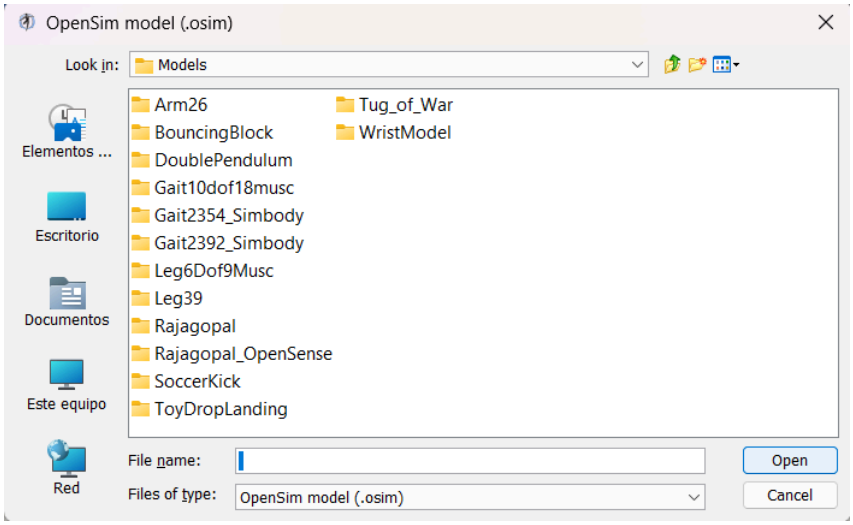
2.6. Edición del set de marcadores de un modelo de OpenSim

De la sección 1 se tiene claro el modelo que se va a usar como base, sin embargo, a veces es necesario que el conjunto de marcadores que contiene el modelo sea modificado en términos de eliminar, editar o agregar un nuevo marcador, con el fin de que concuerde con el set de marcadores seleccionado para su estudio, por lo cual deberá comenzar por realizar una copia del modelo escogido.

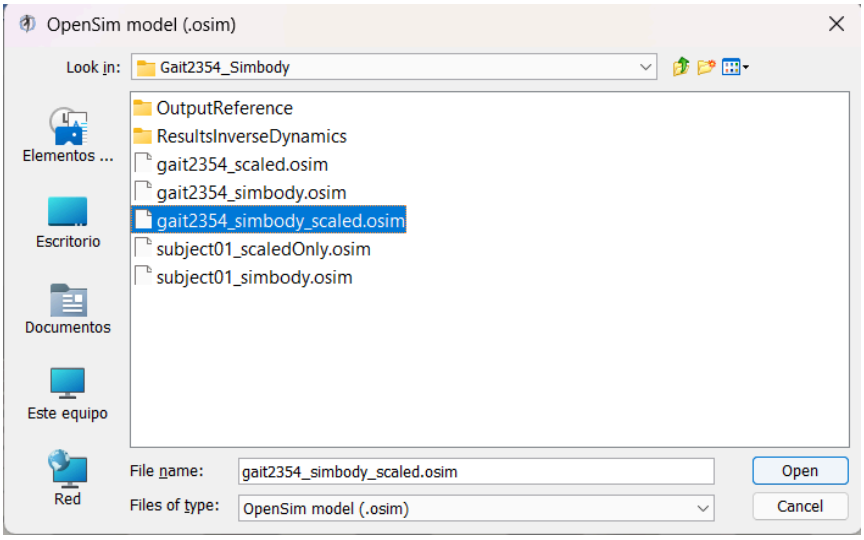
Abra OpenSim y presione *File→Open..*, esto abrirá el explorador de archivos de su dispositivo, busque la carpeta de modelos de OpenSim, generalmente está en la ruta *C:\Users\xxx\Documents\OpenSim\4.5\Models,* una vez allí del clic en la carpeta del modelo y abra el archivo *.osim* como se muestra en **Fig 16.**



a



b



c

Fig. 16 Como abrir un modelo de OpenSim

Posteriormente presione *File→Save As* y renombre el archivo (**Fig. 17**) , esto le permitirá escoger una nueva ubicación para guardar su modelo base y así evitará modificar el modelo que trae OpenSim en caso de necesitarlo en futuras simulaciones.

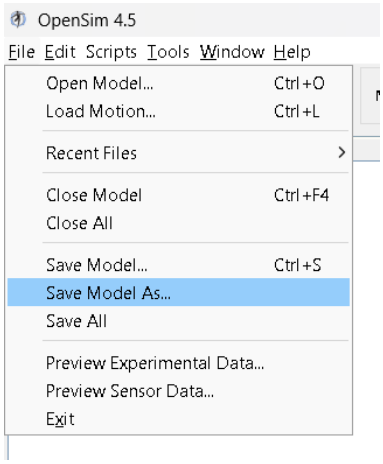


Fig. 17 Como guardar un modelo desde OpenSim

2.6.1. Eliminar marcadores

Para eliminar un marcador solo debe desplegar la lista de marcadores del modelo, seleccionar el marcador que desea eliminar, dar clic derecho y presionar *delete*, como se muestra en la **Fig. 18**

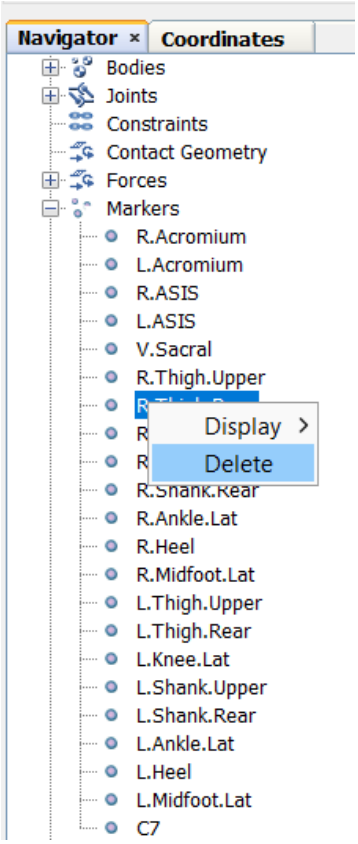


Fig. 18 Como eliminar un marcador desde OpenSim

2.6.2. Agregar un nuevo marcador

Este paso es necesario en caso de que el conjunto de marcadores definido para sus ensayos cuente con marcadores en posiciones diferentes a las ofrecidas por el conjunto de marcadores del modelo de OpenSim, para ello es necesario dar clic derecho sobre el menú *Markers* y presionar *Add New*, como se muestra en la **Fig. 19**

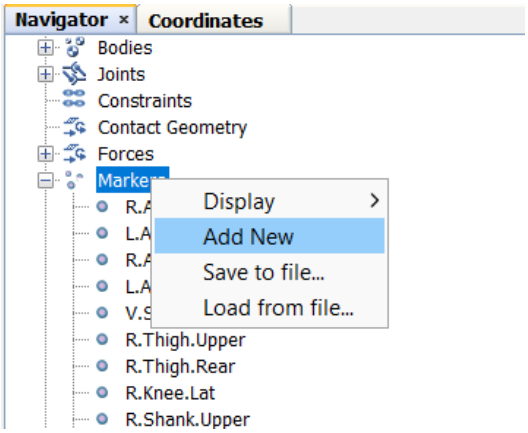


Fig. 19 Como agregar un marcador desde OpenSim

Una vez se crea el nuevo marcador puede moverlo a la ubicación deseada sobre el modelo dando doble clic sobre el mismo (Lo encontrará al final de la lista de marcadores como *NewMarker_*), puede usar los controles en XYZ para posicionarlo sobre el modelo, cuando esté satisfecho con la ubicación del mismo, debe asignarle un *PhysicalFrame: parent_frame,* es decir, un marco de referencia al que estará condicionado y el cual limitará su movimiento durante la ejecución de actividades como el escalado del modelo. Para ejecutar el proceso anterior, revise la ventana de *Properties* ubicada debajo de *Navigator*, allí verá las coordenadas del marcador, el nombre y el *PhysicalFrame: parent_frame* justo debajo de la opción *Socket*, solo debe seleccionar del menú desplegable la opción adecuada según la ubicación del marcador, bien sea torso, pelvis, fémur, ground, entre otros, **Fig. 20.**

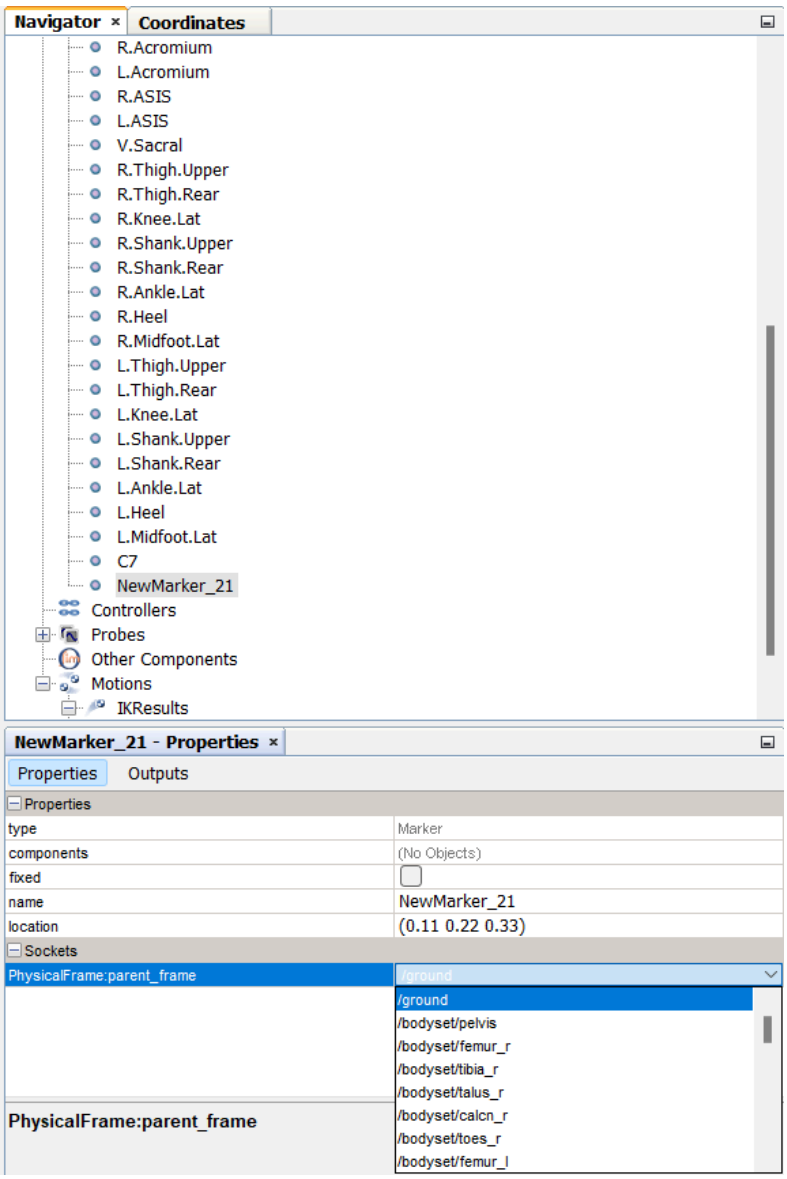


Fig. 20 Como seleccionar el *PhysicalFrame: parent_frame* de un marcador desde OpenSim

2.6.3. Editar un marcador

Para editar la posición, el nombre o el *PhysicalFrame: parent_frame* de un marcador debe dar doble clic sobre el mismo, ya sea en el modelo o desde la lista de marcadores presentada en la ventana de *Navigator* y editar las características en la ventana de *Properties,* al finalizar presione *Enter* para ver los cambios reflejados en el modelo, **Fig. 21.**

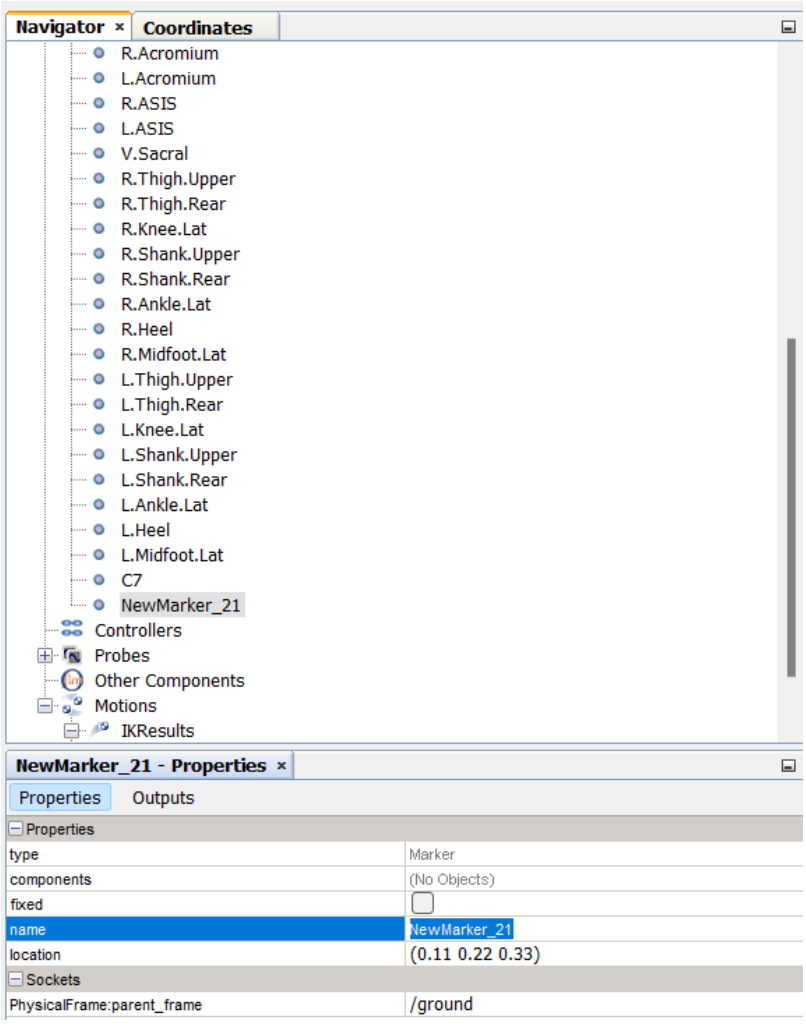


Fig. 21 Como editar un marcador desde OpenSim

2.6.4. Exportar el set de marcadores

Una vez su modelo tenga todos los marcadores en las posiciones deseadas y no pretenda realizar más modificaciones, puede exportar el archivo para usarlo en los próximos pasos, para ello solo debe dar clic derecho sobre la opción *Markers* y luego presionar *Save File*, recuerde nombrar los archivos de forma clara y con etiquetas que le permitan recordar su contenido, **Fig. 22.**

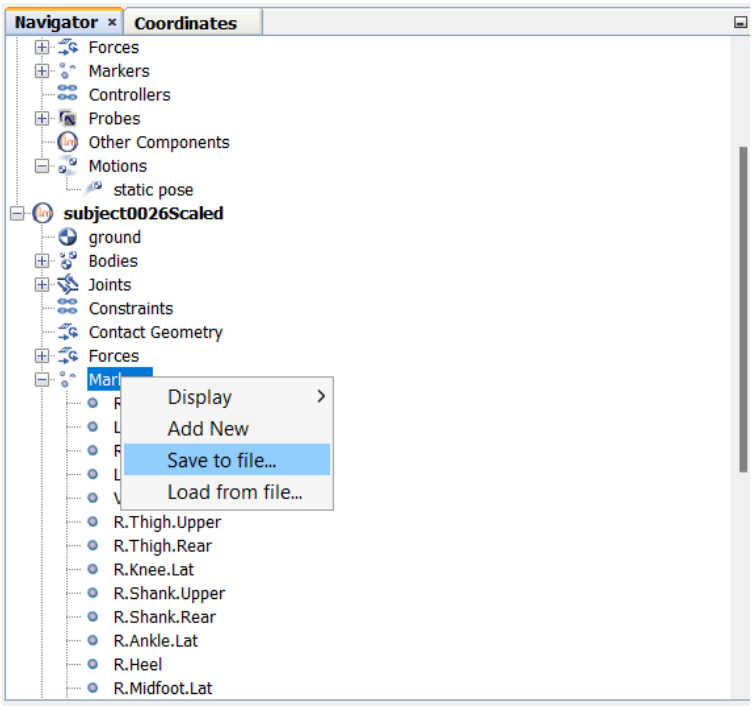


Fig. 22 Como exportar el archivo de marcadores de OpenSim

El archivo de marcadores exportado tiene una estructura similar al presentado en la **Fig. 23,** donde en cada sección *Marker* se encuentra el nombre del marcador, el *PhysicalFrame: parent_frame* al que pertenece, la ubicación espacial del mismo y si está fijado o no.

Subject0026_Modelmarkers.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<OpenSimDocument Version="40500">
    <MarkerSet name="markerset">
        <objects>
            <Marker name="R.Acromium">
                <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame (description: The
                <socket_parent_frame>/bodyset/torso</socket_parent_frame>
                <!--The fixed location of the station expressed in its parent frame.-->
                <location>-0.0049331020225664712 0.4265378353261482 0.17070511058924021</location>
                <!--Flag (true or false) specifying whether the marker is fixed in its parent frame during the marker pl
                <fixed>false</fixed>
            </Marker>
            <Marker name="L.Acromium">…
            </Marker>
            <Marker name="R.ASIS">…
            </Marker>
            <Marker name="L.ASIS">…
            </Marker>
            <Marker name="V.Sacral">…
            </Marker>
            <Marker name="R.Thigh.Upper">…
            </Marker>
            <Marker name="R.Thigh.Rear">…
            </Marker>
            <Marker name="R.Knee.Lat">…
            </Marker>
```

Fig. 23 Contenido del archivo de marcadores de un modelo

Una vez completados los pasos anteriores, se contará con los archivos suficientes para comenzar a realizar los procesos de escalado, cinemática y dinámica inversas.

Un punto muy importante a tener en cuenta es que OpenSim no es un software que guarda automáticamente los archivos ni las modificaciones, por lo cual se recomienda guardar periódicamente los cambios realizados para no perder el trabajo, para esto basta con ir a la barra superior y presionar *File* → *Save All,* **Fig. 24.**

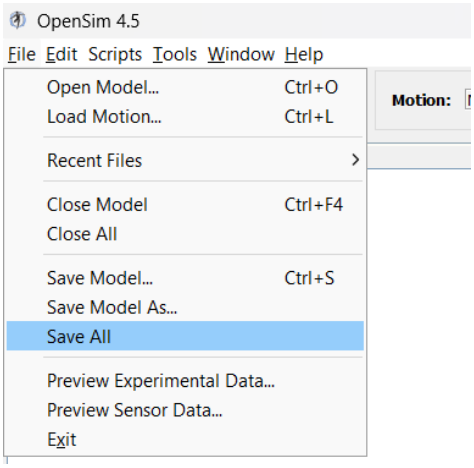


Fig. 24 Como guardar archivos desde OpenSim

3. Escalado

Escalar un modelo consiste en hacer coincidir los parámetros antropométricos de este con los de un sujeto de estudio, para ello es necesario contar con una serie de archivos que ingresarán a la herramienta de escalado de OpenSim, como se ve en la **Fig. 25**. Dado que la guía se enfoca en simplificar el trabajo para el usuario, solo se explicará el procedimiento de escalado automático realizado por el software y no el manual.

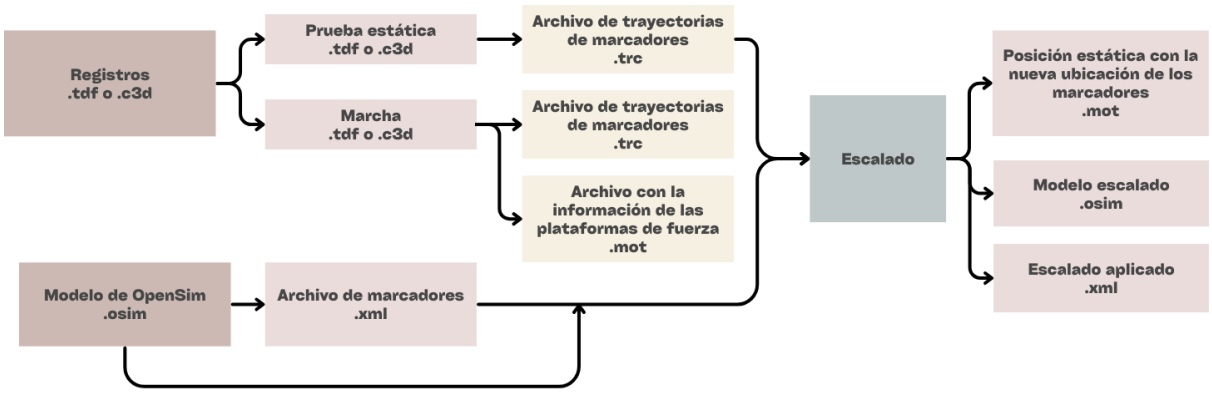


Fig. 25 Archivos de entrada y salida de la herramienta de escalado de OpenSim

3.1. Archivo de configuración

Para utilizar la herramienta *Scale Model* de OpenSim es necesario editar o crear un archivo en formato .xml con una estructura similar a la que se ve en la **Fig. 26**, la forma más directa de obtener este archivo es editar el archivo de configuración (SetUp), que viene en los modelos Gait2354 o Gait2392 de OpenSim, por lo que se explicará qué debe de ir en cada sección del archivo y qué partes es necesario editar para poder realizar el escalado del modelo. Inicialmente realice una copia del archivo subject01_Setup_Scale.xml, que se encuentra en los modelos previamente mencionados, se recomienda cambiar parcialmente el nombre del archivo, reemplazando *subject01* por el nombre o código de su sujeto de prueba. Posteriormente abra el archivo con un editor de código o texto, como sugerencia puede usar Visual Studio Code, Notepad++ o algún otro editor de su preferencia.

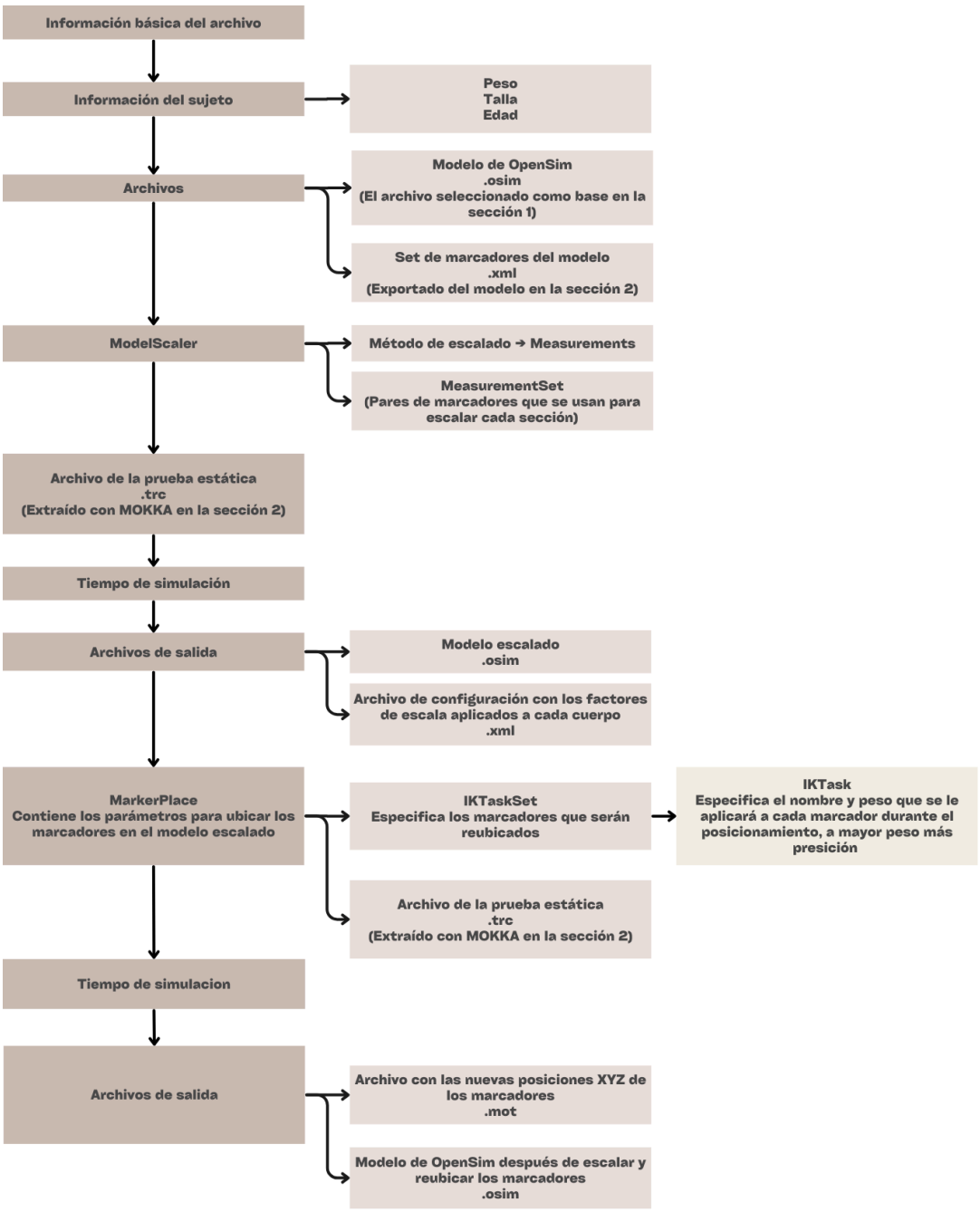


Fig. 26 Estructura del archivo de configuración de la herramienta de escalado de OpenSim

3.1.1. Edición de información general

Con el archivo abierto lo primero será cambiar el nombre del modelo para que contenga el nombre o código del sujeto con el que se está trabajando, posteriormente agregar la talla en milímetros y el peso en kilogramos del sujeto, la

edad es un parámetro puramente informativo por lo que no tiene efecto sobre el escalado. Posteriormente se debe cambiar el archivo del modelo por el nombre con el que se guardó el modelo de OpenSim (.osim) con el que trabajará, **Fig. 27**. Cabe mencionar que tanto el archivo del modelo de OpenSim, así como el de configuración de escalado, el archivo con los marcadores, los archivos .trc y .mot y los archivos que se crearán en las siguientes secciones deben estar en la misma carpeta.

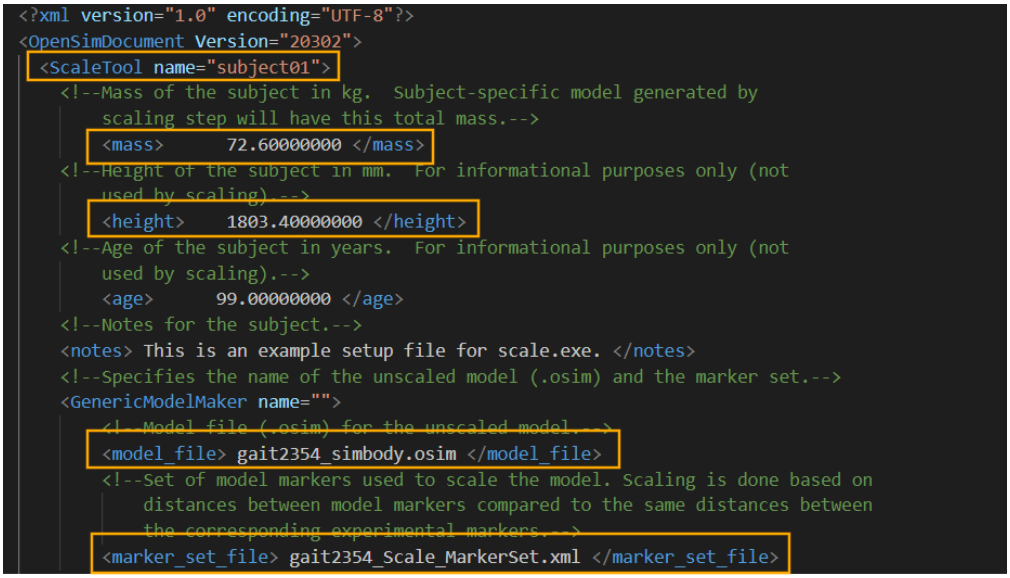


Fig. 27 Como editar el archivo de configuración de escalado

3.1.2. ModelScaler

Esta sección es la que se encargará de realizar el escalado, por lo cual lo primero será modificar el método de escalado en *scaling_order*, para que solo contenga *measurements* y no *manualScale*, esto permitirá que OpenSim calcule los factores de escala de todos los segmentos y los use para el escalado **Fig. 28**. Posteriormente encontrará la subsección de *MeasurementSet*, dentro de esta se describe cada *Measurement*, es decir, las medidas con las que se escalará cada segmento y el par de marcadores para calcular el factor de escala.

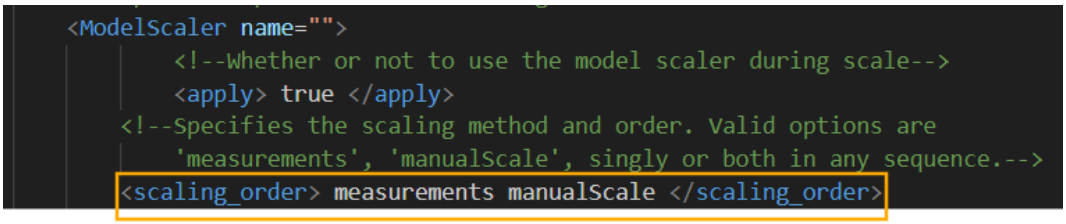


Fig. 28 Etiquetas para definir el tipo de escalado

En la **Fig. 29,** se presenta como ejemplo el cuerpo *torso*, el cual tiene la sección *apply* marcada con *true*, lo que significa que será escalado, entre el par de marcadores *V.Sacral* y *Top.Head*, si revisa sus archivos de marcadores experimentales (.trc) y del modelo (.xml) se dará cuanta que en la parte superior está el nombre de cada marcador y corresponde con el nombre que se coloca en esta sección **Fig. 30,** cabe mencionar que el número de columnas debe ser igual al número de marcadores usados durante la prueba estática, más dos columnas adicionales que contiene el número del cuadro (*Frame* en inglés) y el tiempo (**Fig. 30b**). En caso de que el nombre no coincida remítase a la sección 2 para saber cómo editarlo, adicionalmente, si usted no desea escalar entre este par de marcadores, sino por ejemplo entre *V.Sacral* y un marcador ubicado en la vértebra cervical *C7*, solo debe cambiar el nombre en esta sección, **Fig. 31.** Asumiendo que ya ha posicionado el marcador virtual en el modelo como se explicó anteriormente y que su sujeto de prueba también cuenta con él, por lo que está en el .trc de los registros.

```
<MeasurementSet name="gait2354">
    <objects>
        <Measurement name="torso">
            <!--Flag to turn on and off scaling for this measurement.-->
            <apply> true </apply>
            <!--Set of marker pairs used to determine the scale factors.-->
            <MarkerPairSet name="">
                <objects>
                    <MarkerPair name="">
                        <!--Names of two markers, the distance between which is used to compute a
                            body scale factor. -->
                        <markers> V.Sacral Top.Head </markers>
                    </MarkerPair>
                </objects>
                <groups/>
            </MarkerPairSet>
            <!--Set of bodies to be scaled by this measurement.-->
            <BodyScaleSet name="">
                <objects>
                    <BodyScale name="torso">
                        <!--Axes (X Y Z) along which to scale a body. For example, 'X Y Z' scales
                            along all three axes, and 'Y' scales just along the Y axis.-->
                        <axes> X Y Z </axes>
                    </BodyScale>
                </objects>
                <groups/>
            </BodyScaleSet>
        </Measurement>
```

Fig. 29 Como definir el cuerpo a escalar y los marcadores para realizarlo

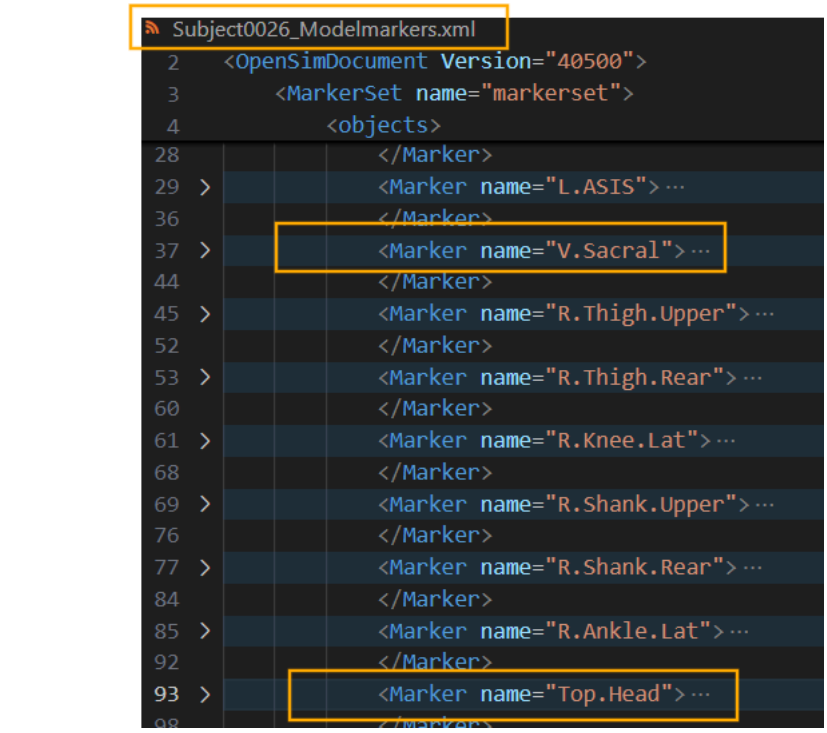


Fig. 30 a) Archivo de marcadores del modelo (.xml). **b)** Archivo de marcadores experimentales (.trc)

```
scaling_order> measurements manualScale </scaling_order>
    <MeasurementSet name="gait2354">
        <objects>
            <Measurement name="torso">
                <!--Flag to turn on and off scaling for this measurement.-->
                <apply> true </apply>
                <!--Set of marker pairs used to determine the scale factors.-->
                <MarkerPairSet name="">
                    <objects>
                        <MarkerPair name="">
                            <!--Names of two markers, the distance between which is used to compute a
                                body scale factor.-->
                            <markers> V.Sacral C7</markers>
                        </MarkerPair>
                    </objects>
                    <groups/>
                </MarkerPairSet>
                <!--Set of bodies to be scaled by this measurement.-->
                <BodyScaleSet name="">
                    <objects>
                        <BodyScale name="torso">
                            <!--Axes (X Y Z) along which to scale a body. For example, 'X Y Z' scales
                                along all three axes, and 'Y' scales just along the Y axis.-->
                            <axes> X Y Z </axes>
                        </BodyScale>
                    </objects>
                    <groups/>
                </BodyScaleSet>
            </Measurement>
```

a

```
Subject0026_Modelmarkers.xml
 2   <OpenSimDocument Version="40500">
 3       <MarkerSet name="markerset">
 4           <objects>
21               <Marker name="R.ASIS">
28               </Marker>
29               <Marker name="L.ASIS">…
36               </Marker>
37               <Marker name="V.Sacral">…
44               </Marker>
45               <Marker name="R.Thigh.Upper">…
59               </Marker>
60               <Marker name="L.Midfoot.Lat">…
67               </Marker>
68               <Marker name="C7">…
73               </Marker>
```

b

Fig. 31 Como definir los marcadores para escalar un cuerpo

Este paso se debe repetir y verificar para cada uno de los cuerpos del modelo, en este caso son 5 cuerpos, torso, pelvis, muslo, pierna y pie, **Fig. 32**, donde en los últimos 3 se agregan los pares de marcadores tanto para el lado derecho como para el lado izquierdo.

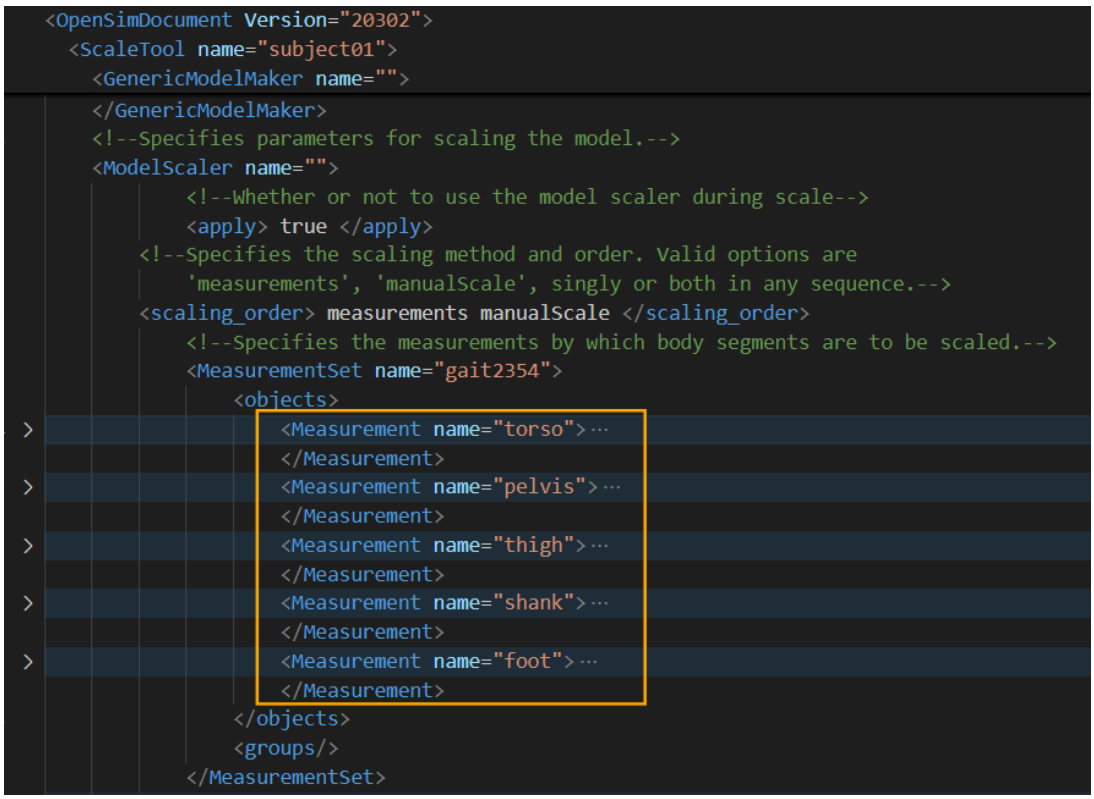


Fig. 32 Cuerpos de un modelo a escalar

Debajo de la etiqueta de cierre de *MeasurementSet* encontrará *ScaleSet,* como este es un archivo de configuración usado en los tutoriales de OpenSim tiene un apartado de factores de escala manual, que como se mencionó anteriormente no se emplea en esta guía para simplificar el proceso, por lo que solo debe eliminar toda esta información, **Fig. 33.**

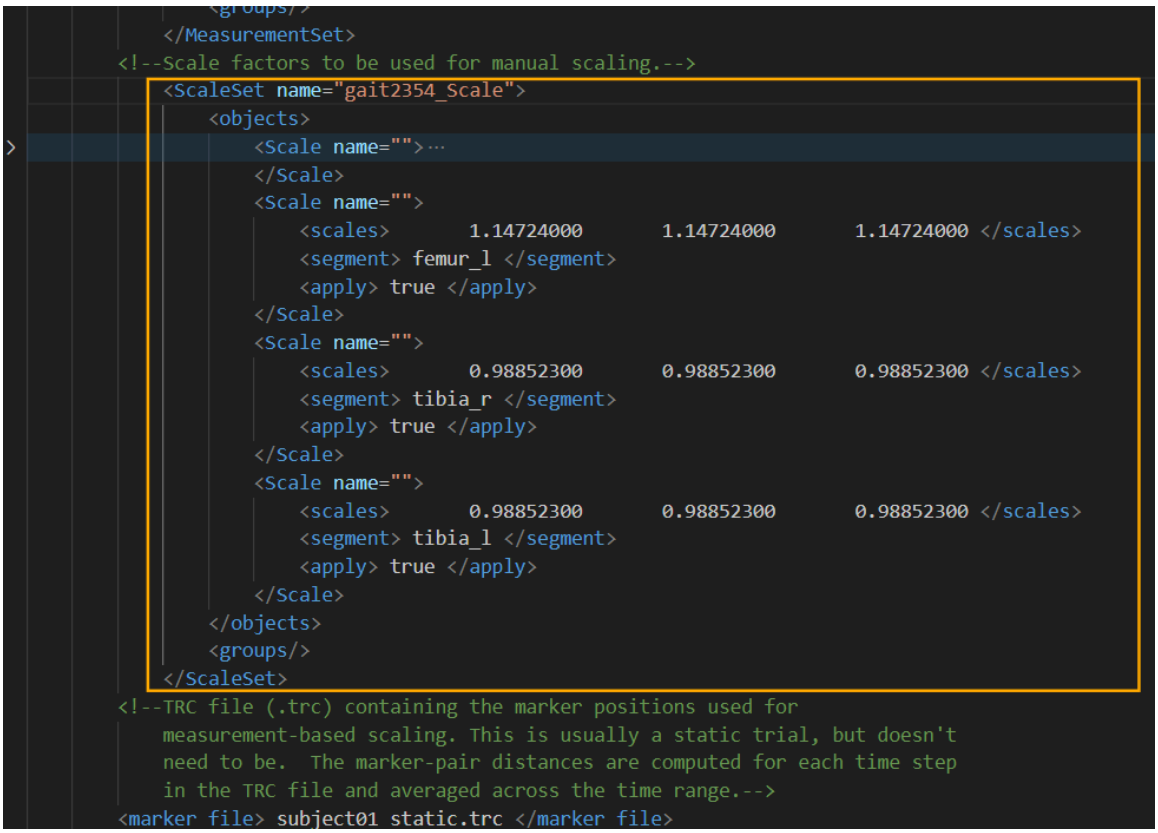

<groups/>
</MeasurementSet>
<!--Scale factors to be used for manual scaling.-->
<ScaleSet name="gait2354_Scale">
<objects>
<Scale name="">…
</Scale>
<Scale name="">
<scales> 1.14724000 1.14724000 1.14724000 </scales>
 femur_l
<apply> true </apply>
</Scale>
<Scale name="">
<scales> 0.98852300 0.98852300 0.98852300 </scales>
 tibia_r
<apply> true </apply>
</Scale>
<Scale name="">
<scales> 0.98852300 0.98852300 0.98852300 </scales>
 tibia_l
<apply> true </apply>
</Scale>
</objects>
<groups/>
</ScaleSet>
<!--TRC file (.trc) containing the marker positions used for
measurement-based scaling. This is usually a static trial, but doesn't
need to be. The marker-pair distances are computed for each time step
in the TRC file and averaged across the time range.-->
<marker_file> subject01_static.trc </marker_file>

Fig. 33 Sección a eliminar del archivo de escalado

En la sección 2 se explicó cómo obtener el .trc a partir del archivo de registro de la prueba estática, por lo que ese es el archivo que debe de colocarse en *marker_file*. Por otro lado, en *time_range* debe de ir el tiempo de simulación en segundos, en este caso está definido entre el segundo 1 y el segundo 2 del registro, puede modificarlo según su registro. Posteriormente se presenta la opción de preservar o no la distribución relativa de la masa entre los segmentos, se sugiere dejarla activada. Finalmente hay dos archivos de salida a los cuales se recomienda renombrar parcialmente sustituyendo *subject01* con el código o nombre de su sujeto de prueba, estos archivos contienen el modelo escalado y los factores de escala que se aplicaron al modelo sin escalar. Los puntos anteriormente mencionados se presentan en la **Fig. 34.**

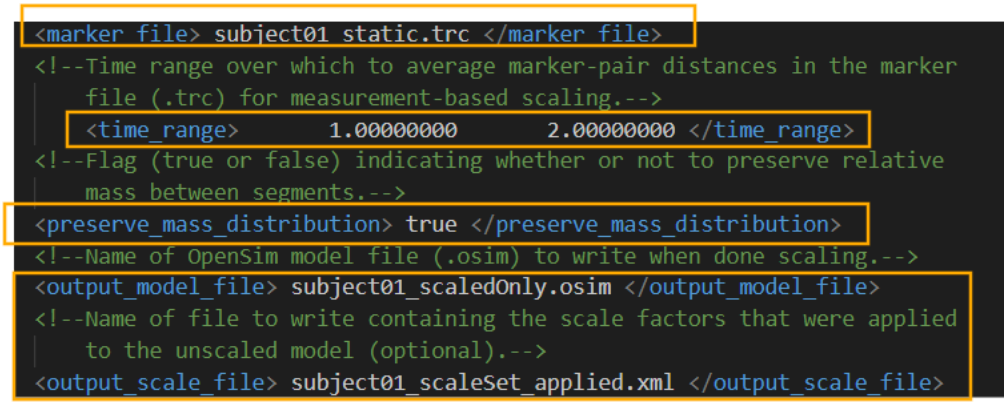


Fig. 34 Parámetros a editar en el archivo de escalado

3.1.3. MarkerPlace

En esta sección del archivo se especifican los parámetros para posicionar los marcadores en el modelo escalado, como se puede ver en la **Fig. 35,** en cada *IKMarkerTask* hay un marcador (deben ser los mismos nombres que están en su archivo .trc de la prueba estática), por lo que verifique si debe cambiar el nombre de alguno, eliminar los que no use su sujeto o agregar extras, para esto es suficiente con cambiar la parte de *name* en *IKMarkerTask* como se muestra en la **Fig. 36,** y en caso de requerir un nuevo marcador solo copie y pegue uno existente y renómbrelo, considerando el peso, el cual es un valor que indica con cuánta precisión se va a buscar una coincidencia entre el marcador experimental y el marcador virtual.

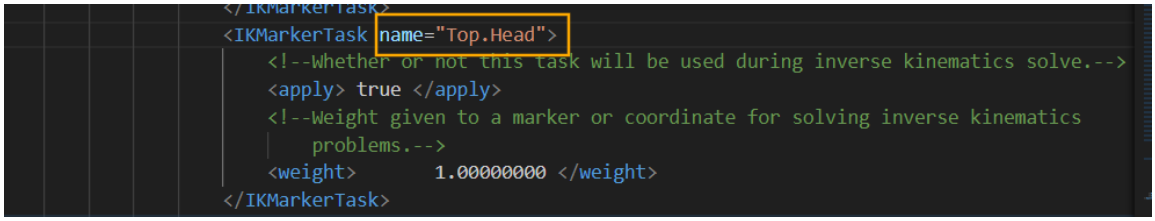


Fig. 35 Etiquetas *IKMarkerTask*

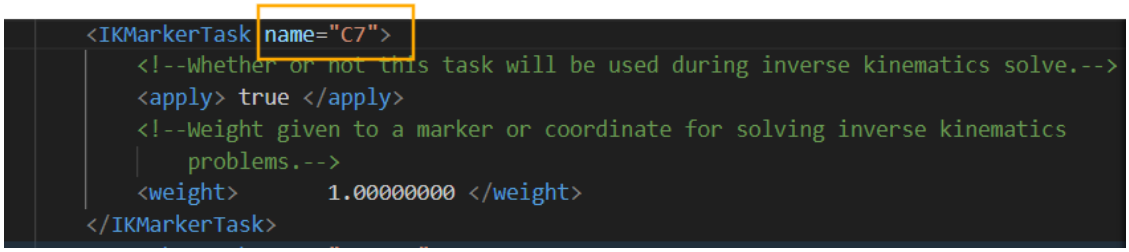


Fig. 36 Crear etiquetas *IKMarkerTask* para un nuevo marcador

Tomando como ejemplos los marcadores *Sternum* y *R.Knee.Lat*, **Fig. 37**, se observa que el peso de ambos es diferente, 1 y 1000 respectivamente, esto hace referencia a la precisión con la que el software intentará ubicar el marcador experimental en el

modelo, a mayor peso mayor precisión, en este caso, los marcadores relacionados con la zona distal del cuerpo, a partir de la pelvis, tienen un mayor peso debido a que se requiere recrear mejor el movimiento captado en los registros, por lo que en caso de copiar y pegar un marcador existente para agregar un marcador que está en los registros, tenga en cuenta el peso que va a escoger según el objetivo de su estudio.

```
<IKMarkerTask name="Sternum">
    <!--Whether or not this task will be used during inverse kinematics solve.-->
    <apply> true </apply>
    <!--Weight given to a marker or coordinate for solving inverse kinematics
        problems.-->
    <weight>       1.00000000 </weight>
</IKMarkerTask>
```

a

```
<IKMarkerTask name="R.Knee.Lat">
    <!--Whether or not this task will be used during inverse kinematics solve.-->
    <apply> true </apply>
    <!--Weight given to a marker or coordinate for solving inverse kinematics
        problems.-->
    <weight>    1000.00000000 </weight>
</IKMarkerTask>
```

b

Fig. 37 Pesos dentro de las etiquetas *IKMarkerTask*

Finalmente, en esta sección también están los *IKCoordinateTask,* que contienen ángulos de algunas articulaciones para posicionar al sujeto adecuadamente según el registro de los marcadores, para simplicidad del modelo se sugiere dejarlos tal cual.

3.1.4. Edición de información final

Por último, solo es necesario volver a escribir el nombre del archivo .trc de la prueba estática, así como definir el tiempo de simulación o dejar el que está por defecto, siempre y cuando este tiempo exista en su archivo. Adicionalmente, debe de renombrar los archivos de salida .mot y .osim, los cuales contienen las nuevas posiciones de los marcadores para la pose estática del modelo y el nuevo modelo de OpenSim escalado y con los marcadores posicionados, respectivamente, **Fig. 38**.

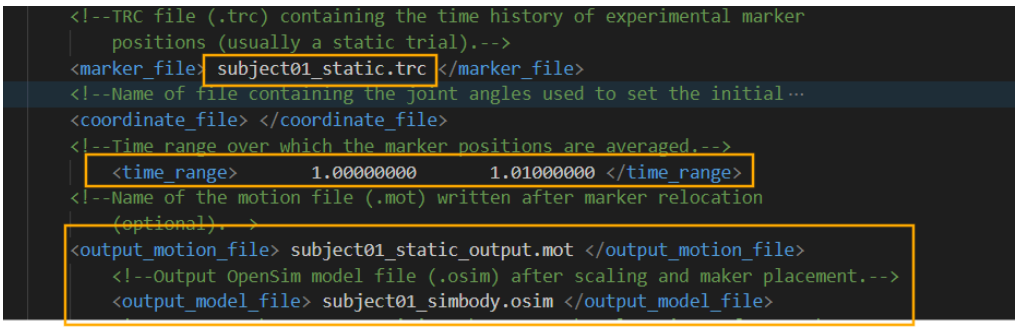
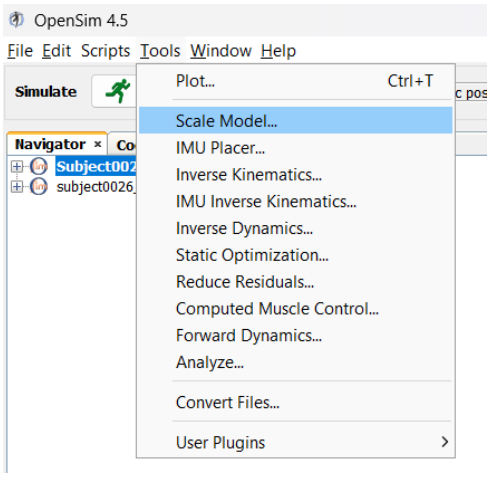


Fig. 38 Parámetros a editar en el archivo de escalado

Cabe destacar que es importante guardar el archivo periódicamente o activar la función de autoguardado si su editor de texto la posee, con el fin de no perder el proceso de edición.

3.2. Ejecutar la herramienta de escalado

Una vez tenga su archivo de configuración para escalado (*ScaleSetup.xml*) guardado, solo debe ir a OpenSim y abrir el modelo sobre el que va a trabajar, posteriormente presione *Tools→Scale Model…,* en la ventana emergente presione *Load* y busque el archivo que acabó de crear, revise que todos los parámetros se carguen adecuadamente y presione *Run,* **Fig. 39.**



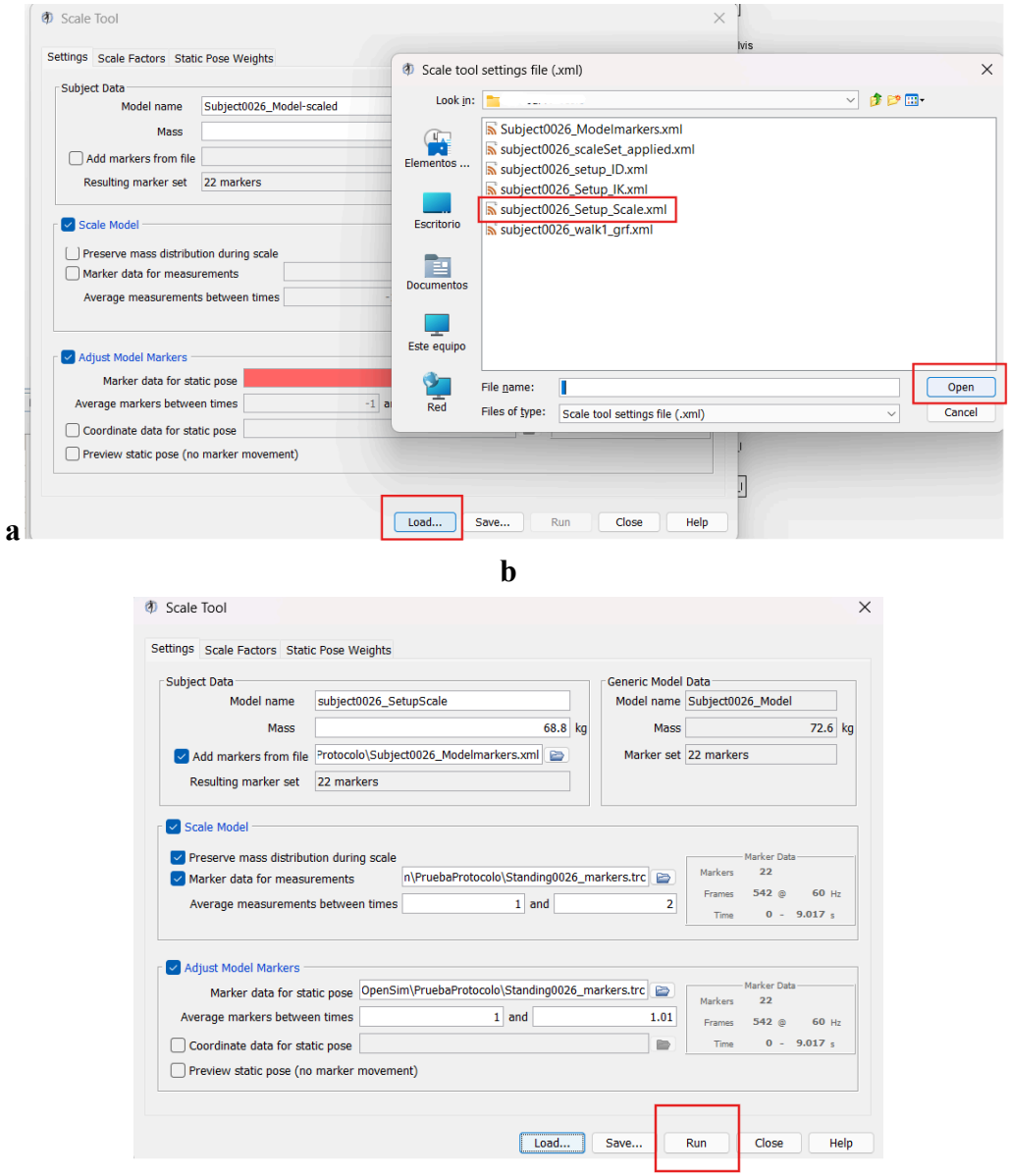


Fig. 39 Pasos para ejecutar la herramienta de escalado

En caso de que no se habilite el botón de *Run*, verifique que el número de marcadores en las secciones *Subject Data* y *Generic Model Data* tengan el mismo número de marcadores, **Fig. 39c**. De lo contrario revise la cantidad de marcadores tanto del archivo del modelo (.xml) como de la prueba estática de su sujeto (.trc) estos deben coincidir tanto en cantidad como en nombre.

Cuando la herramienta termine de ejecutarse, puede notar que tanto en la ventana *Navigator* como en la *Visualizer Window* se abre un nuevo modelo el cual está escalado y con los marcadores reposicionados, **Fig. 40**.

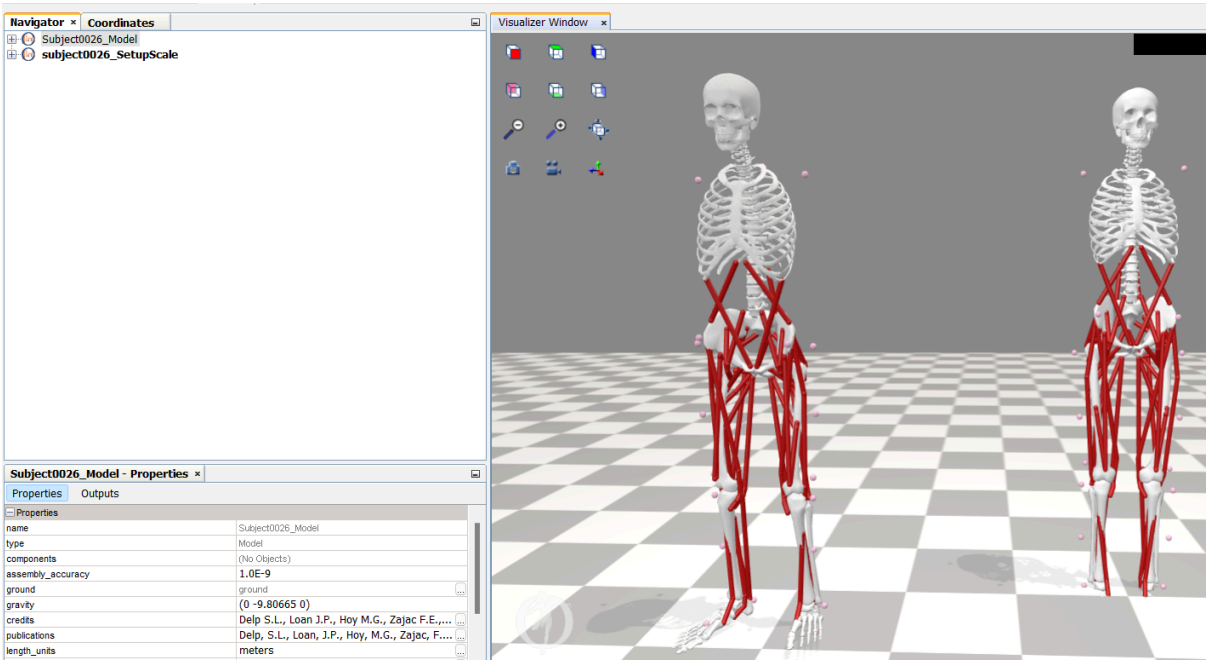


Fig. 40 Nuevo modelo escalado

3.3. Evaluar el resultado del escalado

Una vez haya finalizado el proceso de escalado es posible verificar que las medidas antropométricas del modelo correspondan con las del sujeto de prueba. Inicialmente coloque al sujeto en la posición por defecto, en la ventana *Coordinates* selecciones *Poses* → *Default* (**Fig. 41**). Adicionalmente, es necesario crear dos marcadores nuevos cuyo *PhysicalFrame: parent_frame* sea *ground* (Para saber cómo agregar marcadores puede remitirse a la sección 2), ahora bien, mueva los marcadores entre puntos estratégicos como la cabeza y el tobillo, ambos lados de la pelvis, la cadera y el tobillo, entre otros, **Fig. 42**, con el fin de conocer las coordenadas de estos y por medio de la distancia entre ambos puntos encontrar la longitud de ese segmento, **Fig. 43**, esto le permitirá comparar con las medidas que tiene de los segmentos de su sujeto y calcular el porcentaje de error.

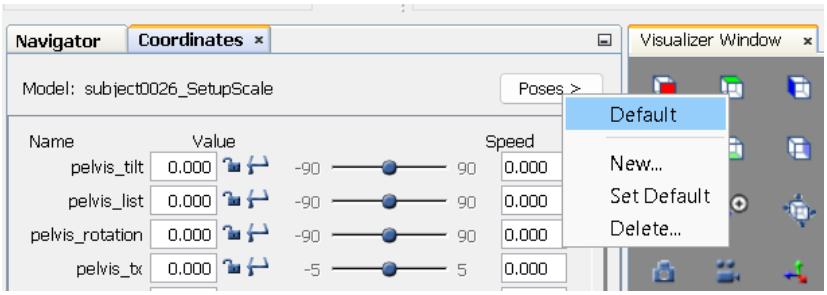


Fig. 41 Recuperar la posición por defecto de un modelo

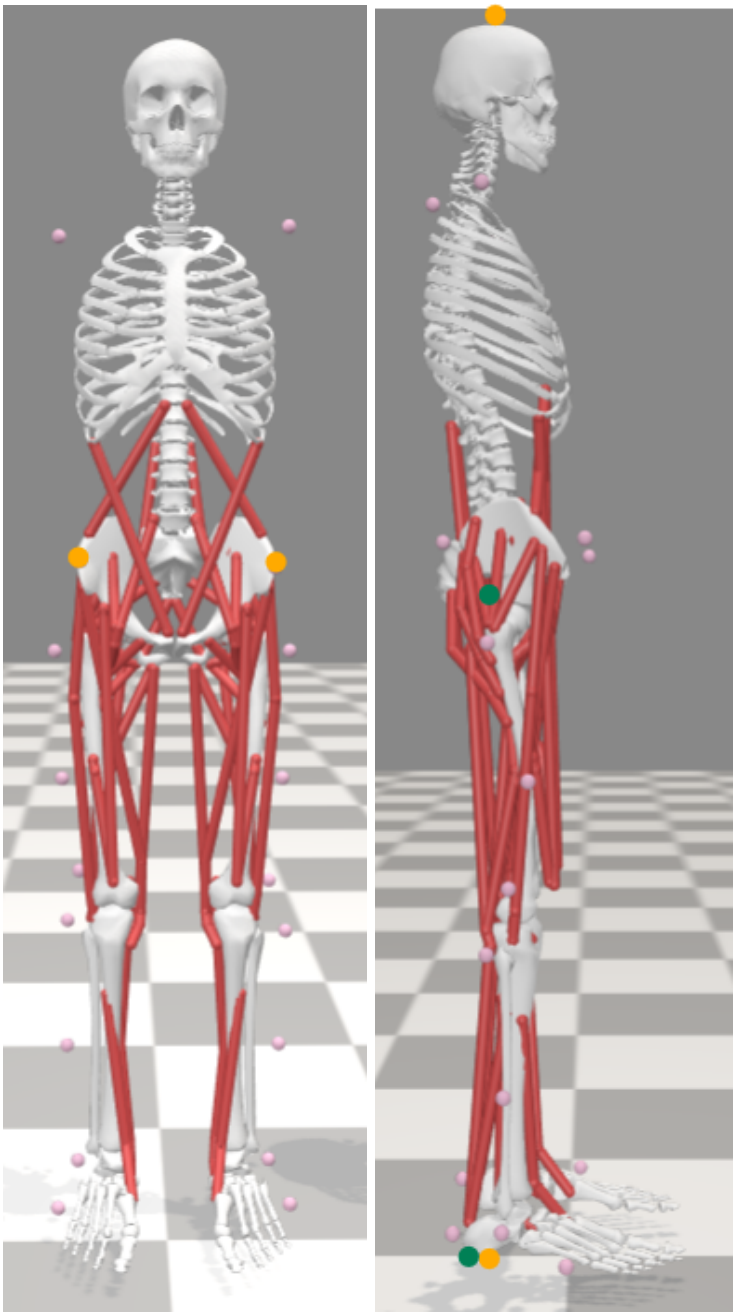


Fig. 42 Marcadores para evaluar la calidad del escalado de un modelo

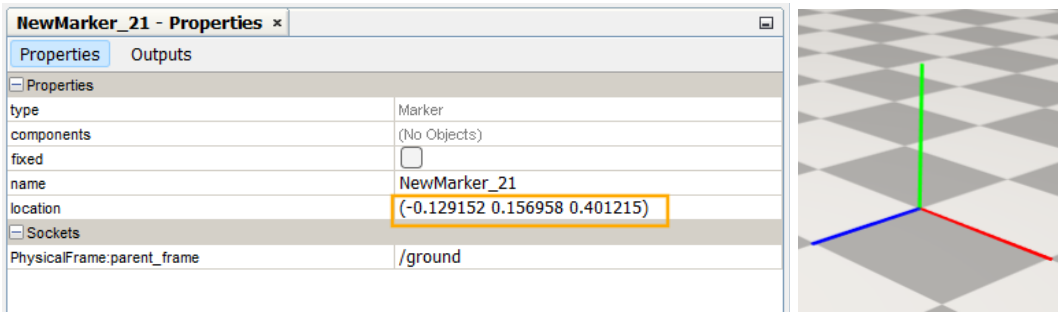


Fig. 43. Coordenadas de un marcador

Nota. Las coordenadas están escritas en orden XYZ (Rojo, Verde, Azul)

4. Cinemática inversa

Esta herramienta de OpenSim procesa para cada instante de tiempo del registro de marcha la posición en XYZ de los marcadores y calcula las coordenadas generalizadas que colocan al modelo en la pose que mejor se ajusta a los datos experimentales en ese instante [5]. Esta simulación parte del modelo previamente escalado, además emplea el archivo con las posiciones de los marcadores durante el registro de marcha (.trc) adquirido con MOKKA en la sección 1, en la **Fig. 44** se presenta un gráfico que describe las entradas y salidas del estudio.

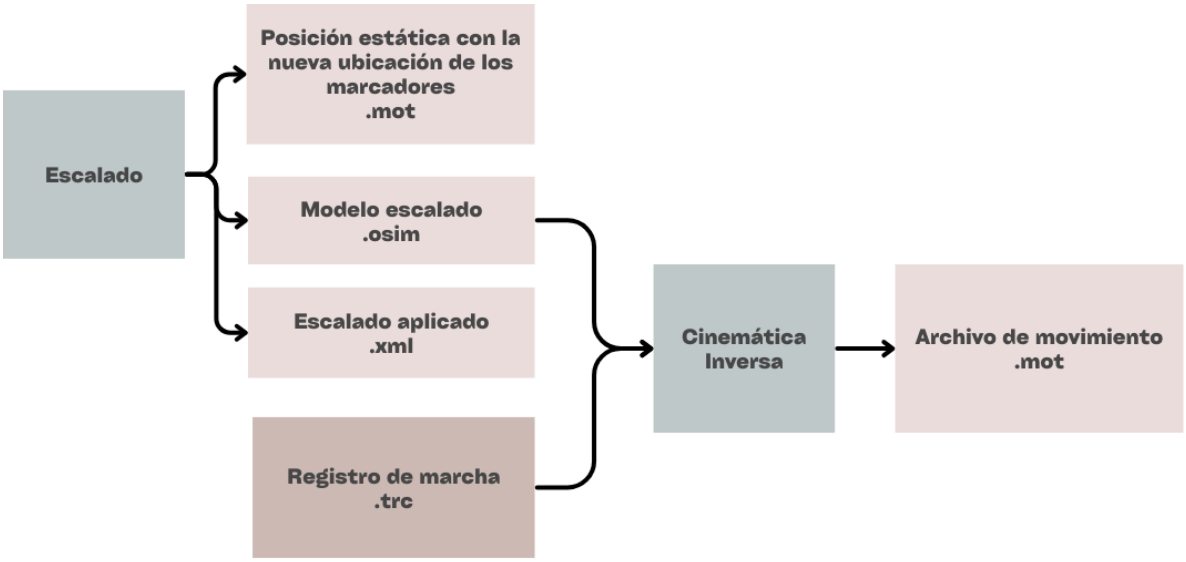


Fig. 44 Archivos de entrada y salida de la herramienta de cinemática inversa de OpenSim

Para ejecutar la herramienta de cinemática inversa se debe partir de un modelo posicionado lo más semejante posible a la posición inicial del sujeto de prueba, para ello puede usar los controles deslizantes de la ventana *Coordinates* en OpenSim, **Fig. 45,** adicionalmente, la posición debe coincidir con la ubicación en el espacio tridimensional, es decir, el modelo debe estar orientado en XYZ igual que en el registro, para verificar

que esta orientación coincida puede hacer uso de la herramienta de previsualización de información experimental que ofrece OpenSim, para ello, con el modelo escalado abierto haga uso de la siguiente ruta: *File → Preview Experimental Data…,* **Fig. 46,** esto abrirá su buscador de archivos, seleccione el archivo con la trayectoria de los marcadores del registro de la prueba estática, una vez cargado lo verá cerca al modelo como una nube de esferas azules, donde cada esfera es un marcador experimental, use los controladores de *Coordinates* para desplazar el modelo y hacer coincidir lo más que pueda ambas posiciones, si desea ajustar la posición de los marcadores del modelo para que correspondan mejor con los marcadores experimentales puede hacerlo siguiendo las recomendaciones de la sección 2.6, recuerda volver a guardar el archivo, **Fig. 47.** Una vez que ambas posiciones sean iguales puede guardarla como la posición por defecto de su modelo haciendo clic en *Poses* > → *Set Default,* **Fig. 48.**

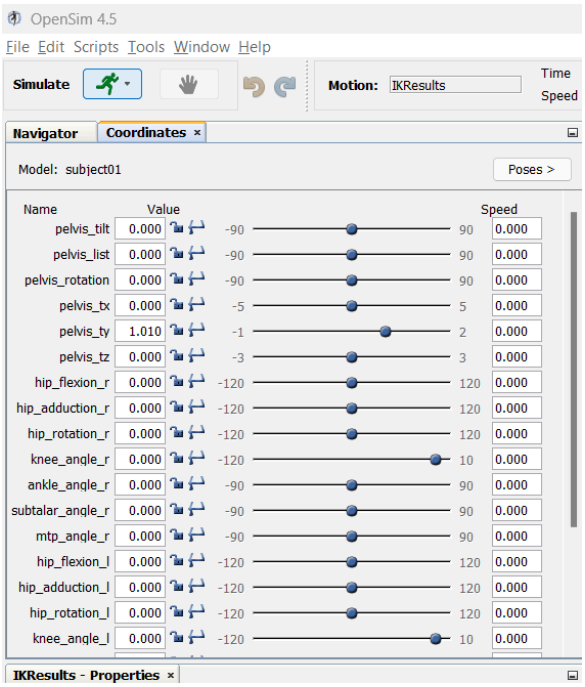


Fig. 45 Ventana de *Coordinates*

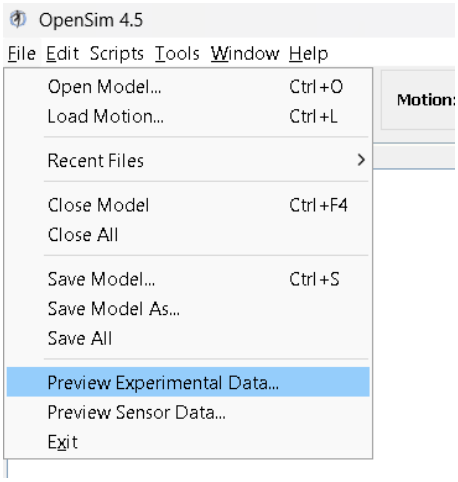


Fig. 46 Previsualizar marcadores experimentales

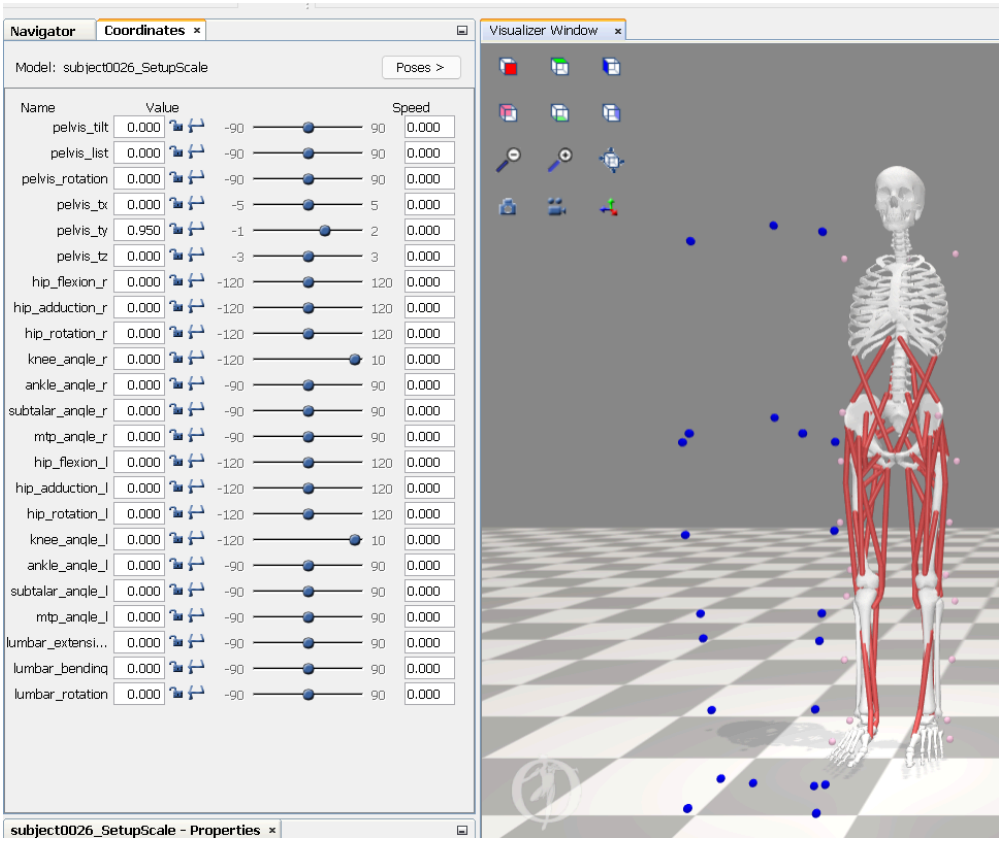


Fig. 47 Visualización de marcadores experimentales

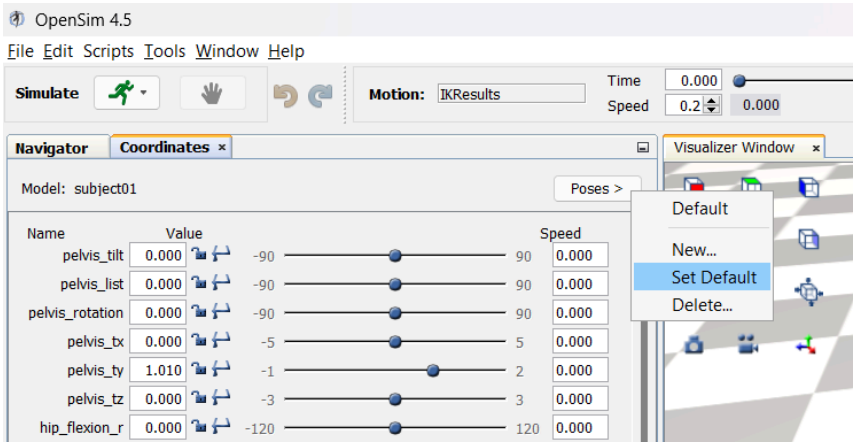


Fig. 48 Reasignar posición por defecto del modelo

Emplee la lista de chequeo proporcionada a continuación, con la cual podrá verificar si cuenta con los archivos necesarios para realizar las siguientes partes de la guía, en caso de no contar con alguno, revise los pasos anteriores para conocer cuál le hace falta y poder ejecutar la simulación de cinemática inversa.

- ☐ Archivo .osim con el modelo seleccionado → Sección 1
- ☐ Archivo .c3d o .tdf con el registro de marcha del sujeto
- ☐ Archivo .c3d o .tdf con el registro de prueba estática del sujeto
- ☐ Archivo .trc con el registro de marcha del sujeto → Sección 2.2
- ☐ Archivo .trc con el registro de prueba estática del sujeto → Sección 2.2
- ☐ Archivo .xml con el set de marcadores del modelo → Sección 2
- ☐ Archivo .xml con la configuración del escalado → Sección 3
- ☐ Archivo .osim con el modelo escalado y los marcadores de la pose estática ubicados (Resultado del escalado)→ Sección 3.2
- ☐ Archivo .mot con la posición estática del modelo (Resultado del escalado) → Sección 3.2
- ☐ Archivo .xml con los factores de escala aplicados al modelo (Resultado del escalado) → Sección 3.2

4.1. Archivo de configuración de cinemática inversa

Para ejecutar la herramienta de cinemática inversa es necesario tener un archivo de configuración con una estructura similar a la que se muestra en la **Fig. 49,** este archivo básicamente tiene el nombre de los marcadores tal cual están nombrados tanto en el modelo como en el archivo de marcha .trc del sujeto, así como el peso de cada marcador, es decir, con cuánta precisión se intentará hacer coincidir el modelo con los datos experimentales.

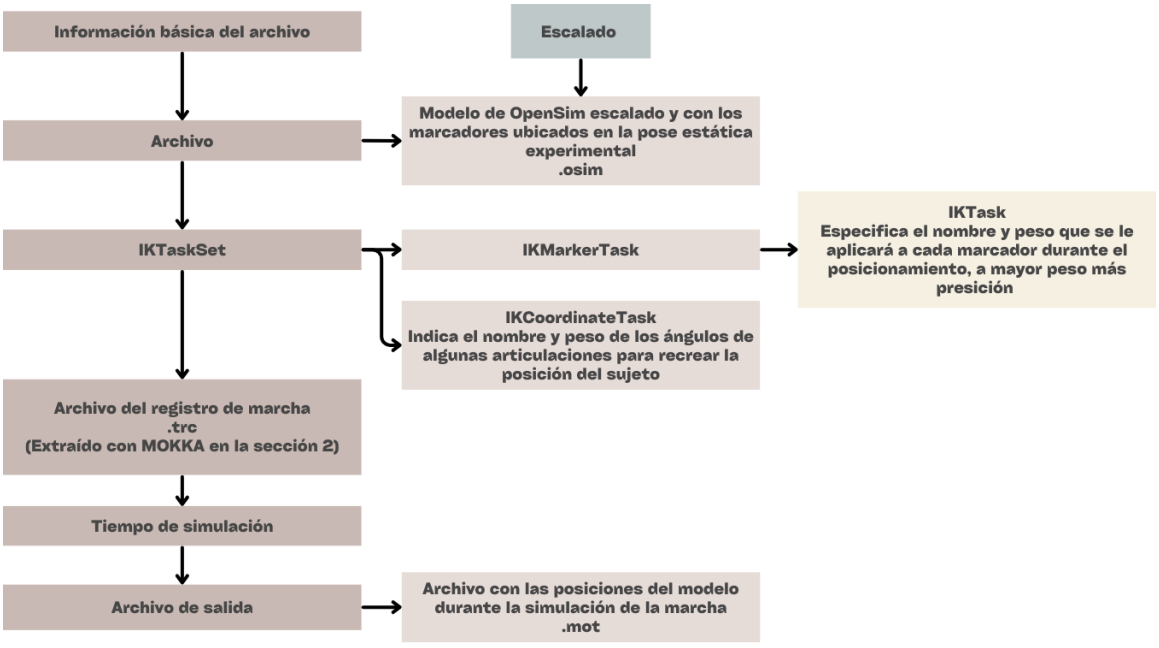


Fig. 49 Estructura del archivo de configuración de la herramienta de cinemática inversa de OpenSim

4.1.1. Creación y edición del archivo de cinemática inversa

Inicialmente realice una copia del archivo *subject01_Setup_IK.xml*, que se encuentra en la carpeta de *Gait2354_Simbody* de los modelos de OpenSim, esto con el fin de no tener que crear el archivo de configuración desde cero, se recomienda cambiar parcialmente el nombre del archivo, reemplazando *subject01* por el nombre o código de su sujeto de prueba. Posteriormente abra el archivo con un editor de texto o código.

Coloque el nombre del modelo .osim que obtuvo como salida del proceso de escalado y ubicación de los marcadores en la pose estática entre las etiquetas *model_file,* como se observa en la **Fig. 50,** en esta también se encuentra marcado un *IKMarkerTask*, en el cual se halla el nombre de un marcador, este debe ser exactamente igual al nombre con el que están los marcadores en el archivo del registro de marcha (.trc) y del modelo, por lo que reemplácelo por el nombre adecuado según su modelo, adicionalmente se debe seleccionar el peso de este marcador dentro de la simulación, a mayor peso mayor será la precisión con la que se ubique en el modelo para replicar la posición experimental, por lo que revise el peso asignado a cada marcador, modifique todos los *IKMarkerTask* del archivo con los nombres de sus marcadores, bien sea editando, eliminando o agregando según sea necesario.

```
subject0026_Setup_IK.xml
<?xml version="1.0" encoding="UTF-8" ?>
<OpenSimDocument Version="30000">
    <InverseKinematicsTool name="subject0026IK">
        <!--Name of the .osim file used to construct a model.-->
        <model_file>subject0026_simbody.osim</model_file>
        <!--A positive scalar that is used to weight the importance of satisfying constraints.A weighting of
        'Infinity' or if it is unassigned results in the constraints being strictly enforced.-->
        <constraint_weight>20</constraint_weight>
        <!--The accuracy of the solution in absolute terms. I.e. the number of significantdigits to which the solution can be trusted.-->
        <accuracy>1e-005</accuracy>
        <!--Markers and coordinates to be considered (tasks) and their weightings.-->
        <IKTaskSet name="gait2354_IK">
            <objects>

                <IKMarkerTask name="R.Acromium">
                    <!--Whether or not this task will be used during inverse kinematics solve.-->
                    <apply>true</apply>
                    <!--Weight given to a marker or coordinate for solving inverse kinematics problems.-->
                    <weight>0.5</weight>
                </IKMarkerTask>
                <IKMarkerTask name="L.Acromium">
                    <!--Whether or not this task will be used during inverse kinematics solve.-->
                    <apply>true</apply>
                    <!--Weight given to a marker or coordinate for solving inverse kinematics problems.-->
                    <weight>0.5</weight>
                </IKMarkerTask>
```

Fig. 50 Parámetros a editar en el archivo de cinemática inversa

Al finalizar este proceso, tendrá tantos *IKMarkerTask* como marcadores tenga su modelo, como se ve en la **Fig. 51.**

```
<IKTaskSet name="gait2354_IK">
    <objects>

        <IKMarkerTask name="R.Acromium">…
        </IKMarkerTask>
        <IKMarkerTask name="L.Acromium">…
        </IKMarkerTask>
        <IKMarkerTask name="C7">…
        </IKMarkerTask>
        <IKMarkerTask name="R.ASIS">…
        </IKMarkerTask>
        <IKMarkerTask name="L.ASIS">…
        </IKMarkerTask>
        <IKMarkerTask name="V.Sacral">…
        </IKMarkerTask>
        <IKMarkerTask name="R.Thigh.Upper">…
        </IKMarkerTask>

        <IKMarkerTask name="R.Thigh.Rear">…
        </IKMarkerTask>
        <IKMarkerTask name="R.Shank.Upper">…
        </IKMarkerTask>
```

Fig. 51 Secciones de *IKMarkerTask*

Posteriormente se encuentra la sección de *IKCoordinateTask,* allí se describen los nombres de los ángulos de algunas articulaciones y los pesos, puede modificar los pesos en caso de ser necesario, sin embargo, se sugiere dejarlo tal cual para simplificar la simulación.

Finalmente, como se presenta en la **Fig. 52,** es necesario agregar el archivo donde está la posición de los marcadores en cada instante del registro de marcha, es decir, el .trc extraído durante la sección 1, adicionalmente modifique la duración de la simulación según la duración de su registro y cambie el nombre del archivo de salida a uno que corresponda con el nombre de su sujeto, este archivo será empleado en el estudio de dinámica inversa y es donde se escriben los resultados de la simulación de cinemática.

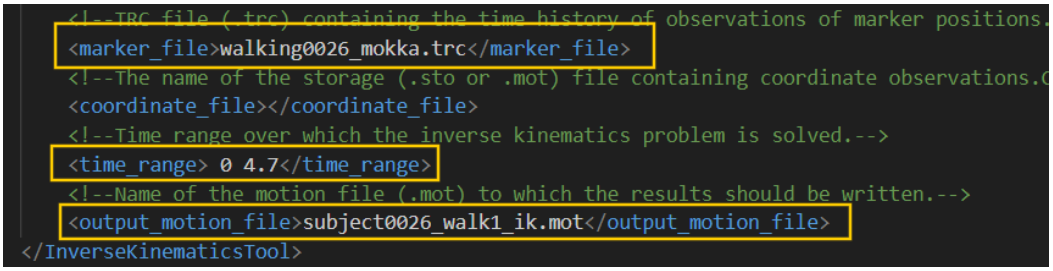
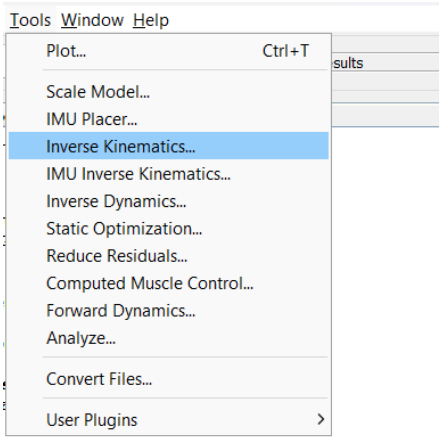


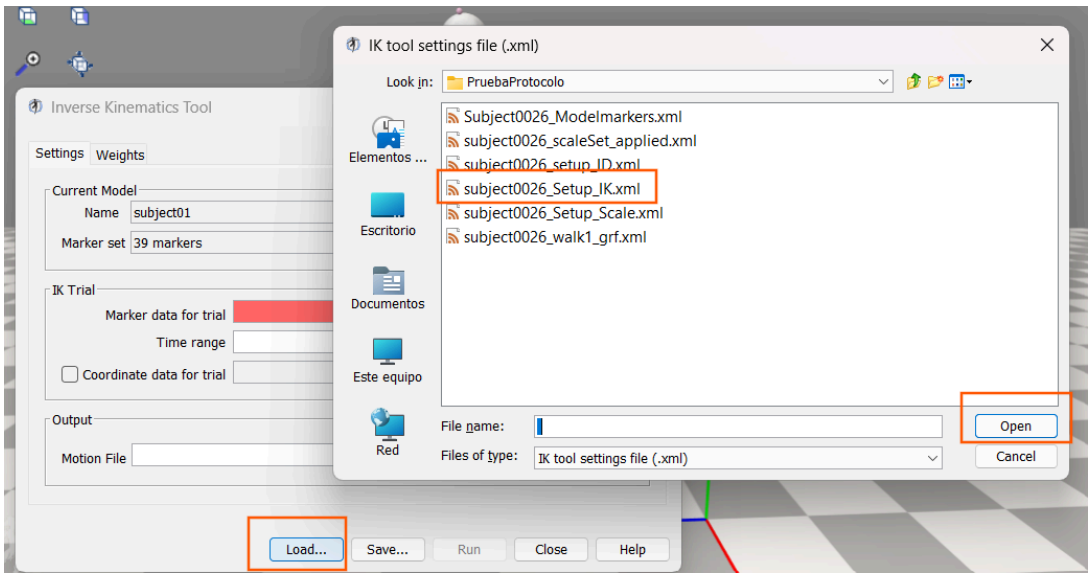
Fig. 52 Parámetros a editar en el archivo de cinemática inversa

4.2. Ejecutar la herramienta de cinemática inversa

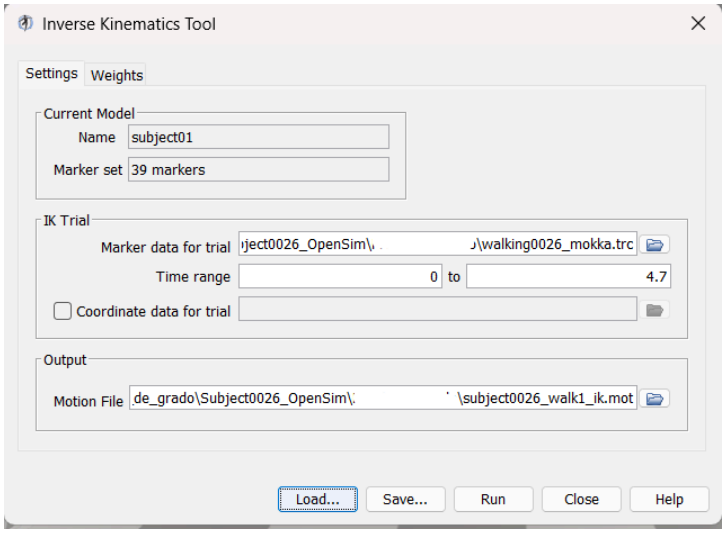
Una vez tenga listo el archivo de configuración basta con ir a OpenSim, abrir el archivo del modelo escalado y con los marcadores reubicados, en caso de que no se encuentre abierto, luego, seguir la ruta *Tools → Inverse Kinematics → Load,* la cual abrirá su explorador de archivos para cargar el archivo de configuración creado, posteriormente presione *run,* **Fig. 53.**



a



b



c

Fig. 53 Ejecutar la herramienta de cinemática inversa

4.3. Evaluar el resultado de cinemática inversa

Para comprobar que la simulación se realizó correctamente comience por revisar el movimiento del sujeto activando la reproducción en bucle en la barra superior (**Fig. 54**), posteriormente presione clic derecho sobre *IKResults* en la ventana de *Navigator*, seleccione la opción de *Associate Motion Data…* y cargue el archivo de los marcadores del registro de marcha .trc (obtenido en la sección 1), **Fig. 55**, en azul se presentarán los marcadores experimentales y en rosado los marcadores del modelo, allí podrá ver que tan ajustados están **Fig. 56**.

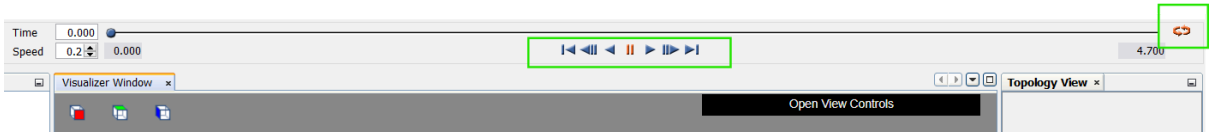


Fig. 54 Controladores de movimiento

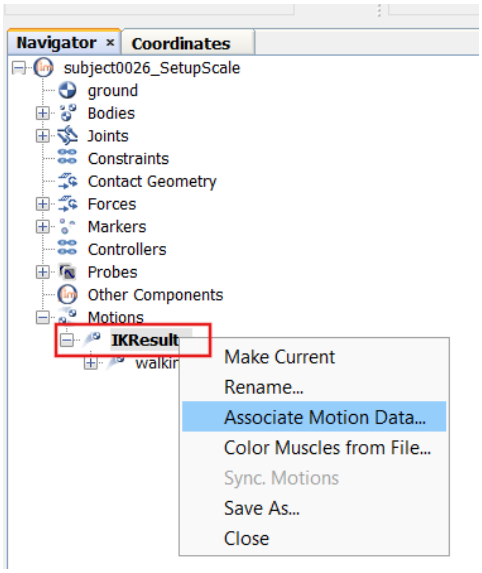


Fig. 55 Cargar archivos para evaluar cinemática inversa

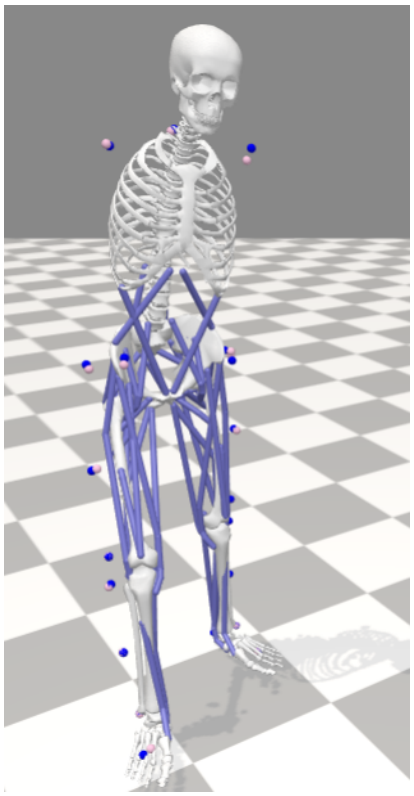


Fig. 56 Visualización de marcadores experimentales y marcadores posicionados sobre el modelo con IK

Adicionalmente, la ventana de *Messages* de OpenSim, **Fig. 57,** permite revisar los valores del error cuadrático total y la raíz cuadrática media (RMS por sus siglas en inglés), del marcador con mayor error de ajuste en cada fotograma. El error cuadrático total indica el error total del conjunto de datos, mientras que el RMS da cuenta del error promedio por dato. Según la guía de buenas prácticas de OpenSim, el valor de RMS no debería superar los 2 cm, mientras que el error máximo de marcador debería estar por debajo de 2-4cm [6]

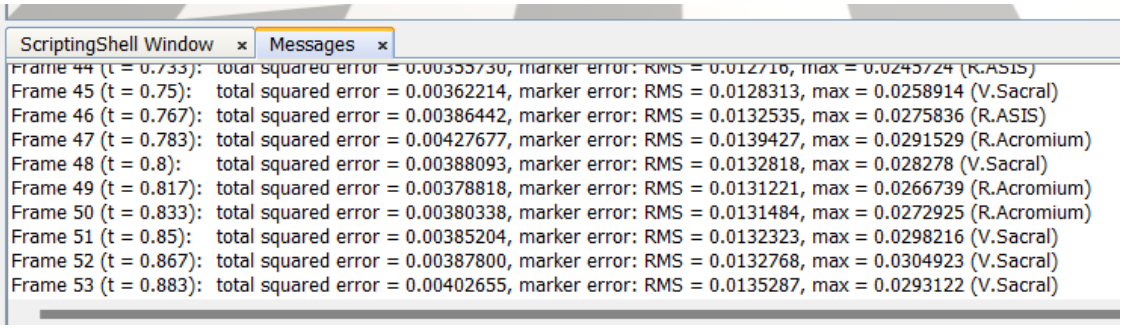


Fig. 57 Ventana de *Messages*

5. Dinámica inversa

Esta herramienta de OpenSim permite obtener las fuerzas netas, así como los torques que producen movimiento en cada articulación [7]. Partiendo de un modelo previamente escalado y al cual se le realizó un estudio de cinemática inversa es posible implementar este análisis mediante dos archivos de configuración. Como se presenta en la **Fig. 58**, se requiere no solo el archivo de salida del estudio de cinemática, sino también el archivo .mot que contiene la información extraída de las plataformas de fuerza empleadas durante el registro, cuyo proceso para obtenerlo se explicó en la sección 2.2.

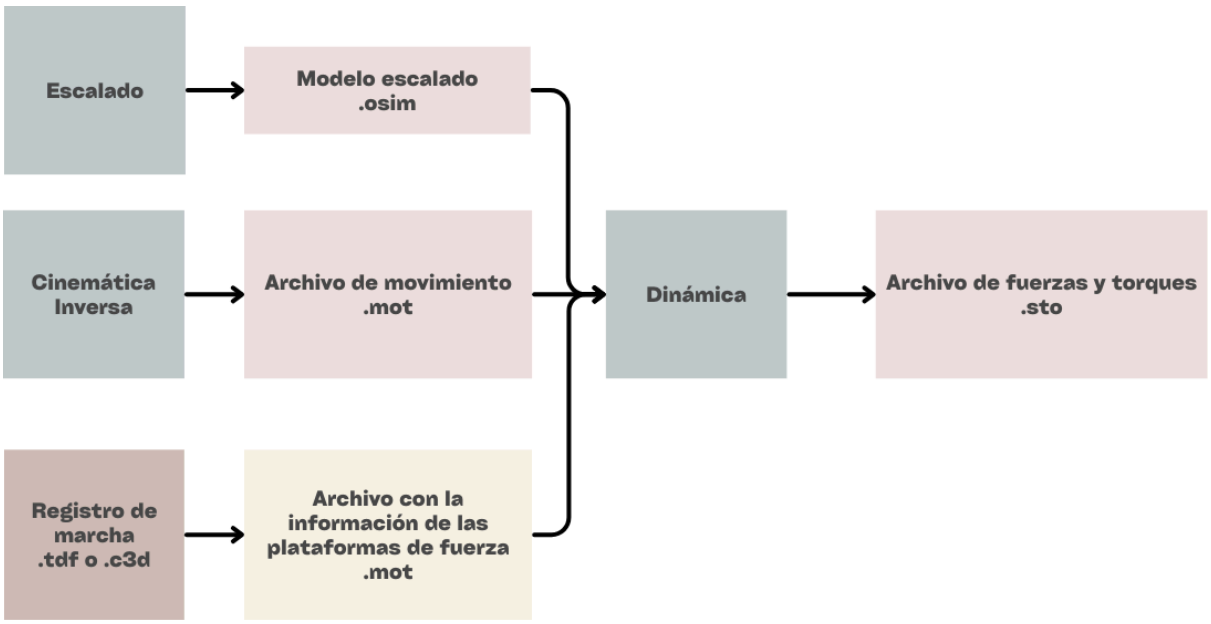


Fig. 58 Archivos de entrada y salida de la herramienta de dinámica inversa de OpenSim

5.1. Archivo de configuración dinámica inversa

Para ejecutar la herramienta de dinámica inversa es necesario tener un archivo de configuración con una estructura similar a la que se muestra en la **Fig. 59.**

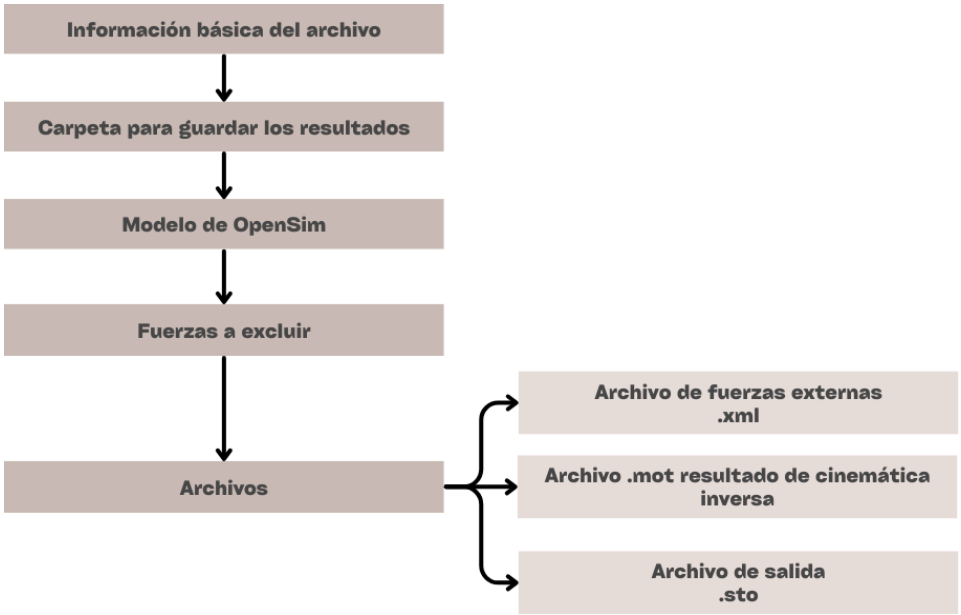


Fig. 59 Archivos de entrada y salida de la herramienta de dinámica inversa de OpenSim

Para obtener este archivo realice una copia del archivo *ssubject01_Setup_InverseDynamics.xml*, que se encuentra en la carpeta de

Gait2354_Simbody de los modelos de OpenSim, esto con el fin de no tener que crear el archivo de configuración desde cero, se recomienda cambiar parcialmente el nombre del archivo, reemplazando *subject01* por el nombre o código de su sujeto de prueba. Posteriormente abra el archivo con un editor de texto o código.

Coloque el nombre de la carpeta donde desea almacenar los resultados del estudio de dinámica inversa, entre las etiquetas *results_directory*, además, agregue el nombre del archivo .osim que obtuvo como salida del proceso de escalado y ubicación de los marcadores en la pose estática entre las etiquetas *model_file,* como se observa en la **Fig. 60.** Antes de seleccionar el tiempo de simulación verifique que este sea exáctamente el inicio y el fin del registro de la plataforma de fuerza, es decir, donde los datos son diferentes de cero, dado que, si se contemplan tiempos en los que no hay datos sobre el vector de fuerza de reacción, la simulación será incorrecta, para esto revise la columna de tiempo y de datos del archivo .mot, **Fig. 61**. En la sección de *forces_to_exclude* indique si desea excluir las fuerzas musculares, los actuadores externos o todo, esto con el fin de desarrollar el estudio solo con las fuerzas que se cargaran de los datos obtenidos de la plataforma de fuerza, estos datos se cargan en un archivo de configuración que se agrega entre las etiquetas, *external_loads_file*, y cuyo proceso de creación se explica posteriormente. Finalmente modifique tanto el nombre del archivo que está entre las etiquetas, c*oordinates_file,* para que coincida con el nombre del archivo .mot que obtuvo del estudio de cinemática inversa, así como el nombre del archivo entre las etiquetas, *output_gen_force_file,* el cual corresponde a un archivo de almacenamiento con los resultados de las fuerzas netas y los torques calculados para cada articulación durante el estudio **Fig. 60**.

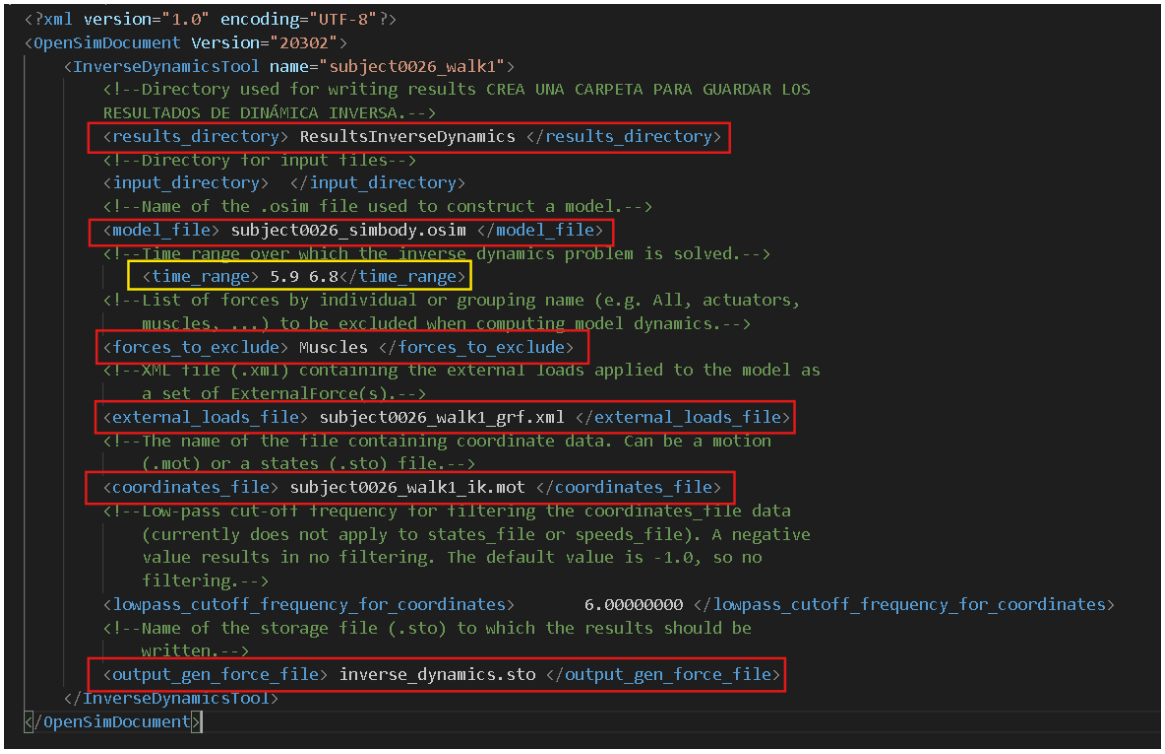


Fig. 60 Parámetros a editar en el archivo de dinámica inversa

walkingC3Dtools.mot

```
0026~aa~Walking 01_79a.mot
version = 1
nRows= 729
nColumns= 10
endheader
time    ground_force_vx ground_force_vy ground_force_vz ground_force_px ground_force_py ground_force_pz ground_torque_x ground
0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.016667    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.033333    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.050000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.066667    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.083333    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.100000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.116667    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.133333    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.150000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.166667    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
0.183333    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
```

```
5.933333    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
5.950000    11.717051   134.596025  -5.856692   0.611835    -0.000439   0.069975    18.941176   0.629569    42.351511
5.966667    -45.783277  400.150244  -11.423564  0.558513    -0.000287   0.086835    49.541485   12.620631   104.521199
5.983333    -83.795151  549.821024  -26.429328  0.574004    -0.000322   0.085031    69.061983   21.749502   152.129164
6.000000    -80.628392  595.754635  -21.547274  0.576480    -0.000327   0.084992    74.855107   18.519058   166.313255
6.016667    -83.856097  597.300474  -21.587856  0.579219    -0.000330   0.086084    74.396150   18.241739   168.378678
6.033333    -87.567553  577.669115  -25.938836  0.581585    -0.000331   0.087032    71.404138   19.449458   164.211680
6.050000    -79.934611  534.534863  -33.913405  0.582281    -0.000329   0.088218    65.442199   20.103847   152.330636
```

Fig. 61 Archivo de la plataforma de fuerza (.mot)

5.1.1. Archivo de fuerzas externas

Para crear el archivo que va entre las etiquetas, *coordinates_file,* realice una copia del archivo *subject01_walk1_grf.xml*, que se encuentra en la carpeta de *Gait2354_Simbody* de los modelos de OpenSim, este archivo tiene una estructura similar a la que se muestra en la **Fig. 62.**

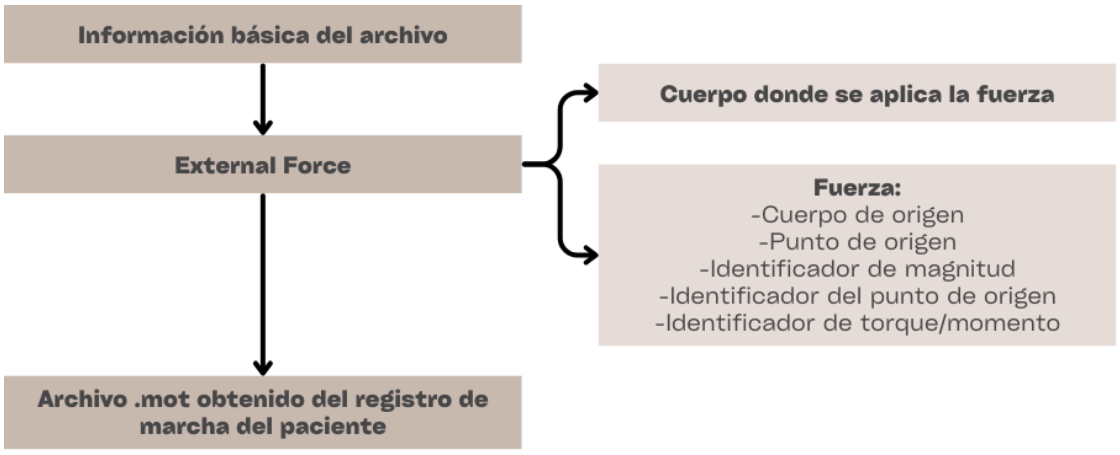
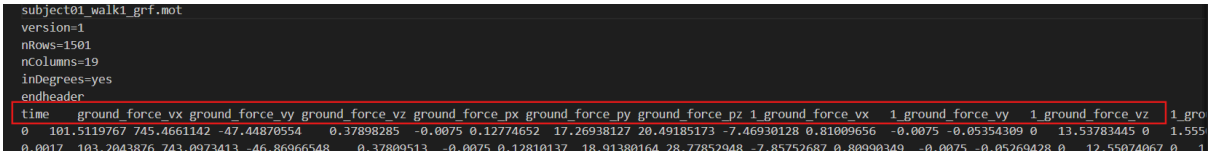


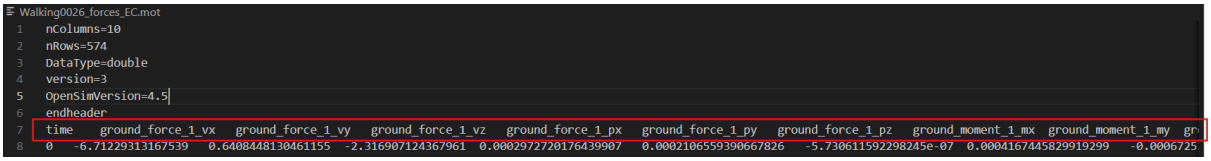
Fig. 62 Estructura del archivo de configuración de fuerzas externas

Para conocer los identificadores de magnitud, punto de origen y momento o torque, es necesario abrir el archivo .mot que se obtuvo de los registros de marcha del

sujeto, este debe de tener una estructura similar a la que se presenta en la **Fig. 63,** donde se marca en rojo la sección con los identificadores de magnitud, punto de origen y momento o torque. La **Fig. 63a** proviene de un archivo de registro de marcha que contiene los identificadores de dos plataformas de fuerza, cuyos nombres están en la **TABLA II.** Hay seis componentes para magnitud, punto de origen y torque, dado que cada plataforma debe registrar el vector en tres dimensiones y son dos plataformas. Adicionalmente, también están los identificadores de un segundo archivo con una sola plataforma de fuerza como se muestra en la **Fig. 63b** y en la **TABLA II**, queda claro que, aunque los nombres son similares, no son idénticos y por tanto es necesario que distinga cuáles son los nombres de su archivo y qué parte corresponde al identificador, el cual está marcado en azul en la **TABLA II,** el identificador excluye la descripción de la componente XYZ, dado que de eso se encarga OpenSim.



a



b

Fig. 63 a) Archivo de una plataforma de fuerza (.mot). **b)** Archivo con dos plataformas de fuerza (.mot)

Nota. En rojo se marcan los indicadores de magnitud, posición y toque de las plataformas de fuerza de cada archivo

TABLA II

Identificación de los indicadores de magnitud, fuerza y torque de las plataformas de fuerza

Identificadores Fig. 63a	**Identificadores Fig. 63b**	**Correspondencia**
ground_force_vx ground_force_vy ground_force_vz 1_ground_force_vx 1_ground_force_vy 1_ground_force_vz	ground_force_1_vx ground_force_1_vy ground_force_1_vz	Componentes de la magnitud del vector de fuerza

ground_force_px ground_force_py ground_force_pz 1_ground_force_px 1_ground_force_py 1_ground_force_pz	ground_force_1_px ground_force_1_py ground_force_1_pz	Punto de origen del vector de fuerza
ground_torque_x ground_torque_y ground_torque_z 1_ground_torque_x 1_ground_torque_y 1_ground_torque_z	ground_moment_1_mx ground_moment_1_my ground_moment_1_mz	Momento del vector

Nota. En azul se marca la parte del identificador que debe colocarse en el archivo de fuerzas externas, se descarga la especificación de componentes XYZ.

Una vez tenga definidos cuáles son sus identificadores puede proceder a editar el archivo de fuerzas externas que copió previamente. Inicialmente debe tener una sección de *ExternalForce* (**Fig. 64**), para cada plataforma de fuerza, por lo que según su estudio considere añadir o eliminar secciones, es decir, si usa una única plataforma de fuerza entonces elimine uno de los bloques de *ExternalsForce*, en este caso, *right* o *left*, luego renombre el bloque restante con el lado de la extremidad sobre la que se aplica la fuerza. Por el contrario, si cuenta con más de dos plataformas entonces copie y pegue un bloque de *ExternalForce* para agregar una fuerza sobre otra extremidad.

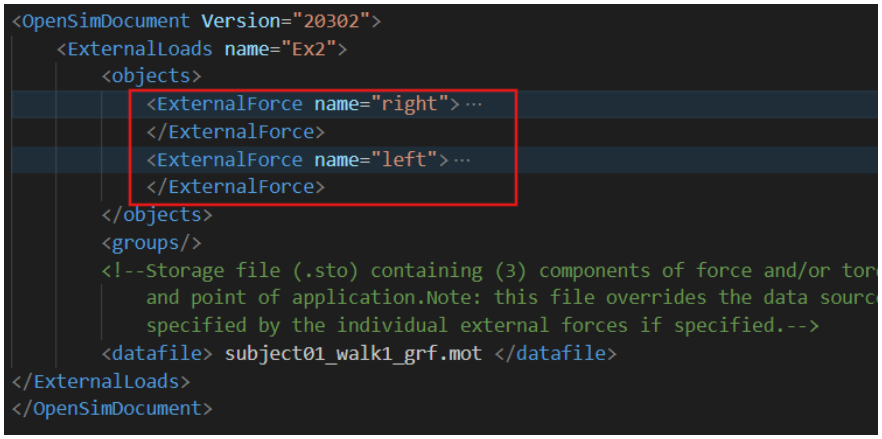


Fig. 64 Secciones de *ExternalForce* a editar

Dentro de esta sección de *ExternalForce* se debe describir el cuerpo al que se le aplicará la fuerza, cuyo nombre debe coincidir con el nombre de este cuerpo en el archivo del modelo de OpenSim, en este caso se aplica sobre el calcáneo derecho como se ve en la

Fig. 65, por lo que modifíquelo por el nombre del cuerpo de su modelo. Posteriormente, entre las etiquetas *force_expressed_in_body* y *point_expressed_in_body* se describe el cuerpo y el punto desde el que surge la fuerza, como son plataformas de fuerza entonces esta se origina desde el piso. Las siguientes etiquetas corresponden a los identificadores de fuerza (magnitud), punto de origen y torque, cambie los nombres a los que corresponden con su archivo. Si revisa con detenimiento la **Fig. 65**, se dará cuenta de que esta contiene los identificadores del archivo presentado en la **Fig. 63a** sin incluir las componentes XYZ como se describió previamente. Finalmente, en las etiquetas *datafile* escriba el nombre del archivo .mot del registro de marcha del que extrajo los nombres de los identificadores.

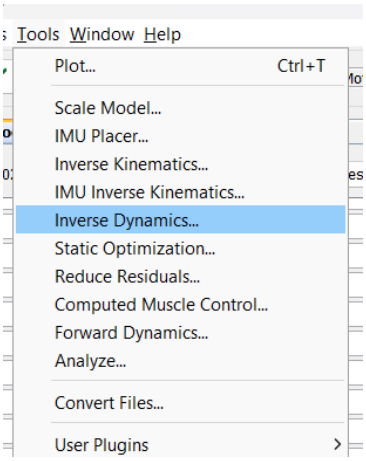
```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSimDocument Version="20302">
    <ExternalLoads name="Ex2">
        <objects>
            <ExternalForce name="right">
                <isDisabled> false </isDisabled>
                <!--Name of the body the force is applied to.-->
                <applied_to_body> calcn_r </applied_to_body>
                <!--Name of the body the force is expressed in (default is ground).-->
                <force_expressed_in_body> ground </force_expressed_in_body>
                <!--Name of the body the point is expressed in (default is ground).-->
                <point_expressed_in_body> ground </point_expressed_in_body>
                <!--Identifier (string) to locate the force to be applied in the data
                    source.-->
                <force_identifier> ground_force_v </force_identifier>
                <!--Identifier (string) to locate the point to be applied in the data
                    source.-->
                <point_identifier> ground_force_p </point_identifier>
                <!--Identifier (string) to locate the torque to be applied in the data
                    source.-->
                <torque_identifier> ground_torque_ </torque_identifier>
                <!--Name of the data source (Storage) that will supply the force data.-->
                <data_source_name> Unassigned </data_source_name>
            </ExternalForce>
            <ExternalForce name="left">···
            </ExternalForce>
        </objects>
        <groups/>
        <!--Storage file (.sto) containing (3) components of force and/or torque
            and point of application.Note: this file overrides the data source
            specified by the individual external forces if specified.-->
        <datafile> subject01_walk1_grf.mot </datafile>
    </ExternalLoads>
</OpenSimDocument>
```

Fig. 65 Parámetros a editar en el archivo de fuerzas externas

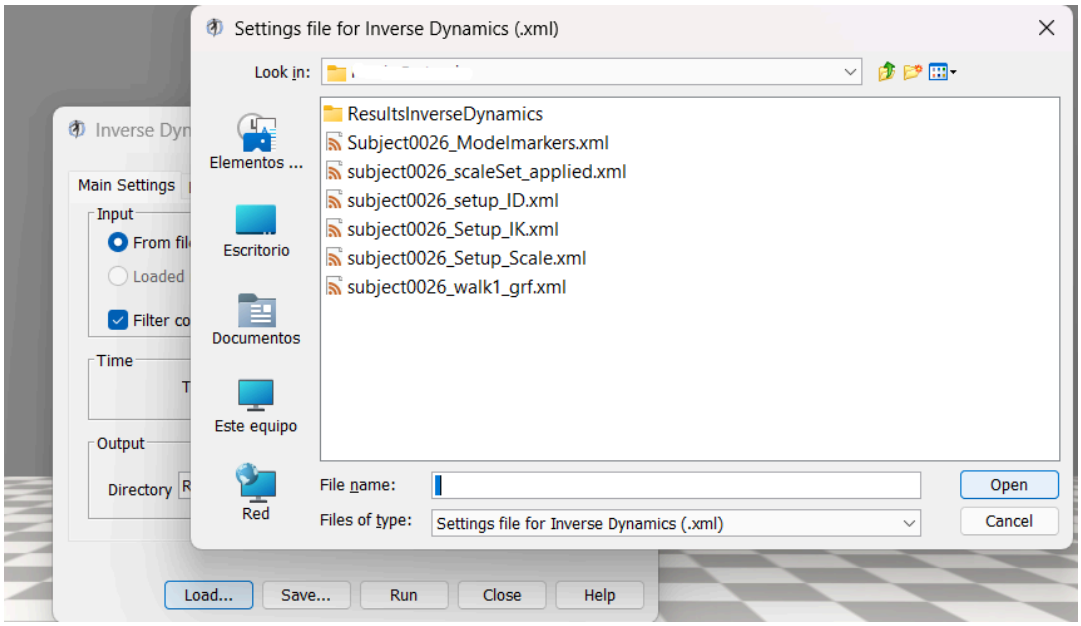
5.2. Ejecutar la herramiento de dinámica inversa

Una vez tenga listo tanto el archivo de configuración como el archivo de cargas externas, basta con ir a OpenSim, abrir el archivo del modelo escalado y con los marcadores reubicados, en caso de que no se encuentre abierto, luego, presionar *Tools* → *Inverse Dynamics* → *Load*, esto abrirá su explorador de archivos y podrá cargar el archivo de configuración creado, **Fig. 66,** verifique en la pestaña

External loads de la ventana emergente que se haya cargado el archivo creado y posteriormente presione *run*.



a



b

Fig. 66 Ejecutar la herramienta de dinámica inversa

5.3. Evaluar el resultado de dinámica inversa

Para comprobar que la simulación se realizó correctamente comience por cargar los datos de la plataforma de fuerza dando clic derecho sobre *IKResults* en la ventana de *Navigator*, seleccione la opción de *Associate Motion Data…* y cargue el archivo .mot del registro de marcha, como se ve en la **Fig. 67,** para ver el movimiento del modelo active la opción de bucle de la barra superior, como lo hizo en cinemática inversa.

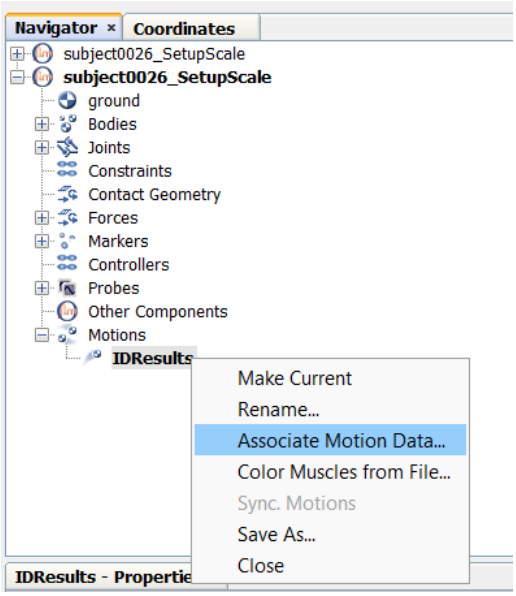


Fig. 67 Cargar el archivo para evaluar la calidad de dinámica inversa

Verifique que el vector de fuerza de reacción del suelo esté orientado correctamente, es decir, la punta de la flecha debe apuntar hacia el eje vertical del cuerpo, y su magnitud debe cambiar conforme el modelo se desplaza, **Fig. 68**.

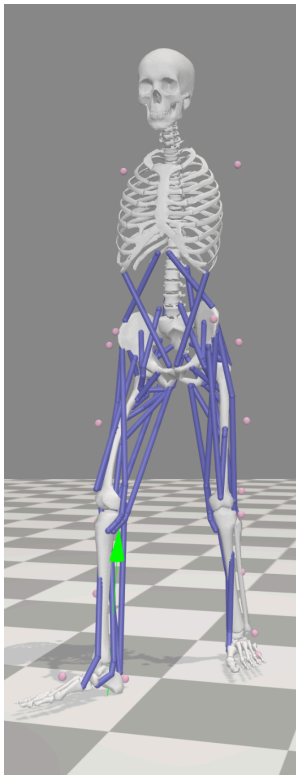
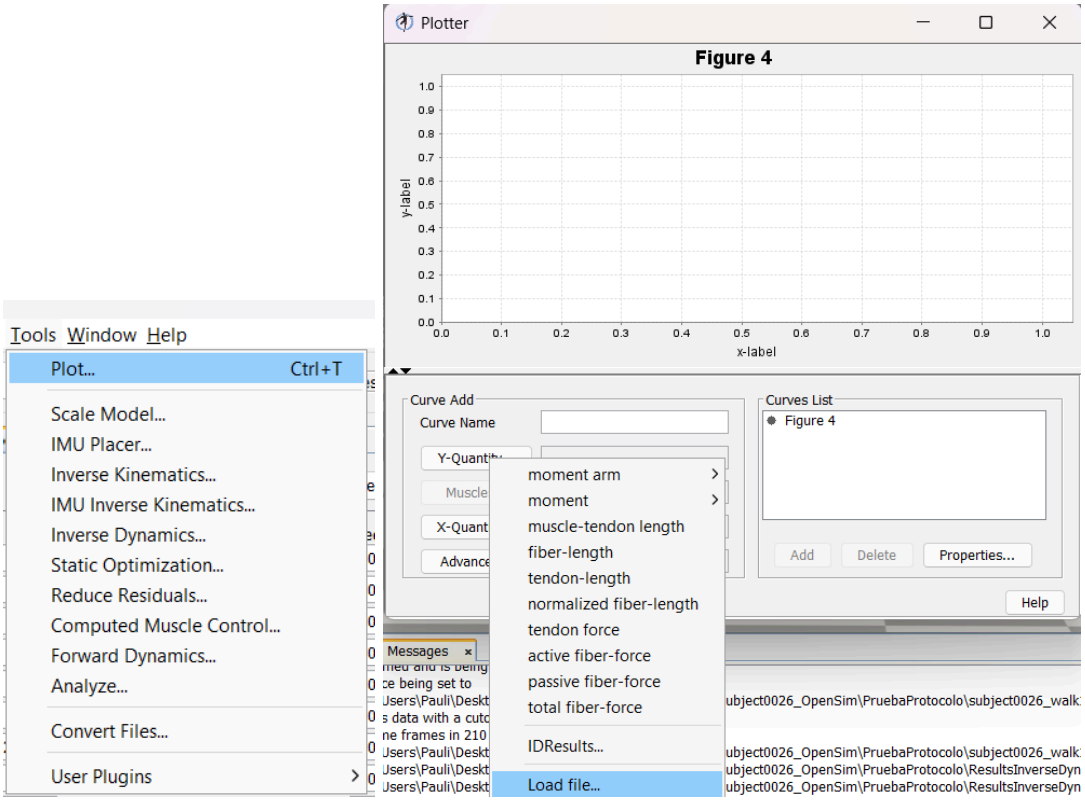
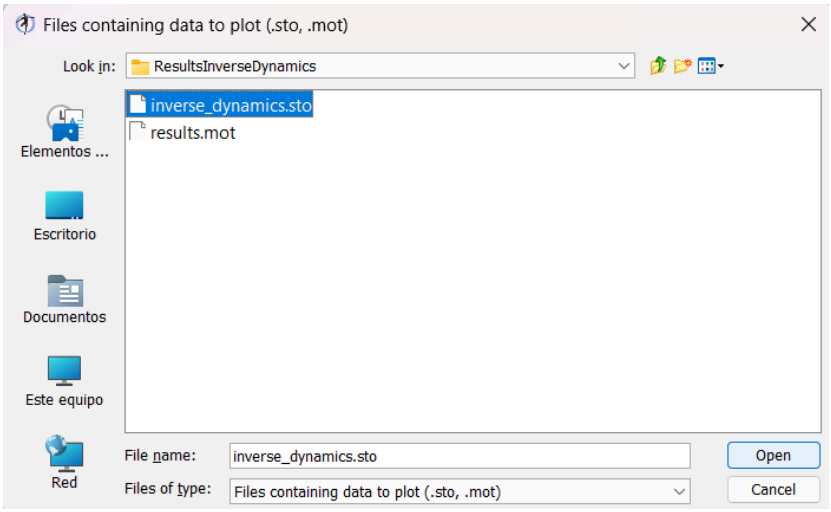
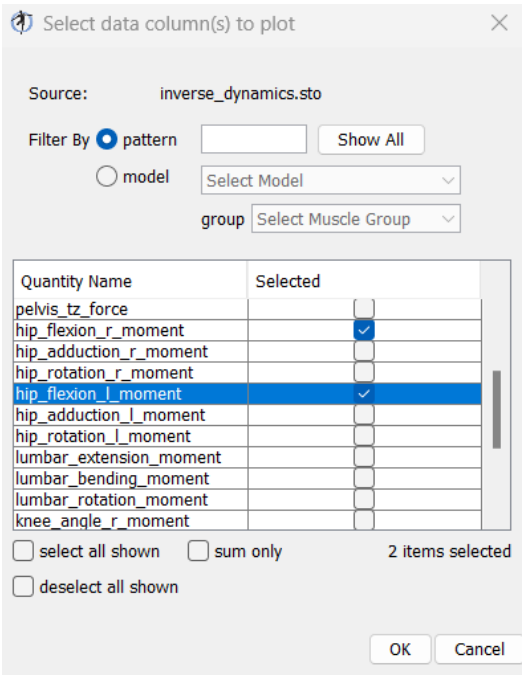


Fig. 68. Visualización del vector de fuerza de reacción del suelo.

En este punto también es útil graficar los momentos o fuerzas que se almacenaron en el archivo de resultados (.sto), para ello haga uso de la siguiente ruta: *Tools→Plot,* en la ventana emergente presione *Y-Quantity→Load File,* esto abrirá el buscador de archivos de su dispositivo, seleccione el archivo .sto que está dentro de la carpeta de resultados de dinámica inversa de su simulación, en la ventana emergente verá una lista con todas las opciones de momentos y fuerzas que puede graficar, seleccione las que necesite según su estudio. Una vez tenga el eje Y del gráfico definido, puede seleccionar el eje X, para esto de clic sobre *X-Quantity* y escoja *time* en la ventana emergente, luego presione *Ok* y en la otra ventana presione *Add*. En la **Fig. 69**, se presenta una versión gráfica de la descripción anterior.




Files containing data to plot (.sto, .mot)
Look in:
ResultsInverseDynamics
inverse_dynamics.sto
results.mot
Elementos ...
Escritorio
Documentos
Este equipo
Red
File name:
inverse_dynamics.sto
Open
Files of type:
Files containing data to plot (.sto, .mot)
Cancel


Select data column(s) to plot
Source:
inverse_dynamics.sto
Filter By
pattern
Show All
model
Select Model
group
Select Muscle Group
Quantity Name
Selected
pelvis_tz_force
hip_flexion_r_moment
hip_adduction_r_moment
hip_rotation_r_moment
hip_flexion_l_moment
hip_adduction_l_moment
hip_rotation_l_moment
lumbar_extension_moment
lumbar_bending_moment
lumbar_rotation_moment
knee_angle_r_moment
select all shown
sum only
2 items selected
deselect all shown
OK
Cancel

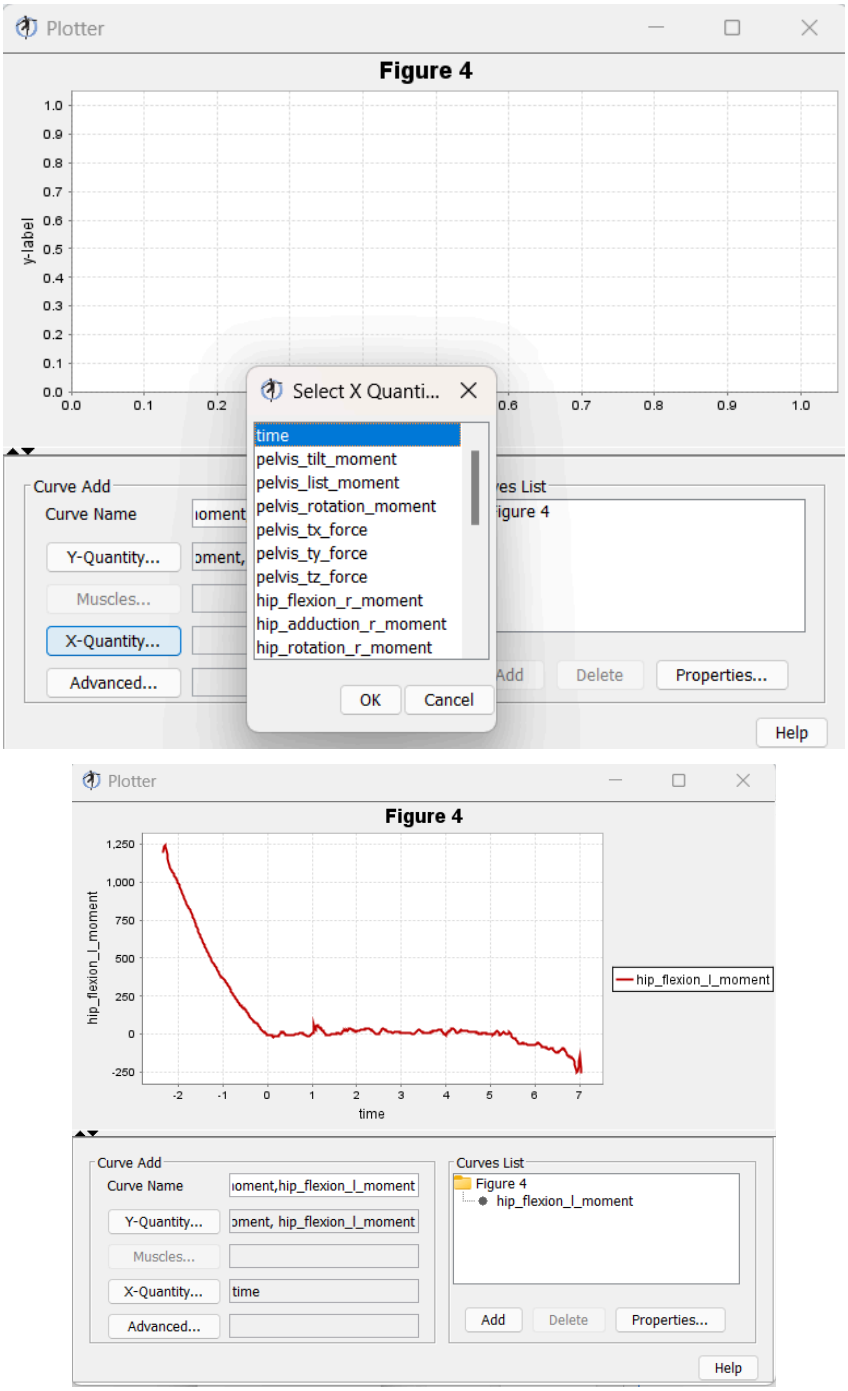


Fig. 69. Herramienta de gráficos de OpenSim

6. Algoritmo de reducción residual

OpenSim cuenta con una herramienta denominada algoritmo de reducción residual (RRA por sus siglas en inglés), el cual es un tipo de dinámica inversa que mejora la precisión de la simulación al reducir los residuos dinámicos de fuerzas y momentos no fisiológicos, es decir, busca ajustar el centro de masa del torso permitiendo variar la cinemática del

modelo a partir de los datos de cinemática inversa, para que este sea dinámicamente coherente con los datos de fuerza de reacción del suelo [8].

Para llevarlo a cabo se requieren una serie de archivos como se muestra en la **Fig. 70**, los cuales contienen información relevante para realizar el estudio, por lo que se describirá cómo construir cada uno.

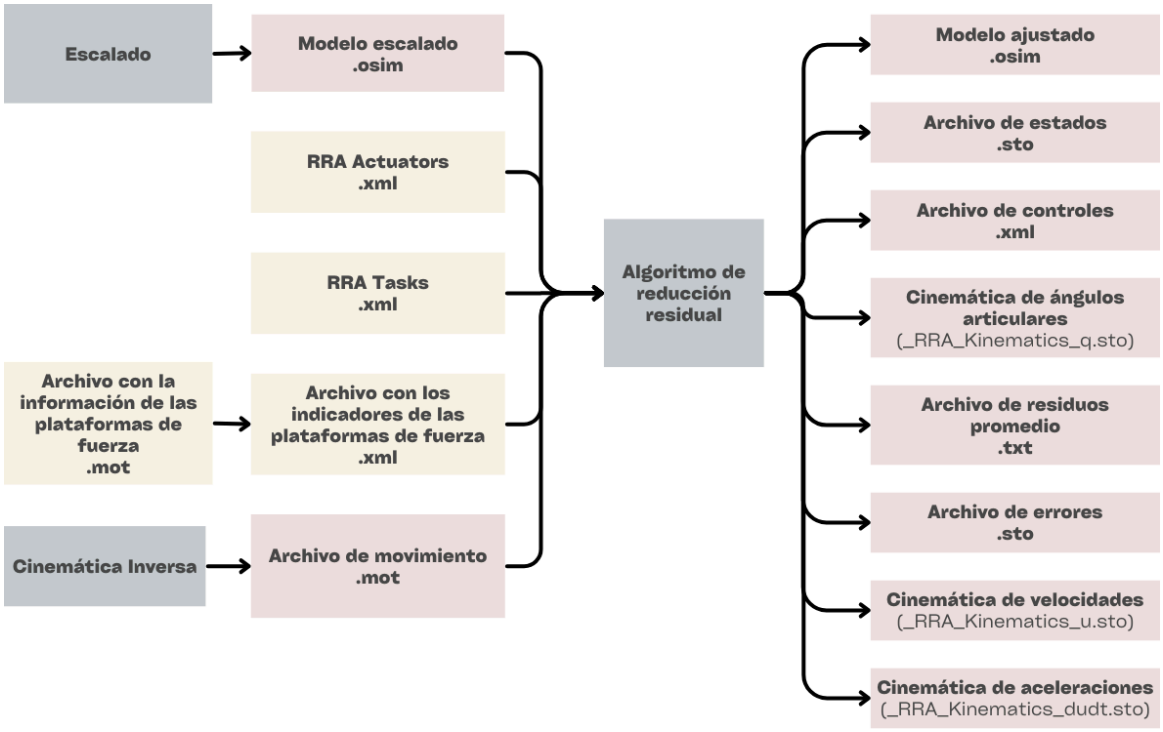


Fig. 70 Archivos de entrada y salida de la herramienta de algoritmo de reducción residual de OpenSim

6.1. Archivo de configuración de RRA

Para emplear la herramienta de RRA de OpenSim es necesario realizar un archivo de configuración. Como se ha hecho tanto para cinemática como dinámica inversa, este archivo también se puede encontrar en la carpeta de *Gait2354* como *subject01_Setup_RRA.xml*, por lo que se recomienda realizar una copia para editarlo, su contenido tiene una estructura similar a la que se muestra en la **Fig. 71**, este recibe seis archivos de entrada y produce ocho archivos de salida:

- Nuevo modelo ajustado.
- Tres archivos de cinemática, uno con las componentes de los ángulos articulares, otro con las velocidades y otro con las aceleraciones.

- Archivo con los errores de posición de las coordenadas generados durante el estudio.
- Archivo con la cinemática ajustada y los estados del modelo correspondientes al movimiento simulado.
- Archivo con los valores residuales promedio.
- Archivo con los controles de las excitaciones de los actuadores.

La configuración del archivo inicial es sencilla siempre y cuando se mantengan los valores empleados en el modelo de *Gait2354*, sin embargo, estos también se pueden modificar en caso de que su estudio lo requiera. Adicionalmente, es necesario realizar una copia de otros dos archivos que configuran RRA, lo cuales se encuentran en la misma carpeta, *gait2354_RRA_Tasks.xml* y *gait2354_RRA_Actuators.xml.*

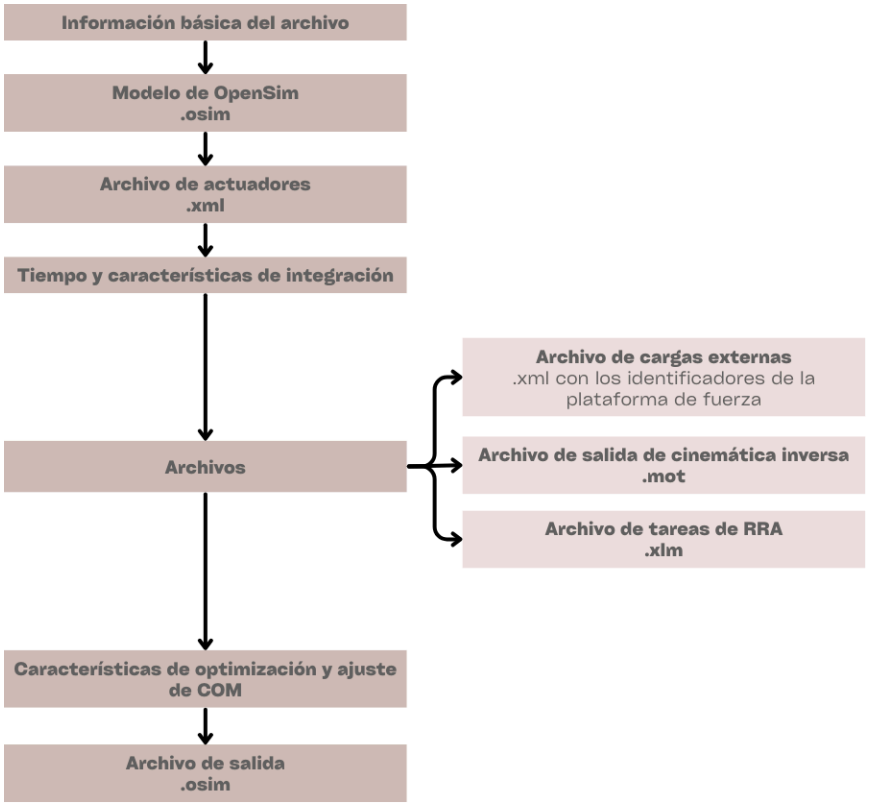


Fig. 71 Estructura del archivo de configuración de la herramienta de algoritmo de reducción residual de OpenSim

Las modificaciones generales de este archivo consisten en reemplazar los nombres de los archivos de entrada y de salida por el nombre con el que se haya guardado cada uno.

En la **Fig. 72** se presentan los primeros cambios que se deben realizar en el archivo, modifique el nombre entre las etiquetas <*model_file*> para que corresponda con su modelo de OpenSim en el que ejecutó cinemática inversa, así mismo, reemplace el nombre del archivo <*force_set_file*> por el nombre con que guardó el archivo de actuadores copiado de *Gait2354*, cuya edición se explicará más adelante. Por otro lado, puede o no modificar el nombre de la carpeta en la que se guardarán los resultados de este estudio. En esta misma figura se marca en amarillo las características de solución de las integrales de RRA, primero se debe determinar un valor de precisión, así como el rango de tiempo del análisis, el valor de precisión dependerá de qué tan importante es la dinámica para su estudio, mientras mayor sea el valor más preciso será el modelo. Ahora bien, los valores de máximo número de pasos de integración, así como el valor máximo y mínimo del paso de integración y la tolerancia deben estar balanceados entre ellos, por lo que es importante comprender a qué hacen referencia y cómo se relacionan. El valor máximo de pasos de integración indica la cantidad total de integraciones antes de finalizar RRA, mientras que el máximo paso de integración indica la cantidad máxima de tiempo durante el cual se espera la solución de cada integral, por lo que a menor tamaño mayor deberá ser la tolerancia al error.

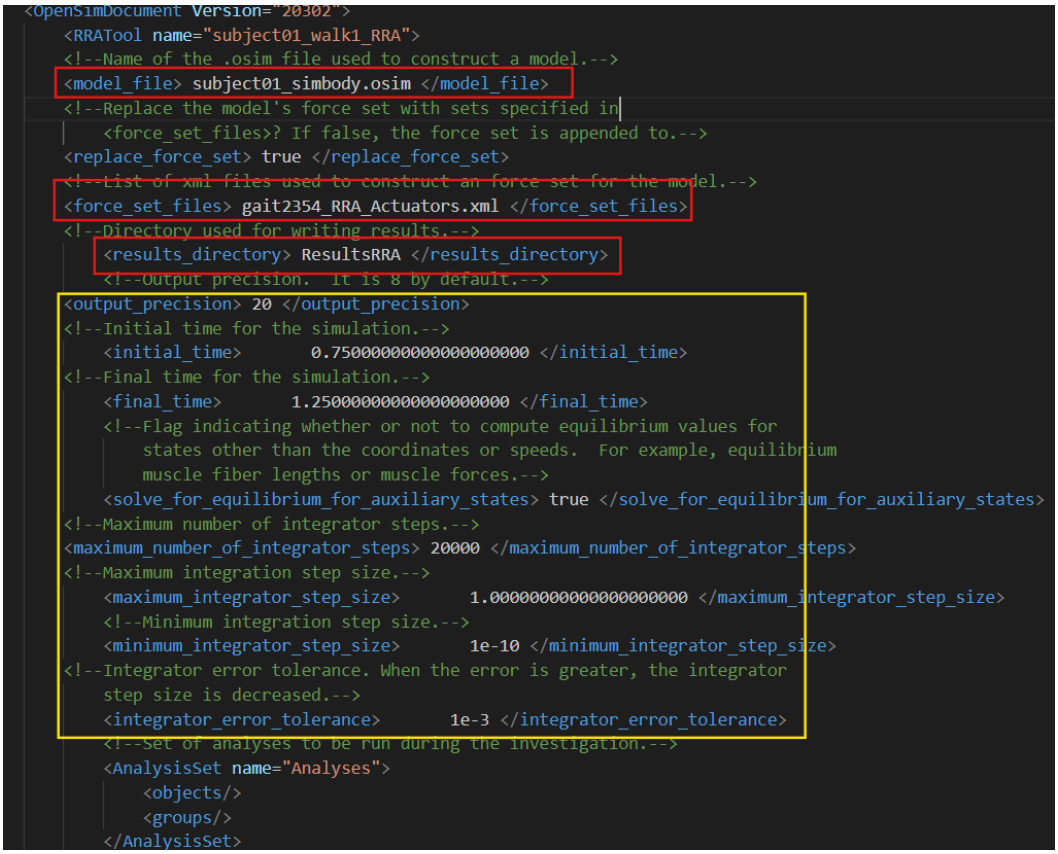


Fig. 72 Parámetros a editar en el archivo de configuración de RRA

Entre las etiquetas <*external_loads_file*> reemplace el nombre del archivo por el archivo que contiene los indicadores y el cuerpo de aplicación del registro de la plataforma de fuerza, usado

en dinámica inversa, mientras que en *<desired_kinematics_file>* reemplace el nombre por el archivo .mot que obtuvo como resultado de cinemática inversa. Entre las etiquetas *<task_set_file>* reemplace el nombre por el archivo de tareas que copió de la carpeta *Gait2354,* como se muestra marcado en la **Fig. 73**. Elimine tanto el contenido como las etiquetas de *<constraints_file>* dado que ya se desaconseja el uso de este archivo.

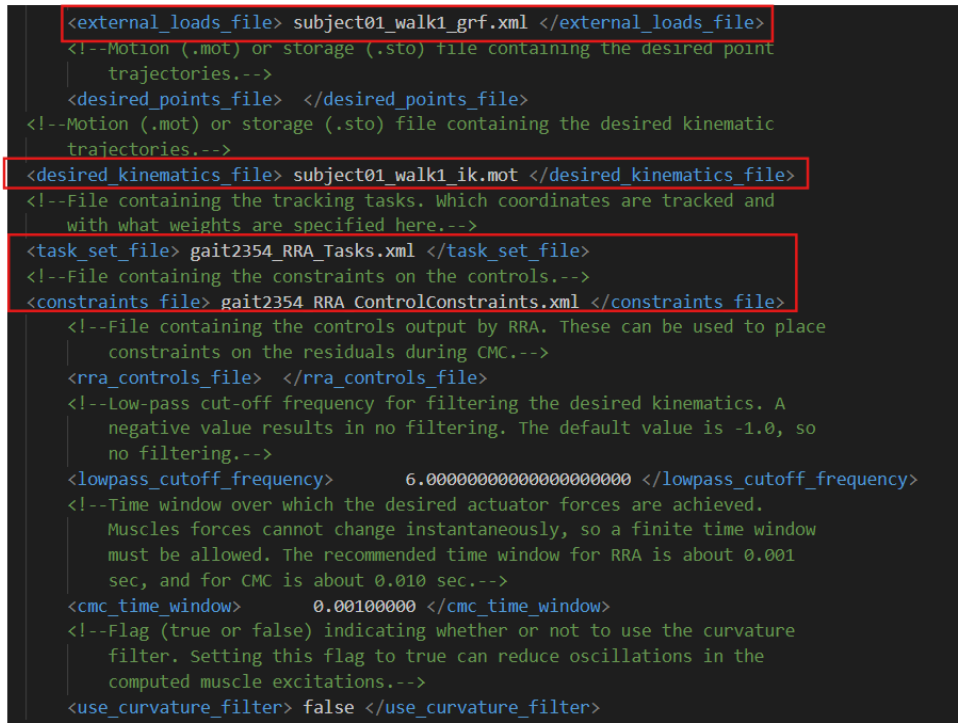


Fig. 73 Parámetros a editar en el archivo de configuración de RRA

Finalmente cambie el nombre del nuevo modelo de salida entre las etiquetas *<output_model_file>* por uno que corresponda con su sujeto de estudio.

6.1.1. RRA Tasks

Este archivo especifica qué coordenadas son importantes y con qué precisión deben rastrearse lo que guía a la herramienta de RRA para ajustar el movimiento del modelo y minimizar los residuos dinámicos. El archivo tiene una sección de *CMC_Joint* para cada movimiento del modelo, al igual que el archivo de *ControlConstraints*, como se muestra en la **Fig. 74,** por lo que se debe revisar que los nombres coincidan con los del modelo que está trabajando, ahora bien, respecto a los valores dentro del resto de etiquetas, su modificación depende de los objetivos propios de cada estudio, por lo que es necesario identificar a qué hace referencia cada una.

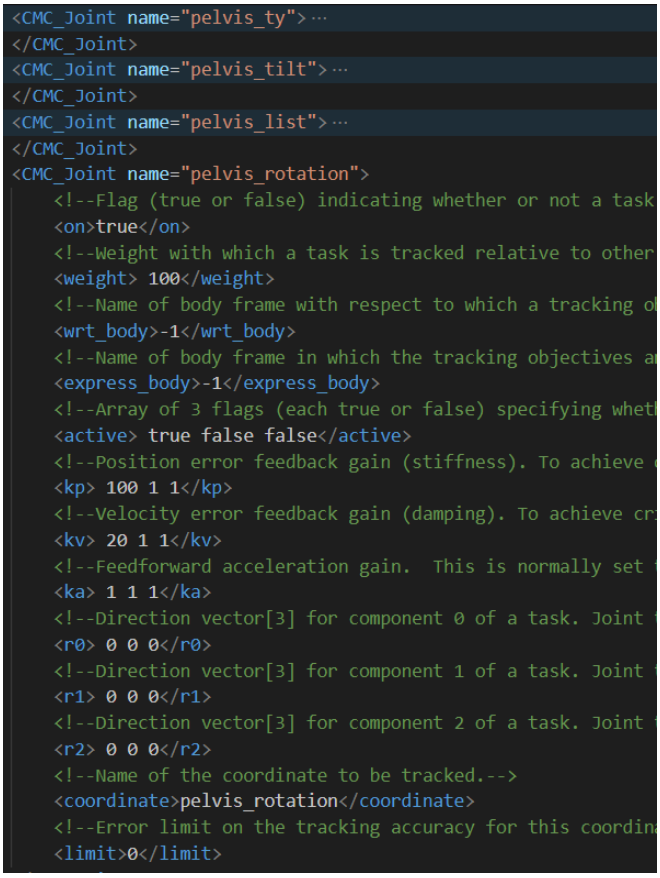


Fig. 74 Parámetros a editar en el archivo de tareas de RRA

- **on:** Dependiendo de si su valor es *true* o *false*, la tarea estará habilitada o deshabilitada respectivamente
- **weight:** Indica el peso con el que se realizará la tarea respecto a otra, a mayor peso mayor será la prioridad y la precisión con que se ejecute esa tarea
- **wrt_body y express_body:** Estas propiedades se usan para definir la posición de un punto y su relación con otros marcos en el modelo. Son relevantes para tareas que implican el seguimiento de trayectorias de puntos en el espacio, pero no se aplican al seguimiento de coordenadas articulares.
- **Active:** Indica qué coordenadas están activas o inactivas, es decir, *true false false* indica que la coordenada X está activa mientras que Y y Z están inactivas, esto permite seguir la trayectoria del punto a través de las tres componentes, sin embargo, para articulaciones solo se activa la primera componente.
- **kp, kv y ka:** Corresponden a los parámetros de la ley de control proporcional derivativo, con la cual se calculan las aceleraciones de la cinemática. Esta ley tiene tanto ganancia de retroalimentación rigidez, como ganancia de amortiguación y ganancia de avance de aceleración las cuales se especifican para cada coordenada dentro de kp, kv y ka respectivamente.

- **r0, r1, r2:** indican vectores de dirección para los tres componentes de una tarea, sin embargo, no se utilizan para tareas de seguimiento de una única coordenada de articulación.

6.1.2. RRA Actuators

En este archivo se definen los actuadores que usará el modelo durante el análisis de RRA, existen tres tipos como se muestra en la **Fig. 75**, punto, torque y coordenada, en cada uno se especifica la fuerza del actuador y los valores mínimos y máximos que puede aplicar, además de algunas características propias que dependen de cada tipo.

```
<PointActuator name="FZ">
        <isDisabled> false </isDisabled>
        <!--Minimum allowed value for control signal. Used primarily when solving…
        <min_control> -infinity </min_control>
        <!--Maximum allowed value for control signal. Used primarily when solving…
        <max_control> infinity </max_control>
  <body> pelvis </body>
        <point>      -0.07243760000000000500       0.00000000000000000000       0.00000000000000000000 </point>
  <point_is_global> false </point_is_global>
        <direction>      -0.00000000000000000000      -0.00000000000000000000       1.00000000000000000000 </direction>
  <force_is_global> true </force_is_global>
        <optimal_force>       4.00000000000000000000 </optimal_force>
</PointActuator>
<TorqueActuator name="MX">
        <isDisabled> false </isDisabled>
        <!--Minimum allowed value for control signal. Used primarily when solving…
        <min_control> -infinity </min_control>
        <!--Maximum allowed value for control signal. Used primarily when solving…
        <max_control> infinity </max_control>
    <bodyA> pelvis </bodyA>
    <bodyB> ground </bodyB>
    <torque_is_global> true </torque_is_global>…
</TorqueActuator>
<TorqueActuator name="MY">…
</TorqueActuator>
<TorqueActuator name="MZ">…
</TorqueActuator>
<CoordinateActuator name="hip_flexion_r">
        <isDisabled> false </isDisabled>
        <!--Minimum allowed value for control signal. Used primarily when solving…
        <min_control> -infinity </min_control>
        <!--Maximum allowed value for control signal. Used primarily when solving…
        <max_control> infinity </max_control>
    <coordinate> hip_flexion_r </coordinate>…
</CoordinateActuator>
```

Fig. 75 Parámetros a editar en el archivo de actuadores de RRA

- **PointActuator:** Además de los límites y la fuerza se debe especificar el cuerpo, el punto en el cuerpo y la orientación de este. Para ubicar el punto debe de buscar el centro de masa del cuerpo de ese actuador, en este caso sería la pelvis, para ello seleccione el cuerpo desplegando la pestaña *Body* en la ventana de *Navigator*, luego seleccione el cuerpo y vaya a la ventana de *Properties*, allí verá el apartado de *mass_center*, donde se indica el punto en el que está el centro de masa de ese objeto, como se muestra en la **Fig. 76**

- **TorqueActuator:** Como parámetros adicionales se especifican el cuerpo A y B, es decir el que ejerce el toque y el que lo recibe, además del eje.
- **CoordinateActuator:** Hay una sección de este tipo para cada movimiento del modelo como en los dos archivos de configuración previos, por lo tanto además de los parámetros de fuerza y límites, se debe agregar el nombre de la coordenada, es decir, del movimiento.

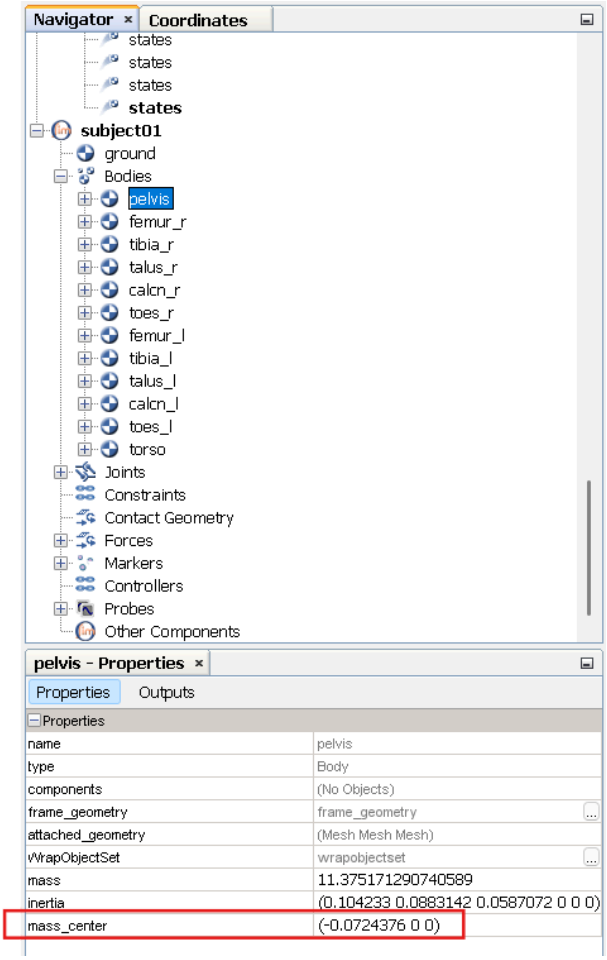
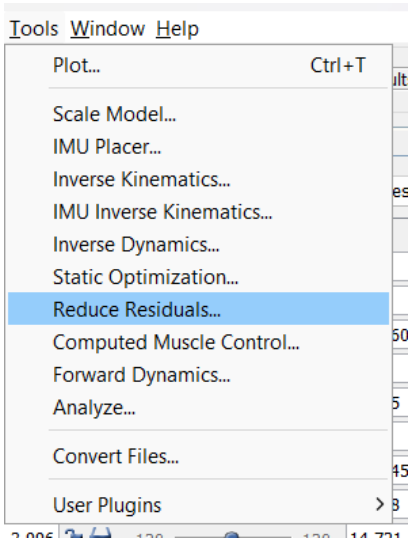


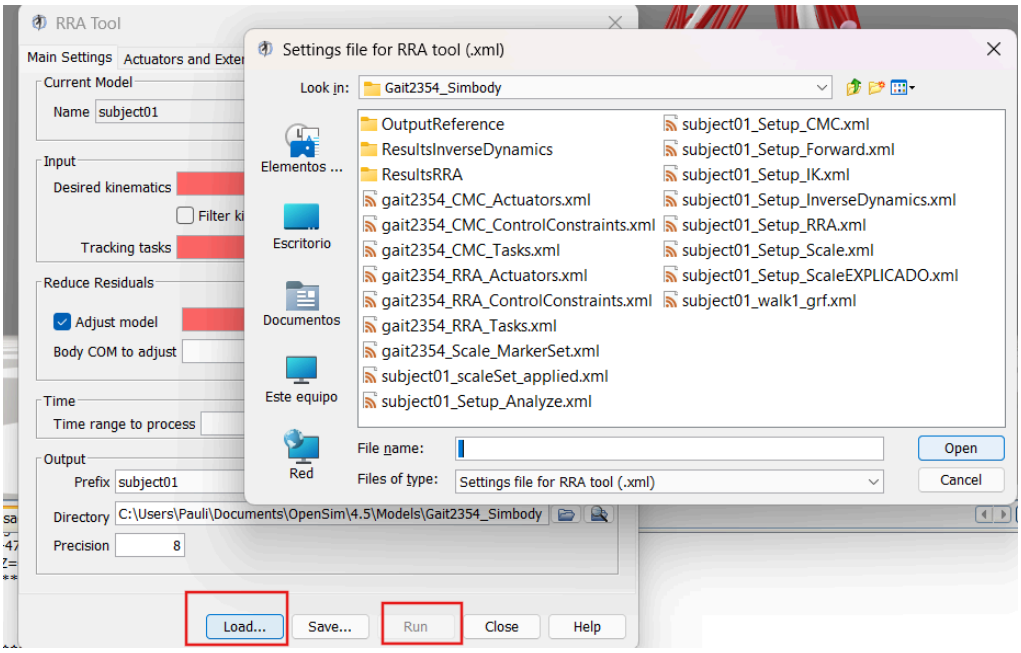
Fig. 76 Visualización del centro de masa de un cuerpo

6.2. Ejecutar la herramienta de RRA

Una vez se hayan editado adecuadamente los archivos de configuración de la herramienta de RRA es posible llevar a cabo el estudio, para esto es necesario seguir la siguiente ruta en OpenSim , como se muestra en la **Fig. 77,** *Tools → Reduce residuals…,* esto abrirá una ventana emergente en la que podrá cargar el archivo de configuración principal de RRA haciendo uso del botón *Load*, como se muestra en la **Fig. 77b,** una vez cargue el archivo verifique que toda la información se haya agregado correctamente y presione *Run.*



a



b

Fig. 77 Ejecutar la herramienta de RRA

6.3. Evaluar el resultado de RRA

Para evaluar la calidad de la simulación OpenSim recomienda que los valores de fuerza y momento residual se mantengan dentro de los límites especificados en la **TABLA III.** Para revisar estos parámetros es necesario desplegar la ventana *Messages* que se encuentra en la zona inferior, como se muestra en la **Fig. 78**, allí encontrará los momentos y fuerzas residuales de cada componente, así como la

reducción de la masa de cada cuerpo, por lo que se recomienda evaluar los resultados de esta ventana para asegurarse de que la simulación sea adecuada.

TABLA III
Criterios de evaluación de RRA

Parámetro	Bueno	Aceptable	Malo
Máxima fuerza residual (N)	0-10	10-25	> 25
RMS de la fuerza residual (N)	0-5	5-10	> 10
Máximo momento residual (Nm)	0-50	50-75	> 75
RMS del momento residual (Nm)	0-30	30-50	> 50
MAX pErr (trans, cm)	0-2	2-5	> 5
RMS pErr (trans, cm)	0-2	2-4	> 4
MAX pErr (rot, deg)	0-2	2-5	> 5
RMS pErr (rot, deg)	0-2	2-5	> 5

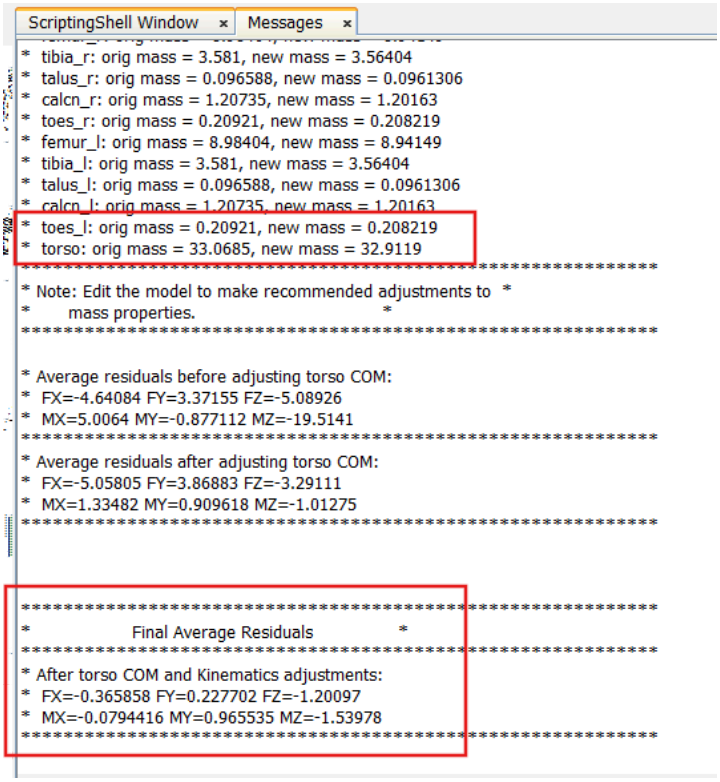


Fig. 78 Resultados de RRA

7. Optimización estática

Esta herramienta de OpenSim complementa la dinámica inversa del modelo al calcular las activaciones y fuerzas musculares necesarias para reproducir un movimiento específico, asegurando el equilibrio entre posiciones, velocidades, aceleraciones y fuerzas externas en cada instante de tiempo [9].

Como se muestra en la **Fig. 79** existen dos formas de ejecutarla, en la opción 1 se hace uso de un archivo de configuración desarrollado en RRA, pero no se requieren los resultados de esta simulación, mientras que en la opción 2 si se parte de los resultados cinemáticos para los ángulos de las articulaciones obtenidos con RRA, por lo que se explicará cómo llevar a cabo ambas alternativas.

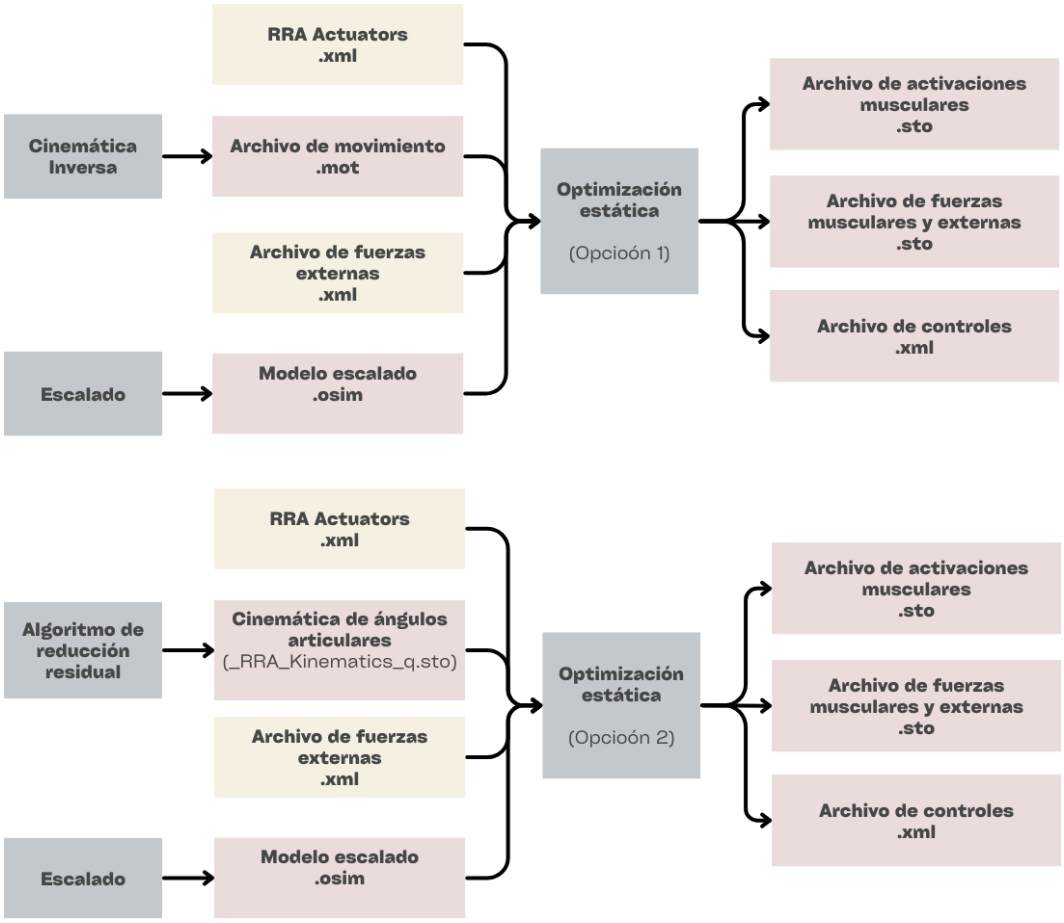


Fig. 79 Archivos de entrada y salida de la herramienta de optimización estática de OpenSim

7.1. Opción 1

Los archivos de entrada contemplan el modelo escalado con el que se va a trabajar, el archivo de movimiento que se obtuvo de emplear la herramienta de cinemática inversa (.mot), el archivo de fuerzas externas (.xml) que hace parte de la configuración de dinámica inversa (Contiene los indicadores de las plataformas de fuerza) y el archivo de actuadores que se configura para RRA (.xml). Dado que en las secciones anteriores se ha explicado cómo obtener cada uno de ellos, se recomienda revisar esta información en caso de no contar con alguno.

7.1.1. Ejecutar la herramienta de Optimización estática

Una vez tenga el modelo de OpenSim abierto presione *Tools* → *Static Optimization* ..., como se ve en la **Fig. 80.** En la ventana emergente cargue los archivos correspondientes, **Fig. 81**, en la sección de *Input* (Marcado en rojo) cargue el archivo de movimiento que obtuvo como resultado de cinemática inversa (.mot) y active la casilla de *Filter coordinates*, en *Time* (Marcado en verde) agregue el rango de tiempo que usó para ejecutar dinámica inversa, en la sección de *Output* (Marcado en azul) agregue al final de la dirección \ResultsSO, lo que creará una carpeta para guardar estos resultados. Ahora vaya a la pestaña de *Actuators and external loads* y en la sección de *External Loads* (Marcado en amarillo) active la casilla y agregue el archivo de las fuerzas externas que empleó en dinámica inversa (.xml), aquel que contiene los indicadores del vector de fuerza de reacción del suelo, finalmente presione *Run*.

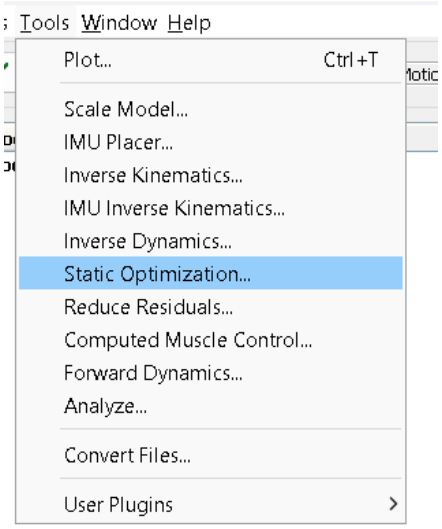
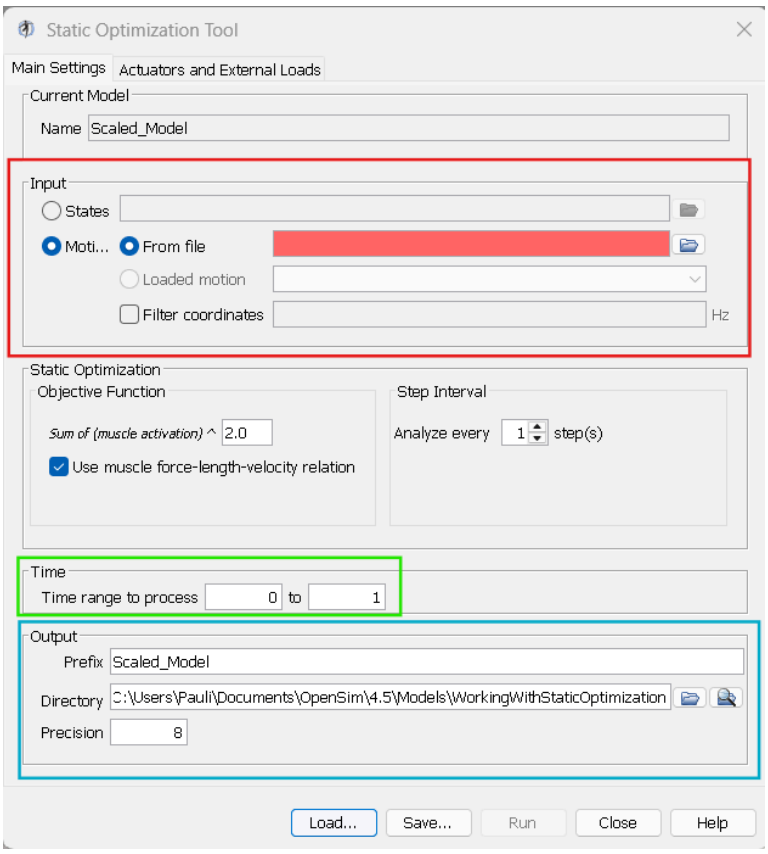
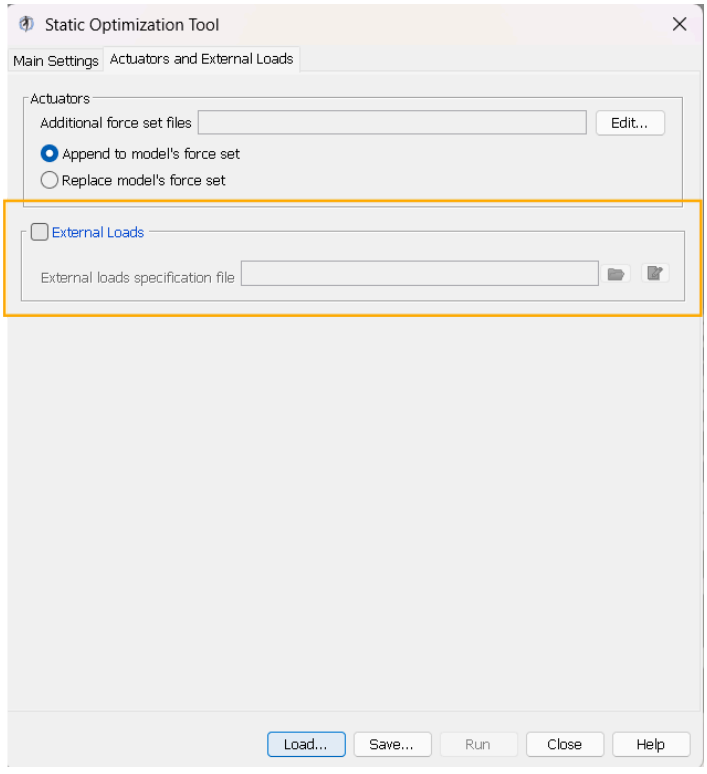


Fig. 80 Herramienta de optimización estática


Static Optimization Tool
Main Settings
Actuators and External Loads
Current Model
Name Scaled_Model
Input
States
Moti...
From file
Loaded motion
Filter coordinates
Hz
Static Optimization
Objective Function
Sum of (muscle activation) ^ 2.0
Use muscle force-length-velocity relation
Step Interval
Analyze every 1 step(s)
Time
Time range to process 0 to 1
Output
Prefix Scaled_Model
Directory C:\Users\Pauli\Documents\OpenSim\4.5\Models\WorkingWithStaticOptimization
Precision 8
Load...
Save...
Run
Close
Help

a


Static Optimization Tool
Main Settings
Actuators and External Loads
Actuators
Additional force set files
Edit...
Append to model's force set
Replace model's force set
External Loads
External loads specification file
Load...
Save...
Run
Close
Help

b

Fig. 81 Secciones a configurar de la herramienta de optimización estática

7.2. Opción 2

Los archivos de entrada contemplan el modelo escalado con el que se va a trabajar, el archivo de cinemática que contiene los ángulos articulares (.sto), el cual se obtuvo como resultado de ejecutar RRA, el archivo de fuerzas externas (.xml) que hace parte de la configuración de dinámica inversa y el archivo de actuadores de RRA (.xml), dado que en las secciones anteriores se ha explicado cómo obtener cada uno de ellos, se recomienda revisar esta información en caso de no contar con alguno.

7.2.1. Ejecutar la herramienta de Optimización estática

Una vez tenga el modelo de OpenSim abierto presione *Tools* → *Static Optimization*…, como se ve en la **Fig. 80**. En la ventana emergente cargue los archivos correspondientes, **Fig. 81**, en la sección de *Input* (Marcado en rojo) cargue el archivo de cinemática que contiene los ángulos articulares (.sto), el cual se obtuvo como resultado de ejecutar RRA, en *Time* (Marcado en verde) agregue el rango de tiempo que usó para ejecutar dinámica inversa, en la sección de *Output* (Marcado en azul) agregue al final de la dirección \ResultsSO, lo que creará una carpeta para guardar estos resultados, cabe mencionar que en esta opción no se quiere el filtro activado en la opción anterior. Posteriormente vaya a la pestaña de *Actuators and external loads* y en la sección de *External Loads* (Marcado en amarillo) active la casilla y agregue el archivo de las fuerzas externas que empleó en dinámica inversa (.xml), aquel que contiene los indicadores del vector de fuerza de reacción del suelo, finalmente presione *Run*.

7.3. Evaluar el resultado de optimización estática

Para evaluar los resultados de ambas opciones es suficiente con emplear la herramienta de gráficos de OpenSim como se explicó en dinámica inversa, en el eje Y cargue el archivo de activación (.sto) que obtuvo como resultado de optimización estática, allí encontrará cada uno de los músculos de su modelo, puede seleccionar los que desee evaluar. Para el eje X seleccione *Time*. Puede buscar información de EMG o de activación muscular para compararlos con las activaciones simuladas, debe verificar que las magnitudes y los patrones coinciden. Si considera que sus resultados no son adecuados, puede ser debido a

que los valores de fuerza óptima del archivo de actuadores son muy grandes o pequeños, para esto considere modificarlos teniendo en cuenta que una fuerza óptima de gran magnitud requiere una activación muscular pequeña para producir un par o fuerza grande, por lo cual, una activación pequeña y una fuerza óptima pequeña no producirán un par lo suficientemente grande.

Adicionalmente, puede graficar las fuerzas residuales, para ello en lugar de cargar el archivo de activación en el eje Y, cargue el archivo de fuerzas (.sto), que también se obtiene como resultado de la optimización estática, allí puede revisar las componentes de las fuerzas que le interesen y ver si la magnitud es la esperada según su modelo.

8. Control muscular computacional

El control muscular computacional (CMC por sus siglas en inglés) es una herramienta utilizada en simulaciones de modelos musculoesqueléticos cuyo objetivo es calcular los niveles de excitación muscular necesarios para que las coordenadas generalizadas de un modelo sigan una trayectoria específica [10].

Para llevarlo a cabo se requieren una serie de archivos como se muestra en la **Fig. 82**, los cuales contienen información relevante que permite ejecutar el estudio, por lo que se describirá cómo construir cada uno.

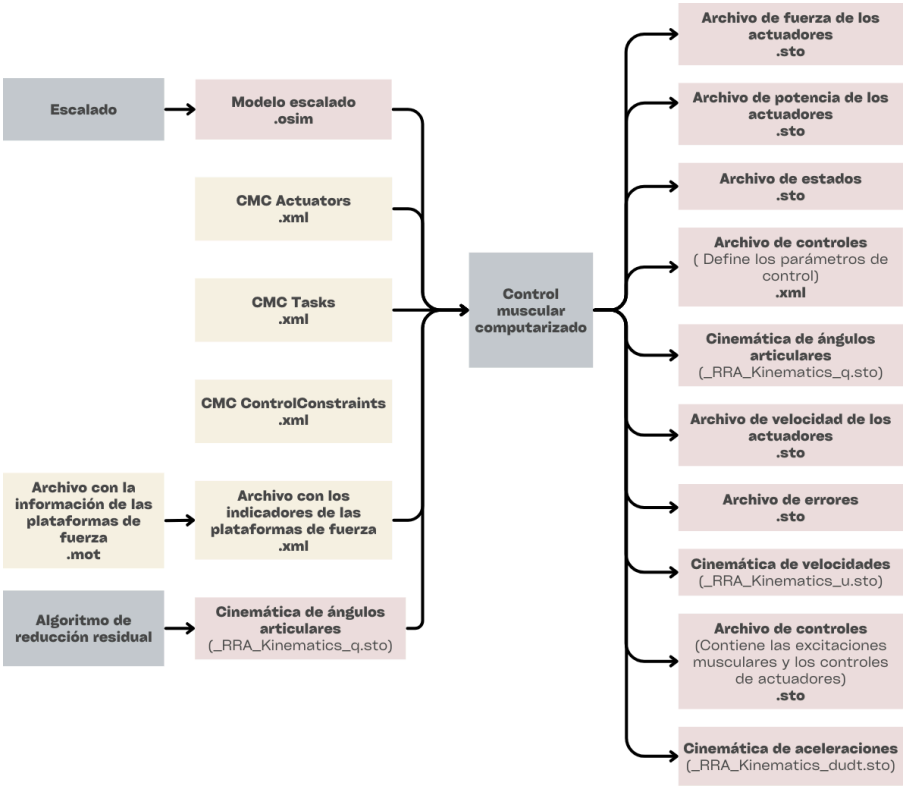


Fig. 82 Archivos de entrada y salida de la herramienta de control muscular computarizado de OpenSim

8.1. Archivo de configuración de CMC

Para emplear la herramienta de CMC de OpenSim es necesario realizar un archivo de configuración, así como se ha hecho en las sesiones anteriores, este archivo también se puede encontrar en la carpeta de *Gait2354* como *subject01_Setup_CMC.xml*, por lo que se recomienda realizar una copia para editarlo. Su contenido tiene una estructura similar a la que se muestra en la **Fig. 83,** este recibe seis archivos de entrada y produce diez archivos de salida.

- ***_controls.sto:** Controles de los actuadores residuales y de reserva además de las excitaciones de cada músculo.
- ***_controls.xml:** Contiene los parámetros de control empleados durante CMC, como restricciones de excitación, límites mínimos y máximos, y nodos de control en el tiempo.
- ***_Actuation_force.sto**: Fuerzas tanto musculares como de reserva y residuales.

- ***_states.sto:** Archivo con los estados del modelo, es decir los ángulos articulares, las longitudes de las fibras musculares, las velocidades y las activaciones.
- ***_RRA_Kinematics_dudt.sto:** Cinemática con las componentes de aceleración.
- ***_RRA_Kinematics_u.sto:** Cinemática con las componentes de velocidad.
- ***__RRA_Kinematics_q.sto:** Cinemática con las posiciones de los ángulos articulares.
- ***_Actuation_power.sto:** Incluye las potencias generadas por los actuadores a lo largo del movimiento simulado.
- ***_Actuation_speeds.sto:** Almacena las velocidades de los actuadores durante el movimiento.
- ***_pErr.sto:** Contiene los errores de cada coordenada en el tiempo.

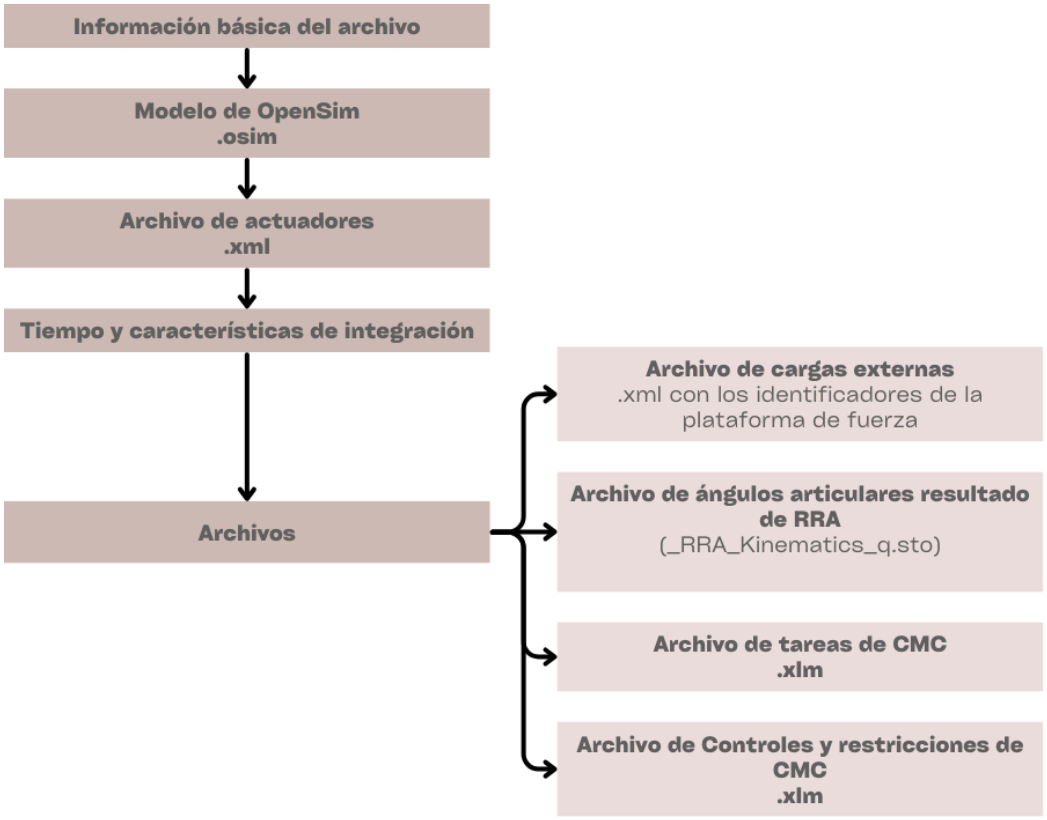


Fig. 83 Estructura del archivo de configuración de la herramienta de control muscular computarizado de OpenSim

Es necesario realizar una copia de otros tres archivos que configuran CMC, lo cuales se encuentran en la misma carpeta, *gait2354_CMC_ControlConstraints.xml, gait2354_CMC_Tasks.xml* y *gait2354_CMC_Actuators.xml.*

Las modificaciones generales de este archivo de configuración consisten en reemplazar los nombres de los archivos de entrada y de salida por el nombre con el que se haya guardado cada uno. En la **Fig. 84** se presentan los primeros cambios que se deben realizar en el archivo, modifique el nombre entre las etiquetas <*model_file*> para que corresponda con su modelo de OpenSim en el que ejecutó RRA, dado que a este se le ha ajustado el centro de masa del torso para reducir los residuos. Reemplace el nombre del archivo <*force_set_file*> por el nombre con que guardó el archivo de actuadores copiado de *Gait2354*, cuya edición se explicará más adelante. Por otro lado, puede o no modificar el nombre de la carpeta en la que se guardarán los resultados de este estudio. En esta misma figura se marcan las características de solución de las integrales de CMC, primero se debe determinar el valor de precisión, este dependerá de qué tan importante es la dinámica para su estudio, mientras mayor sea el valor más preciso será el modelo, acto seguido debe describir el rango de tiempo del análisis, use el mismo valor empleado para dinámica inversa, pero disminuya el tiempo inicial en al menos 0.03 segundos que es el tiempo que le toma a la herramienta comenzar a ejecutarse. Ahora bien, los valores de máximo número de pasos de integración, así como el valor máximo y mínimo del paso de integración y la tolerancia deben estar balanceados entre ellos, por lo que es importante comprender a qué hacen referencia y cómo se relacionan. El valor máximo de pasos de integración indica la cantidad total de integraciones antes de finalizar CMC, mientras que el máximo paso de integración indica la cantidad máxima de tiempo durante el cual se espera la solución de cada integral, por lo que a menor tamaño mayor deberá ser la tolerancia al error.

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSimDocument Version="20302">
    <CMCTool name="subject0026_walk1">
    <!--Name of the .osim file used to construct a model.-->
    <model_file> subject01_simbody_adjusted.osim </model_file>
    <!--Replace the model's force set with sets specified in
        <force_set_files>? If false, the force set is appended to.-->
    <replace_force_set> false </replace_force_set>
    <!--List of xml files used to construct an force set for the model.-->
    <force_set_files> gait2354_CMC_Actuators.xml </force_set_files>
    <!--Directory used for writing results.-->
    <results_directory> ./ResultsCMC </results_directory>
        <!--Output precision.  It is 20 by default.-->
    <output_precision> 20 </output_precision>
    <!--Initial time for the simulation.-->
        <initial_time>       0.80000000 </initial_time>
    <!--Final time for the simulation.-->
        <final_time>       1.20000000 </final_time>
        <!--Flag indicating whether or not to compute equilibrium values for
            states other than the coordinates or speeds.  For example, equilibrium
            muscle fiber lengths or muscle forces.-->
        <solve_for_equilibrium_for_auxiliary_states> true </solve_for_equilibrium_for_auxiliary_states>
    <!--Maximum number of integrator steps.-->
    <maximum_number_of_integrator_steps> 30000 </maximum_number_of_integrator_steps>
    <!--Maximum integration step size.-->
        <maximum_integrator_step_size>       1.0000 </maximum_integrator_step_size>
        <!--Minimum integration step size.-->
        <minimum_integrator_step_size>       1e-10 </minimum_integrator_step_size>
    <!--Integrator error tolerance. When the error is greater, the integrator
        step size is decreased.-->
        <integrator_error_tolerance>       1e-5 </integrator_error_tolerance>
    <!--Set of analyses to be run during the investigation.-->
    <AnalysisSet name="Analyses">
        <objects>
            <Kinematics name="Kinematics">
                <!--Flag (true or false) specifying whether whether on. True by default.-->
                <on> true </on>
```

Fig. 84 Parámetros a editar en el archivo de configuración de CMC

Entre las etiquetas de *AnalysisSet* se especifican la lista de análisis que se pueden realizar mientras se ejecuta la herramienta de CMC, entre las opciones se encuentran cinemática, análisis muscular, actuadores, entre otros, en este caso se encuentran activos cinemática y actuadores, donde el primero se centra en estudiar en movimiento articular del modelo y el segundo se hace cargo de las fuerzas y los momentos generados por los actuadores. Para ello se definen los tiempos de inicio y finalización, que en este caso están entre menos infinito e infinito dado que se ejecutan durante toda la simulación, adicionalmente se define la frecuencia con la que se deben registrar los resultados durante la integración numérica, entre las etiquetas de <*step_interval*>, en este caso cada 10 pasos de integración se registran los resultados, **Fig. 85**. Debe considerar si para su estudio es relevante ejecutar estos análisis y cuáles de ellos desea emplear.

```
<AnalysisSet name="Analyses">
    <objects>
        <Kinematics name="Kinematics">
            <on> true </on>
                <!--Start time.-->
                <start_time> -infinity </start_time>
                <!--End time.-->
                <end_time> infinity </end_time>
            <!--Specifies how often to store results during a simulation. More
                specifically, the interval (a positive integer) specifies how many
                successful integration steps should be taken before results are
                recorded again.-->
            <step_interval> 10 </step_interval>
            <!--Flag (true or false) indicating whether the results are in degrees or
                not.-->
            <in_degrees> true </in_degrees>
                <!--Names of generalized coordinates to record kinematics for.  Use 'all'
                    to record all coordinates.-->
                <coordinates> all </coordinates>
        </Kinematics>
        <Actuation name="Actuation">
            <!--Flag (true or false) specifying whether whether on. True by default.-->
            <on> true </on>
                <!--Start time.-->
                <start_time> -infinity </start_time>
                <!--End time.-->
                <end_time> infinity </end_time>
            <!--Specifies how often to store results during a simulation. More
                specifically, the interval (a positive integer) specifies how many
                successful integration steps should be taken before results are
                recorded again.-->
            <step_interval> 10 </step_interval>
            <!--Flag (true or false) indicating whether the results are in degrees or
                not.-->
            <in_degrees> true </in_degrees>
        </Actuation>
```

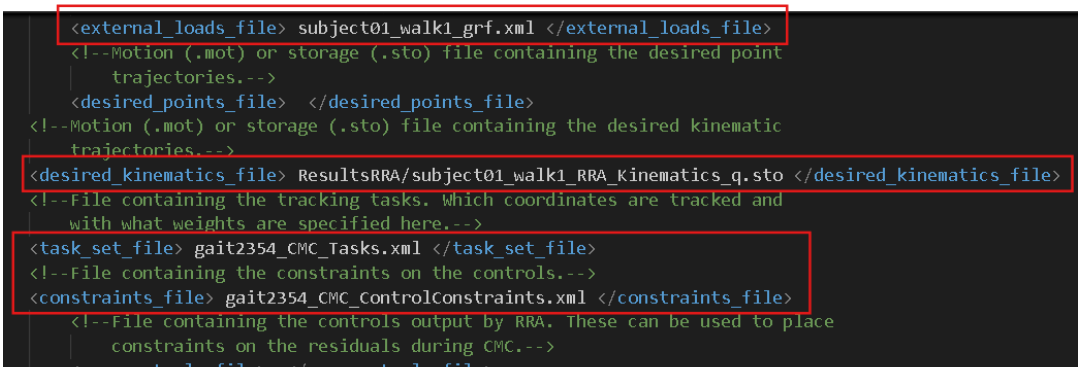
Fig. 85 Parámetros a editar en el archivo de configuración de CMC

Entre las etiquetas <*external_loads_file*> reemplace el nombre del archivo por el archivo que contiene los indicadores y el cuerpo de aplicación del registro de la plataforma de fuerza, usado en dinámica inversa, mientras que en <*desired_kinematics_file*> reemplace el nombre por el archivo que obtuvo como resultado de RRA que contiene las componentes de los ángulos articulares, finalmente, entre las etiquetas <*task_set_file*> y <*constraints_file*> reemplace el nombre por los archivo de tareas y restricciones que copió de la carpeta *Gait2354,* como se muestra marcado en la **Fig. 86**.

Fig. 86 Parámetros a editar en el archivo de configuración de CMC

8.1.1. CMC ControlConstraints

En este archivo se definen los límites de control para los actuadores de un mod



elo, los cuales determinan los rangos en los que los actuadores pueden aplicar fuerzas o momentos, controlando así el comportamiento del modelo durante las simulaciones, esto asegura que los actuadores operen dentro de rangos fisiológicamente razonables y evita soluciones irreales o físicamente imposibles durante la simulación. Dentro de cada sección de *ControlLinear*, como se marca en la **Fig. 87,** está descrito un músculo del modelo, por lo que debe revisar uno a uno y verificar que los nombres correspondan con su modelo y que no falte ninguno, para revisar los nombres de los músculos de su modelo puede desplegar las pestañas de *Forces* → *Muscles*, dentro de la ventana de *Navigator* en OpenSim, allí encontrará la lista de músculos del modelo, como se muestra en la **Fig. 88**. Por otro lado, los valores de las etiquetas dentro de cada sección pueden ser modificados teniendo en cuenta lo que representa cada uno, entre las etiquetas <*default_min*> y <*default_max*> van los valores mínimos y máximos del control, es decir, los límites de este, los cuales están expresados como una proporción de la fuerza óptima del actuador, mientras que las constantes <*kp*> y <*kv*> son ganancias de control para la dinámica del actuador, como en un sistema de control proporcional derivativo.

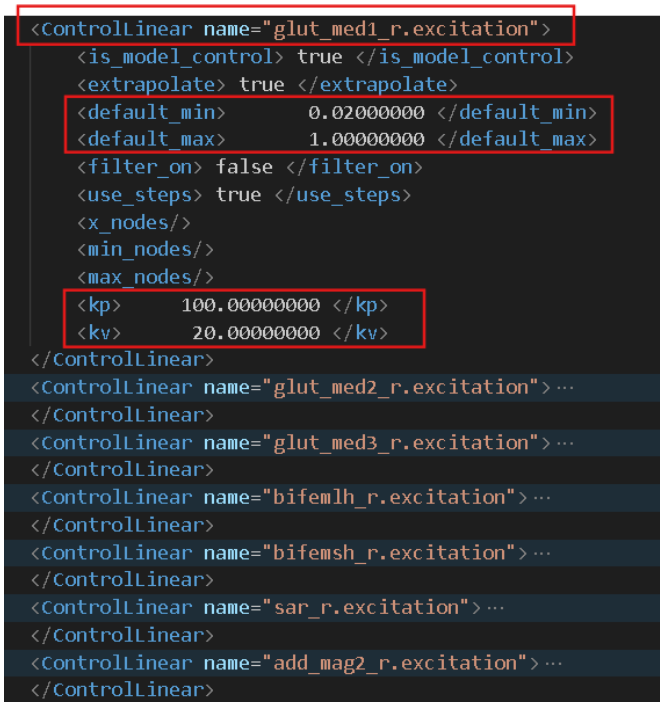


Fig. 87 Parámetros a editar en el archivo de *ControlConstraints* de CMC

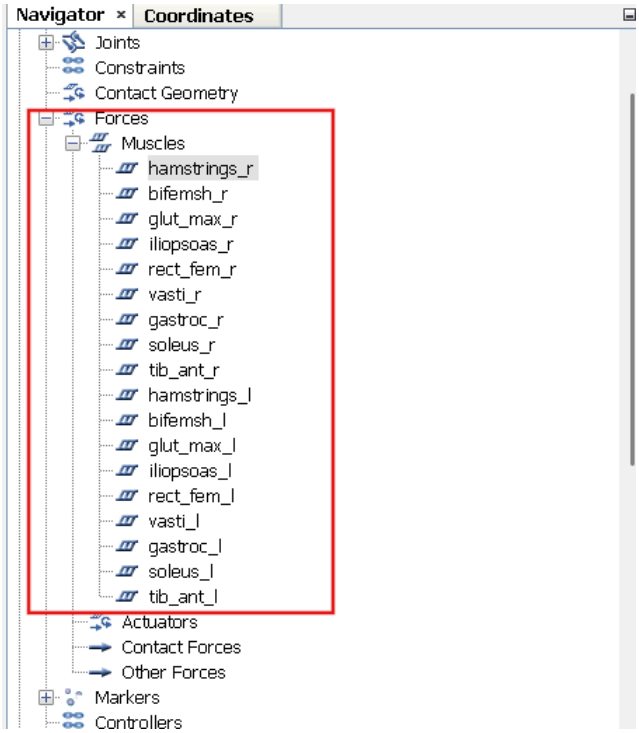


Fig. 88 Visualización de los músculos del modelo

8.1.2. CMC Tasks

Este archivo especifica qué coordenadas son importantes y con qué precisión deben rastrearse, lo que guía a la herramienta de CMC para ajustar el movimiento del modelo. El archivo tiene una sección de *CMC_Joint* para cada movimiento del modelo, como se muestra en la **Fig. 89**, por lo que se debe revisar que los nombres coincidan con los del modelo que está trabajando y verificar que no falte ninguno, para revisar los nombres de los movimientos de su modelo puede abrir la ventana de *coordinates* en OpenSim, al lado de *Navigator* y allí encontrará la lista de movimientos del modelo, como se muestra en la **Fig. 90**. Ahora bien, respecto a los valores dentro del resto de etiquetas, su modificación depende de los objetivos propios de cada estudio, por lo que es necesario identificar a qué hace referencia cada una.

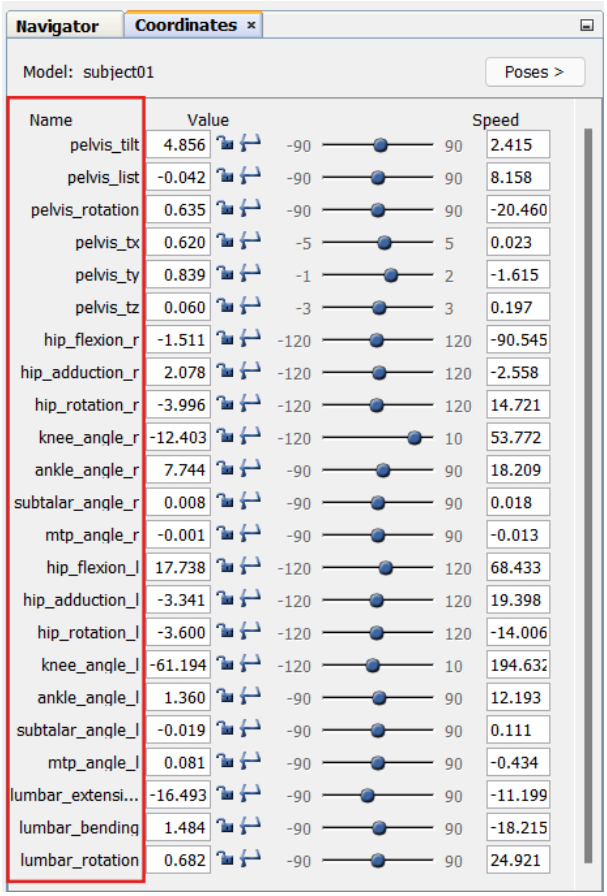


Fig. 89 Visualización de las articulaciones y movimientos del modelo

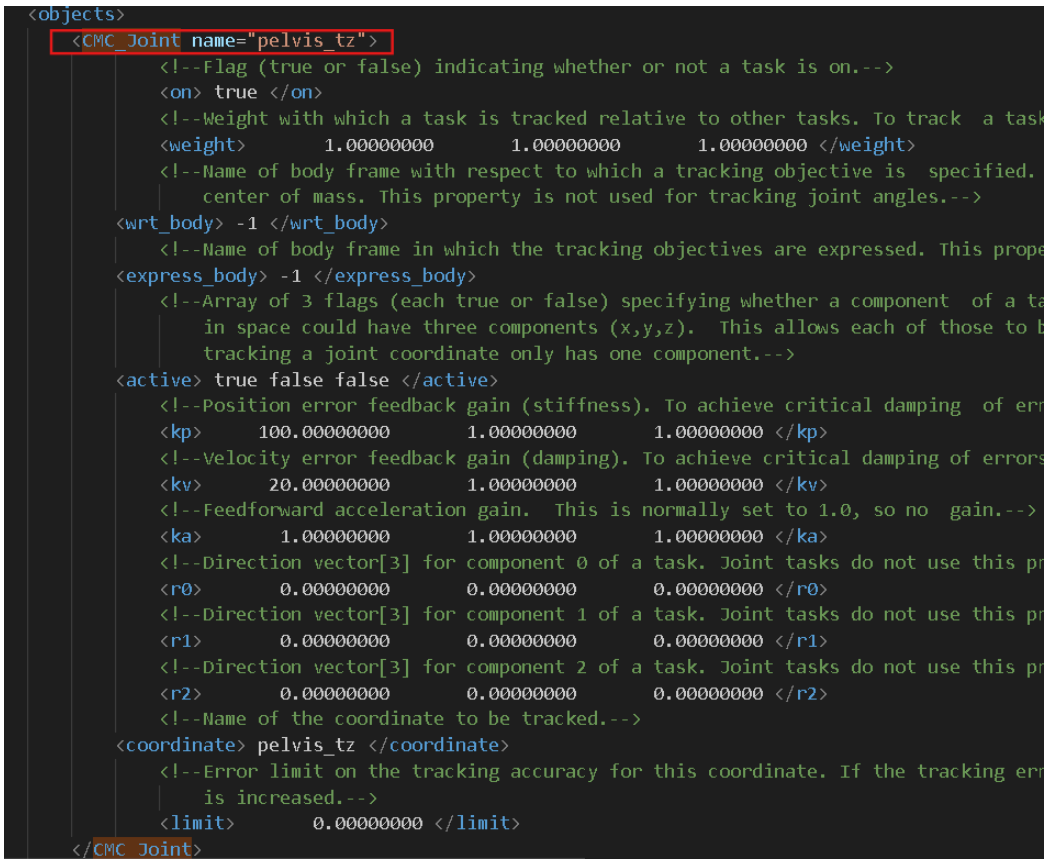


Fig. 90 Parámetros a editar en el archivo de tareas de CMC

- **on:** Dependiendo de si su valor es *true* o *false*, la tarea estará habilitada o deshabilitada respectivamente
- **weight:** Indica el peso con el que se realizará la tarea respecto a otra, a mayor peso mayor será la prioridad y la precisión con que se ejecute esa tarea
- **wrt_body y express_body:** Estas propiedades se usan para definir la posición de un punto y su relación con otros marcos en el modelo. Son relevantes para tareas que implican el seguimiento de trayectorias de puntos en el espacio, pero no se aplican al seguimiento de coordenadas articulares.
- **Active:** Indica qué coordenadas están activas o inactivas, es decir, *true false false* indica que la coordenada X está activa mientras que Y y Z están inactivas, esto permite seguir la trayectoria del punto a través de las tres componentes, sin embargo, para articulaciones solo se activa la primera componente.
- **kp, kv y ka:** Corresponden a los parámetros de la ley de control proporcional derivativo, con la cual se calculan las aceleraciones de la cinemática. Esta ley tiene tanto ganancia de retroalimentación rigidez, como ganancia de amortiguación y ganancia de avance de aceleración las cuales se especifican para cada coordenada dentro de kp, kv y ka respectivamente.

- **r0, r1, r2:** indican vectores de dirección para los tres componentes de una tarea, sin embargo, no se utilizan para tareas de seguimiento de una única coordenada de articulación.

8.1.3. CMC Actuators

En este archivo se definen los actuadores que usará el modelo durante el análisis de CMC, existen tres tipos como se muestra en la **Fig. 91**, punto, torque y coordenada, en cada uno se especifica la fuerza del actuador y los valores mínimos y máximos que puede aplicar, además de algunas características propias que dependen de cada tipo.

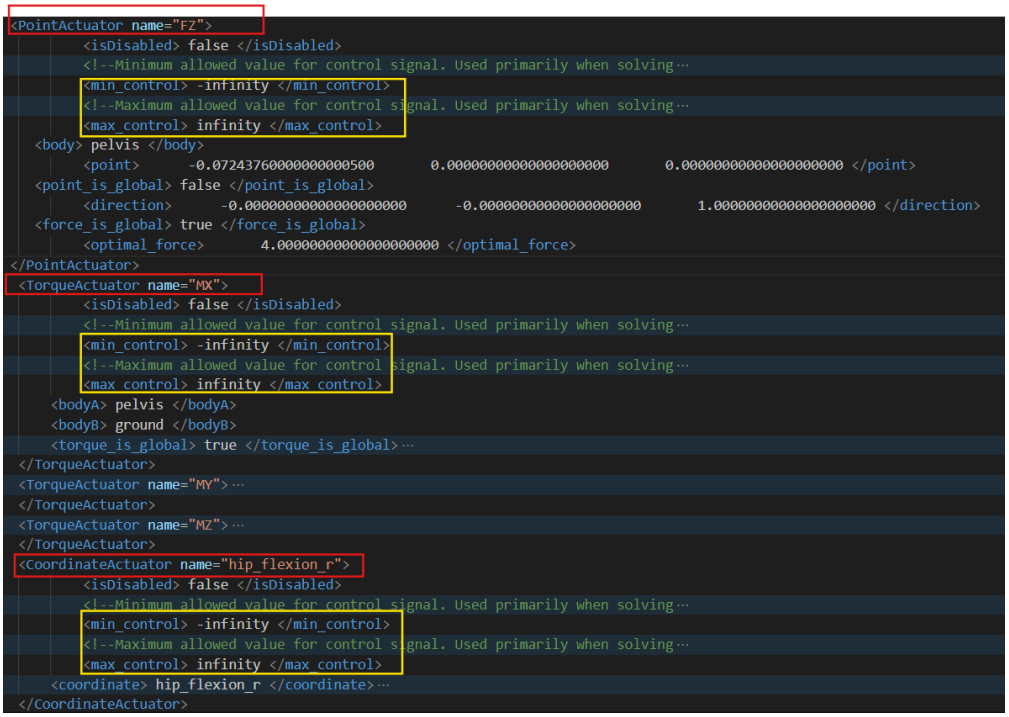


Fig. 91 Parámetros a editar en el archivo de actuadores de CMC

- **PointActuator:** Además de los límites y la fuerza se debe especificar el cuerpo, el punto en el cuerpo y la orientación de este. Para ubicar el punto debe de buscar el centro de masa del cuerpo de ese actuador, en este caso sería la pelvis, para ello seleccione el cuerpo desplegando la pestaña *Body* en la ventana de *Navigator*, luego seleccione el cuerpo y vaya a la ventana de *Properties*, allí verá el apartado de *mass_center*, donde se indica el punto en el que está el centro de masa de ese objeto, como se muestra en la **Fig. 92.**
- **TorqueActuator:** Como parámetros adicionales se especifican el cuerpo A y B, es decir el que ejerce el toque y el que lo recibe, además del eje.

- **CoordinateActuator:** Hay una sección de este tipo para cada movimiento del modelo como en los dos archivos de configuración previos, por lo tanto además de los parámetros de fuerza y límites, se debe agregar el nombre de la coordenada, es decir, del movimiento.

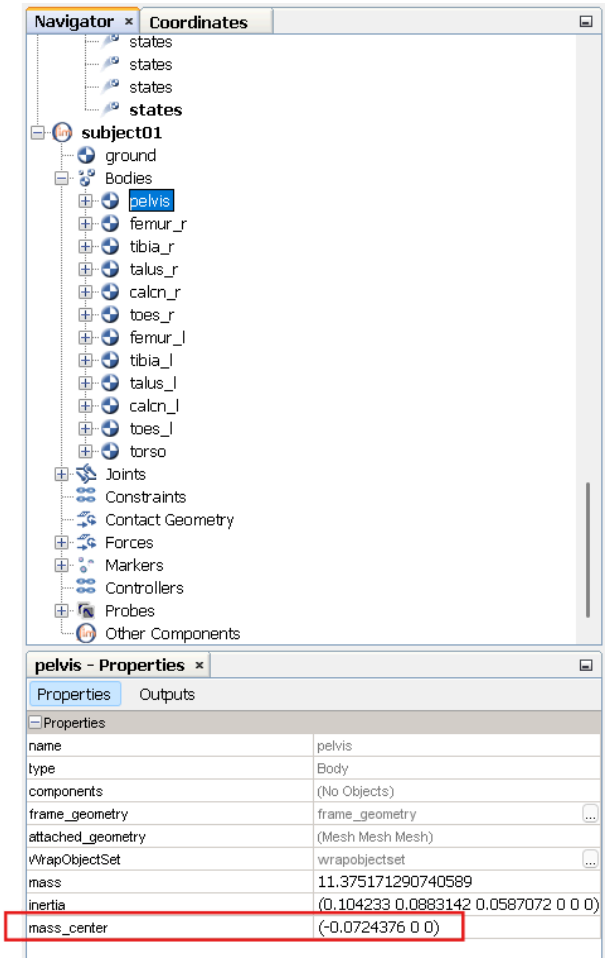
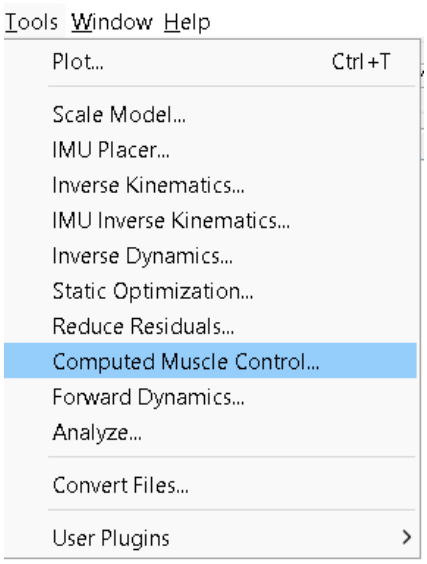


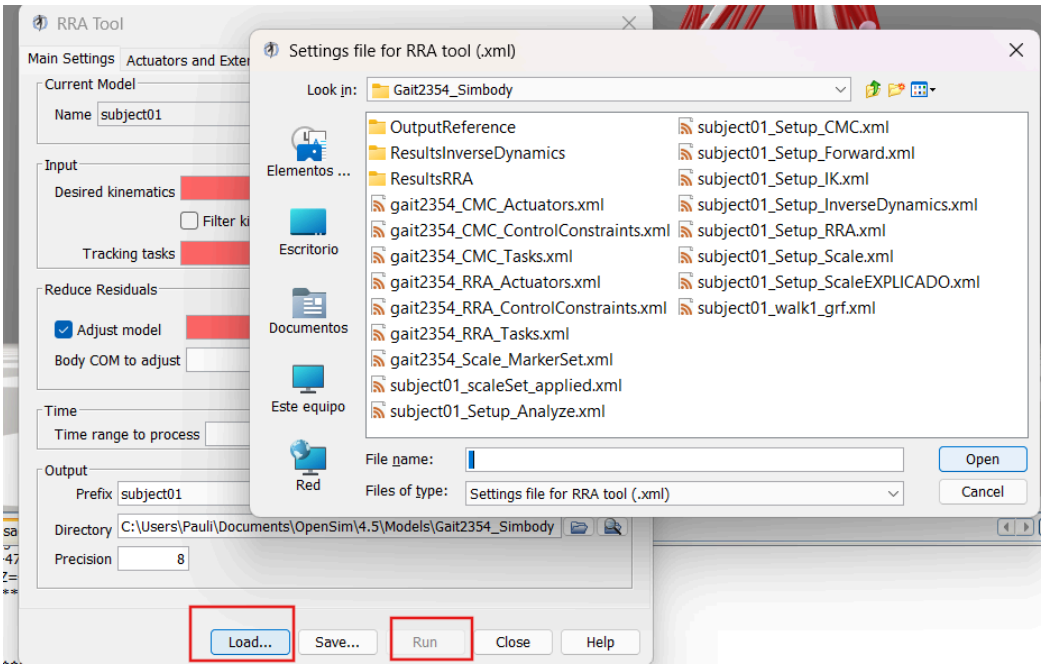
Fig. 92 Visualización del centro de masa de un cuerpo

8.2. Ejecutar la herramienta de CMC

Una vez se hayan editado adecuadamente los archivos de configuración de la herramienta de CMC es posible llevar a cabo el estudio, para esto es necesario seguir la siguiente ruta en OpenSim, como se muestra en la **Fig. 93a,** *Tools* → *Computed Muscle Control…,* esto abrirá una ventana emergente en la que podrá cargar el archivo de configuración principal de CMC haciendo uso del botón *Load*, como se muestra en la **Fig. 93b**. Una vez cargue el archivo verifique que toda la información se haya agregado correctamente y presione *Run.*



a



b

Fig. 93 Ejecutar analizador de OpenSim

8.3. Evaluar el resultado de CMC

Hay varias consideraciones que debe tener en cuenta al momento de verificar CMC:

- Revisar que los pares de reserva máximos del actuador sean inferiores al 10% del par máximo de la articulación, para ello revise el archivo con el sufijo

**_Actuation_force.sto* y busque las columnas que corresponden con los pares de reserva, generalmente con el sufijo *reserve*.

- Verificar que las fuerzas residuales máximas y los momentos residuales máximos estén dentro de los rangos presentados en la **TABLA IV.** Para ello busque las columnas con nombres como *FX_ground_pelvis, FY_ground_pelvis, FZ_ground_pelvis* (fuerzas residuales) y *MX_ground_pelvis, MY_ground_pelvis, MZ_ground_pelvis* (momentos residuales), en el mismo archivo del ítem anterior.
- Revisar los errores cinemáticos de posición y rotación, para ello abra el archivo con el sufijo **_pErr.sto,* y compare los errores con los umbrales indicados en la **TABLA IV**.
- Comparar la cinemática calculada con RRA y con CMC. Para ello haga uso de la herramienta *Plot* de OpenSim y cargue los archivos con los sufijos **_Kinematics_q.sto* de CMC y **_Kinematics_q.sto* de RRA. En general, deberían coincidir bien siempre que se tenga activado el *fast optimization target*, que se encuentra entre las etiquetas de <*use_fast_optimization_target*> del archivo de configuración general de CMC.
- Comparar las activaciones musculares simuladas con la información experimental de EMG en caso de que cuente con registros, sino puede recurrir a la literatura. Las activaciones deben mostrar una coincidencia tanto temporal como de magnitud.

TABLA IV

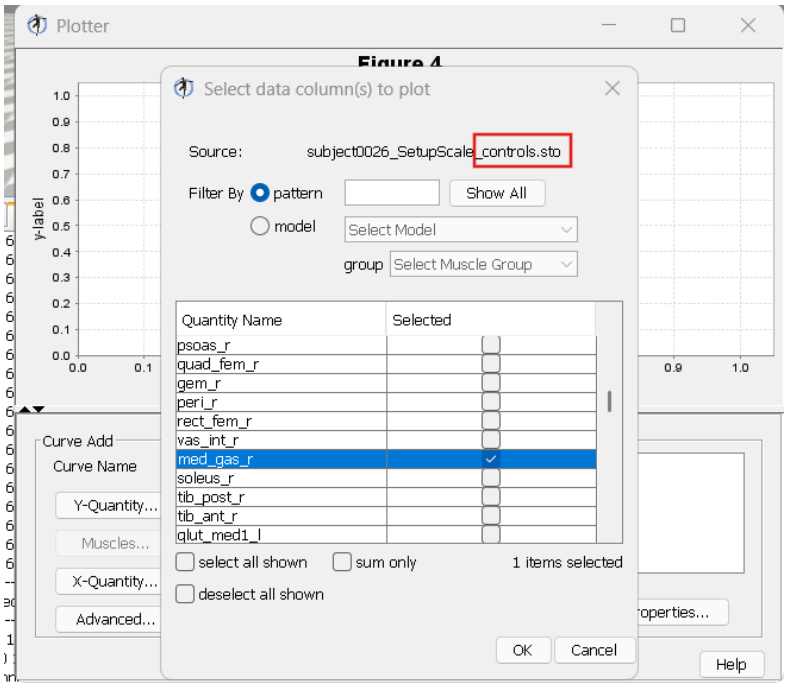
Criterios de evaluación de CMC

Parámetro	Bueno	Aceptable	Malo
Máxima fuerza residual (N)	0-10	10-25	> 25
RMS de la fuerza residual (N)	0-10	10-25	> 25
Máximo momento residual (Nm)	0-50	50-75	> 75
RMS del momento residual (Nm)	0-30	30-50	> 50
MAX pErr (trans, cm)	0-1	1-2	> 2
RMS pErr (trans, cm)	0-1	1-2	> 2
MAX pErr (rot, deg)	0-2	2-5	> 5
RMS pErr (rot, deg)	0-2	2-5	> 5
Máxima reserva (Nm)	0-25	25-50	> 50

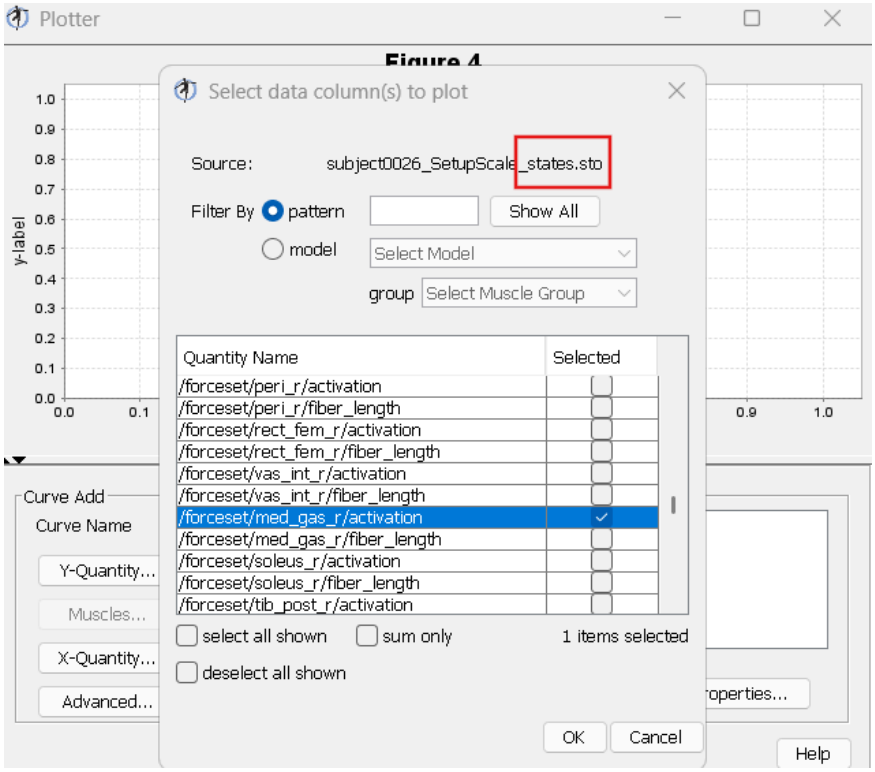
RMS de la reserva (Nm)	0-10	10-25	> 25

8.4. Corregir CMC

Para mejorar el ajuste de las activaciones musculares con los valores de EMG es posible modificar el archivo de *ControlConstraints* que empleó para el estudio, para ello debe de graficar primero una comparación entre los valores de excitación muscular de reserva empleados por CMC, los cuales se encuentran en el archivo de controles (sufijo **_controls.sto*), y los estados musculares calculados, los cuales se alojan en el archivo de estados (sufijo **_states.sto*). Abra la herramienta de *Plot* de OpenSim y en el eje Y cargue primero el archivo de controles y seleccione el músculo de interés, seleccione tiempo en el eje X y agregue la gráfica, como se ve en la **Fig. 94a**, posteriormente en la misma figura cargue el archivo de estados, de nuevo en el eje Y, seleccione el mismo músculo, verificando que tenga el sufijo *Activation* y añádalo a la gráfica, **Fig. 94b**.



a



b

Fig. 94 Ejecutar herramienta de gráficas para evaluar CMC

En la **Fig. 95** se presenta un ejemplo de una mala coincidencia entre el trazado proveniente del archivo de controles (*soleus_r*) y de estados (*soleus_r_activation*), el trazado adecuado debería ser el de controles, donde este músculo no presenta una magnitud de activación grande durante la fase de balanceo, por lo cual hay que disminuir el valor de activación entre 0.72 y 0.99seg para mejorar el ajuste.

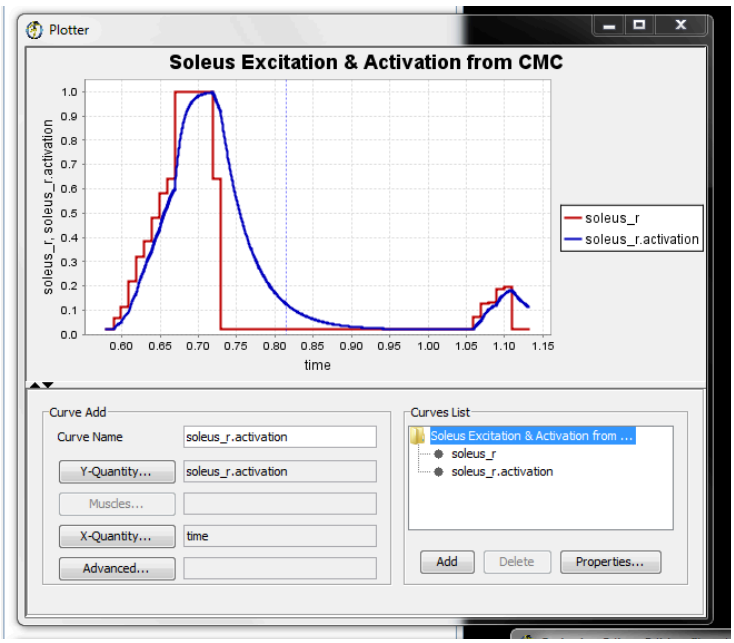


Fig. 95 Excitación y activación del músculo sóleo

Nota. fuente: https://opensimconfluence.atlassian.net/wiki/spaces/OpenSim/pages/53089721/CMC+Settings+Files+and+XML+Tag+Definitions

Para mejorar este comportamiento se debe modificar el archivo de *ControlConstraints*, comience por dar clic en *Edit→Excitations*, como se muestra en la **Fig. 96**, posteriormente cargue el archivo mencionado y seleccione el músculo de interés, esta herramienta le permite agregar nodos mínimos y máximos los cuales controlan la magnitud de activación del músculo en el tiempo, el valor de un nodo se empieza a aplicar antes de llegar al siguiente nodo, lo que crea una transición suave en lugar de un cambio brusco. Esto ayuda a que la activación del músculo cambie gradualmente y el movimiento en la simulación sea más realista.

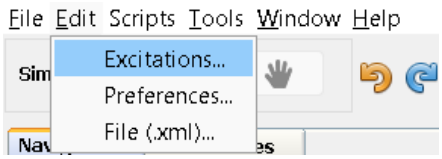


Fig. 96 Herramienta de excitaciones de OpenSim

Para controlar el editor tenga en cuenta las funciones descritas en la **TABLA V.**

TABLA V

Controles de la herramienta de excitaciones de OpenSim

Acción	Instrucción

Ampliar región	Seleccionar un área de interés con el ratón, anclando o haciendo clic izquierdo y arrastrando hacia la parte inferior derecha para ampliar un rectángulo.
Alejar o reajustar datos	Anclar o hacer clic izquierdo y arrastrar hacia la izquierda.
Hacer zoom al centro	Utilizar la tecla I.
Alejar el centro	Utilizar la tecla O.
Mover a la izquierda	Utilizar la tecla L.
Mover a la derecha	Utilizar la tecla R.
Mover hacia arriba	Utilizar la tecla U.
Mover hacia abajo	Utilizar la tecla D.
Activar modo de selección	Mantener presionado el botón CTRL (CTRL + option en Mac).
Seleccionar puntos individuales	Hacer clic izquierdo con el ratón.
Acumular selección	Utilizar Shift + clic izquierdo del ratón.
Seleccionar con cuadro	Dibujar un cuadro para seleccionar todos los puntos dentro de él.
Deseleccionar todos los puntos	Hacer clic izquierdo en el fondo.
Agregar puntos al panel	Hacer clic derecho dentro de un panel y utilizar el menú emergente para agregar puntos.

Agregue el trazado de puntos que considere necesarios, una vez finalice presione *save* para sobreescribir el archivo de *ControlConstraints.* En la **Fig. 97** Se presenta un ejemplo de los nodos que se emplearon para corregir la activación del músculo sóleo mencionado anteriormente, donde la línea roja conecta los nodos.

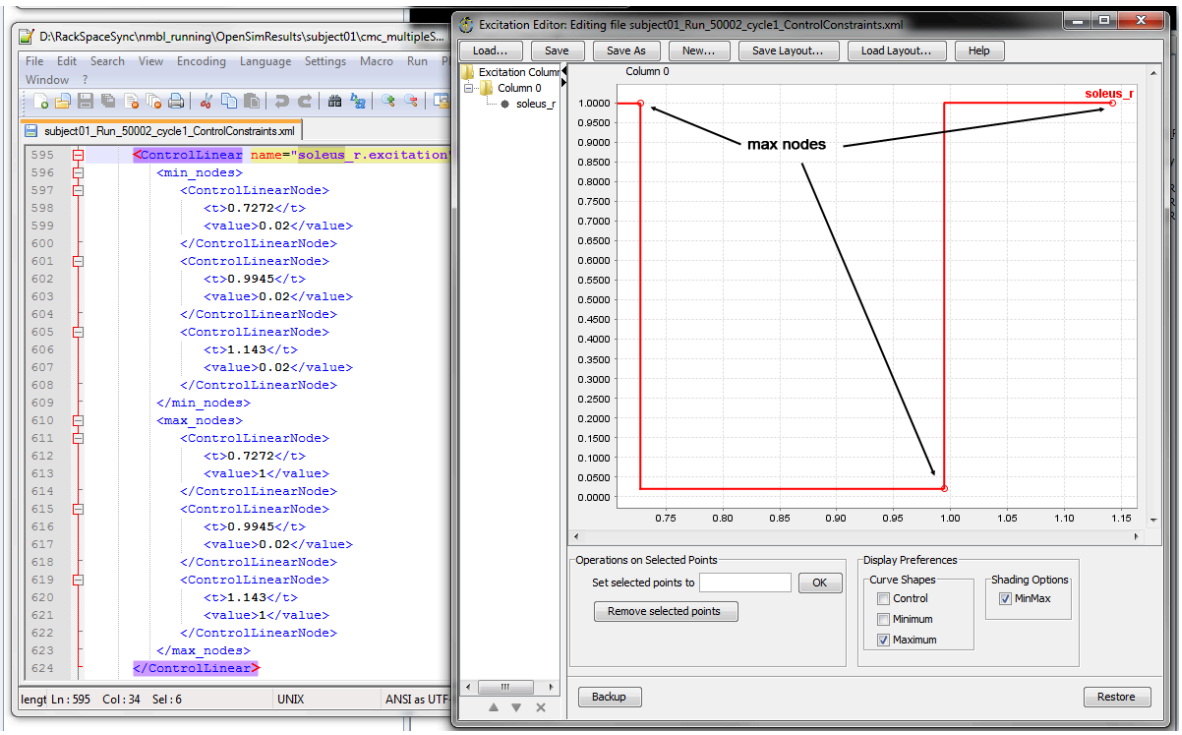


Fig. 97 Corrección de activación muscular

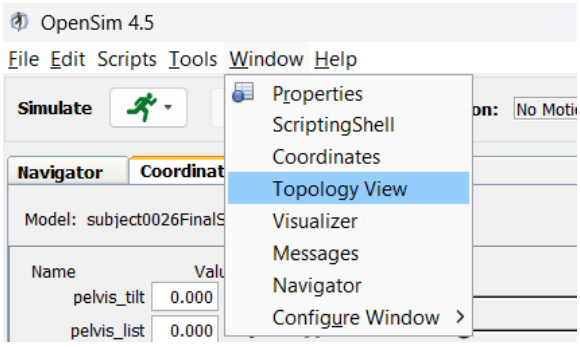
9. Simulación de amputación transfemoral

Para realizar la simulación de amputación transfemoral es necesario ejecutar modificaciones geométricas en el modelo, desde la sustitución de músculos y huesos por un dispositivo protésico hasta cambios en la longitud tanto del fémur como de los músculos que actúan sobre él.

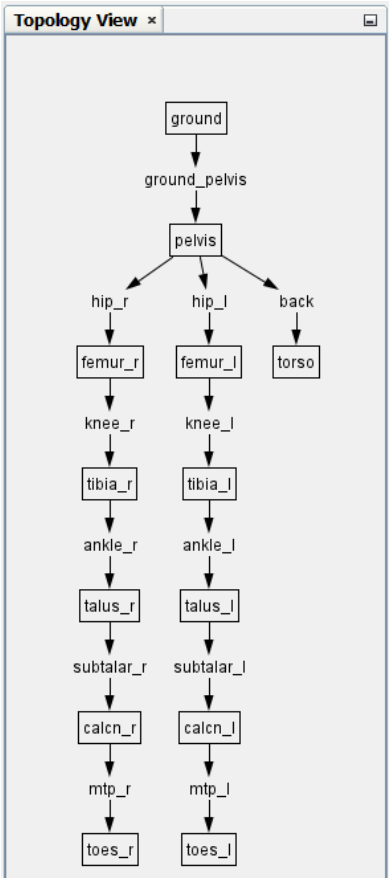
9.1. Eliminación de componentes

En esta sección se describe el proceso para eliminar todos los componentes óseos y musculares debajo del fémur del modelo seleccionado en la sección 1, se recomienda realizar una copia del mismo para no modificar el archivo original en caso de requerirse en estudios futuros.

Para esto hay que tener en cuenta las siguientes consideraciones, primero, despliegue la ventana de *Topology View* de su modelo de OpenSim, esta se encuentra en la barra superior, *Windows* → *Topology View*, como se muestra en la **Fig. 98a**. Esta herramienta permite ver un diagrama de su modelo en términos de cuerpos y articulaciones (**Fig. 98b),** lo que facilita la compresión de la estructura jerárquica del mismo, dado que OpenSim se rige por una estructura Padre-Hijo, es decir que cada articulación tiene un objeto Padre que coordina o dirige el movimiento de un objeto Hijo.



a



b

Fig. 98 Topology view

Cada cuadro es un cuerpo, dentro del cual puede ubicarse más de una geometría, como es el caso de la tibia, la cual contiene los huesos de la tibia y la fíbula o peroné, mientras que entre las flechas se describen las articulaciones, las cuales hacen relación al movimiento relativo entre un conjunto de cuerpos.

En segundo lugar, es necesario conocer las estructuras óseas y articulares que se van a remover, a partir de este diagrama y considerando que se realizará una simulación de amputación transfemoral de miembro izquierdo, se eliminarán y posteriormente se reemplazarán todos los componentes listados debajo de *femur_l* en la **Fig. 98b.**

Como última consideración, es necesario conocer los músculos que se deben reinsertar y aquellos cuya longitud remanente no es suficiente como para mantenerlos en el modelo, por lo que se recomienda revisar la anatomía de los miembros inferiores en relación a los puntos de origen e inserción de los músculos, así como los músculos que están representados en su modelo de OpenSim, para ver qué músculos componen el modelo basta con que despliegue el menú *Forces* → *muscles* → *all* en la ventana de *Navigator*, como se muestra en la **Fig. 99**, allí podrá ver un listado de todos los músculos del modelo y si hace doble clic en cualquiera de ellos se mostrará su ubicación sobre el modelo.

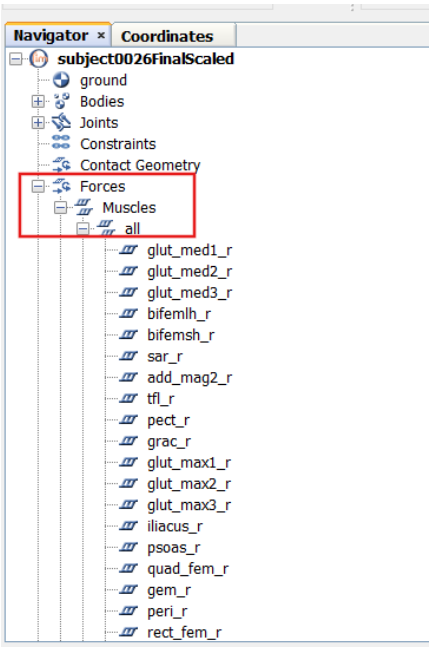


Fig. 99 Músculos del modelo

9.1.1. Componentes óseos

Comience por abrir el archivo .osim de su modelo en un editor de código o de texto, la estructura del archivo consta de 9 sesiones (**Fig. 100**) donde se describen diferentes componentes de la simulación.

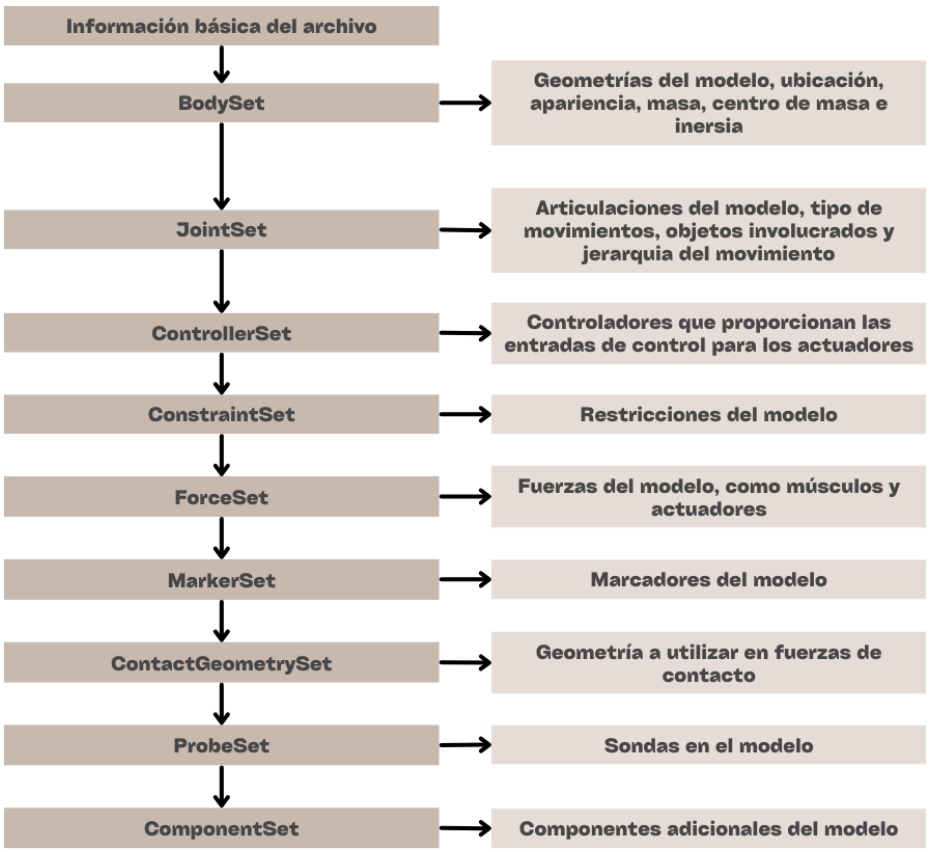


Fig. 100 Estructura de un modelo de OpenSim

En este caso, como la amputación será de la pierna izquierda, los cuerpos a eliminar serán *tibia_l, talus_l, calcn_l* y *toes_l,* por lo que se debe desplegar la sección *BodySet* y buscar estos cuerpos, elimine desde la etiqueta de apertura de cada uno, hasta la de cierre (*<Body ... > ... </Body>*), como se ve en la **Fig. 101.** En caso de que su extremidad sea la derecha o la altura de la amputación sea diferente, debe considerar cuáles son los cuerpos que debe eliminar, para esto le sirve analizar el diagrama que se presentó en la **Fig. 98b**.

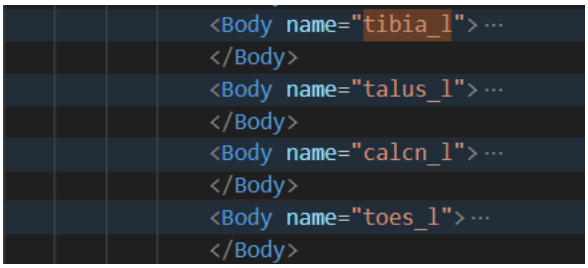


Fig. 101 Secciones de *Body* dentro del modelo de OpenSim

Posteriormente despliegue la sección *JointSet* y elimine las articulaciones que ya no son funcionales, en este caso son *knee_l, ankle_l, subtalar_l, mpt_l.* Para verificar cuáles son sus articulaciones de interés revise la ventana de *Topology View* de su modelo, debe

eliminar entre las etiquetas que abren y cierran cada articulación, dependiendo de su modelo sus etiquetas pueden variar dado que OpenSim tiene diferentes tipos de articulación, para identificarlas fácilmente busque el nombre de la articulación y colapse la vista de la misma en el editor de texto o código que esté empleando, esto le permitirá ver sólo las etiquetas de apertura y cierre, adicionalmente verifique que tengan el sufijo *Joint*. En la **Fig. 102** se muestra un ejemplo de las articulaciones a eliminar, en este modelo son *CustomJoint.*

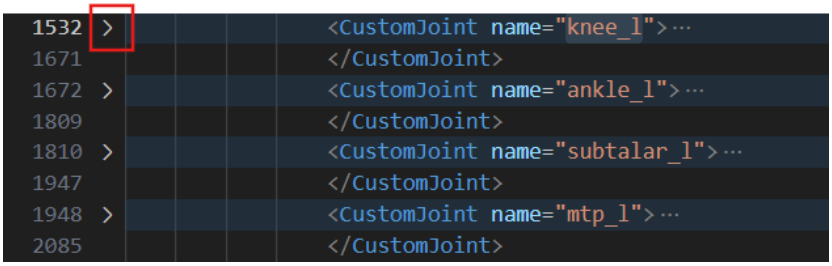


Fig. 102 Secciones para las articulaciones dentro de un modelo

9.1.2. Componentes musculares

Eliminar los componentes musculares que ya no actúan sobre el modelo requiere mayor atención al detalle, en la **Fig. 103** se presenta un diagrama que describe de manera general qué contiene un modelo muscular tipo *Thelen2003Muscle*, estos se alojan dentro de la sección *ForceSet* y cuentan con una subsección *Thelen2003Muscle* para cada músculo del modelo. OpenSim emplea dos modelos musculares, los tipo Thelen y los tipo Millard, para revisar las especificaciones de cada uno puede remitirse a los siguientes enlaces: Thelen2003, Millard2012.

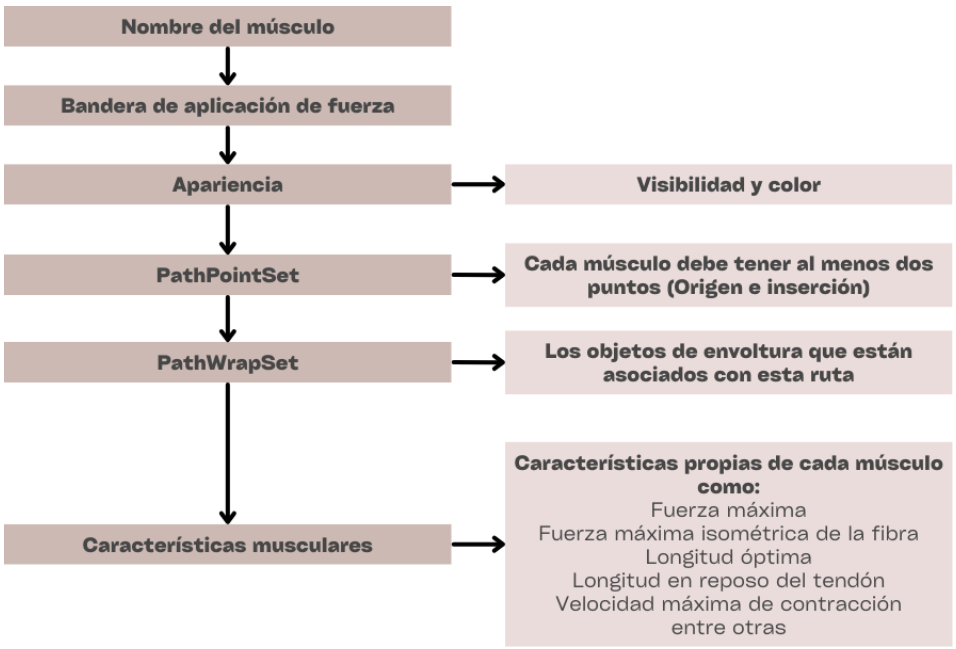


Fig. 103 Estructura de un modelo muscular tipo *Thelen2003Muscle*

Para eliminar adecuadamente los puntos de los músculos que ya no actúan debido a que la porción ósea fue eliminada se recomienda usar la herramienta de búsqueda del editor de texto o código, generalmente se activa usando *Ctrl+F,* se mostrarán algunos ejemplos de músculos para entender la dinámica de eliminación de puntos o músculos según el caso, dado que la lista completa de músculos depende del modelo seleccionado.

Soleus_l: Como se ve en la **Fig. 104,** este músculo tiene solo dos puntos, origen e inserción (*Soleus_l-P1 y Soleus_l-P2*), en la tibia y el calcáneo respectivamente, por lo que se puede eliminar por completo desde las etiquetas de apertura y cierre *<Thelen2003Muscle> ... </Thelen2003Muscle>,* puesto que tanto la tibia como el calcáneo se eliminaron previamente.

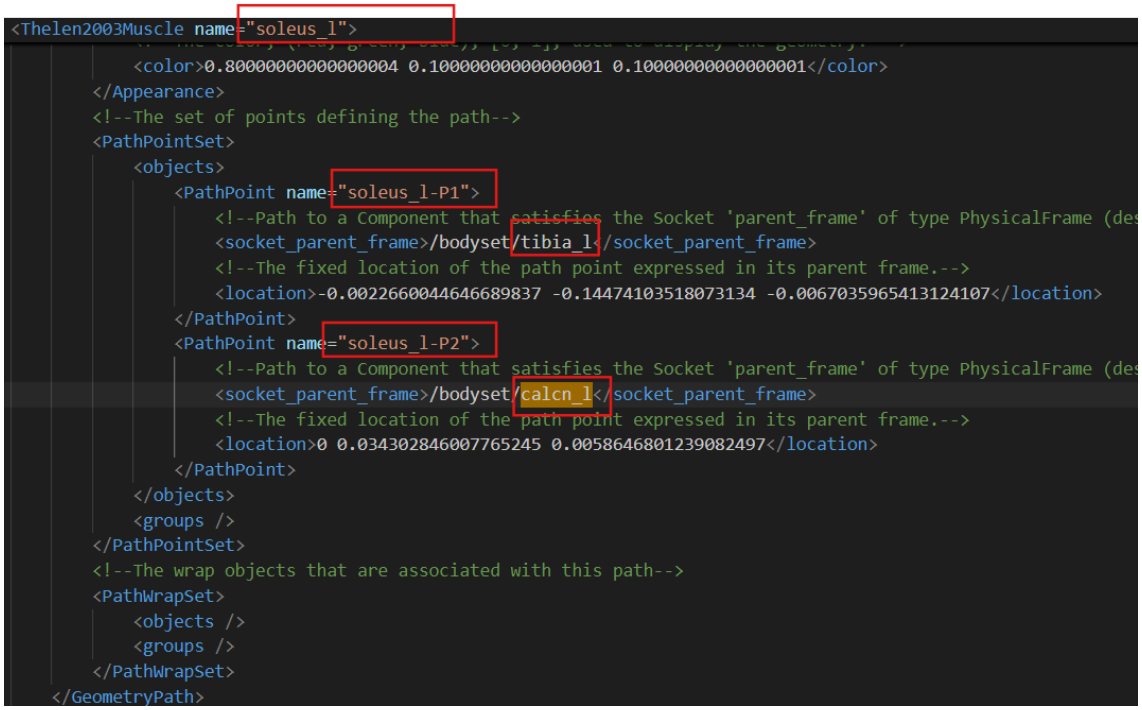


Fig. 104 Parámetros a editar dentro de un músculo

bifemlh_l: En la **Fig. 105**, se ve que este músculo tiene 3 puntos, uno en el fémur y dos en la tibia, los de la tibia se deben eliminar, sin embargo, el músculo no puede quedar con un solo punto, debe tener al menos dos, por lo cual, se eliminará el tercer punto desde las etiquetas de *PathPoint* y al segundo punto se le modificará el *socket_parent_frame* hacia la tibia derecha (**Fig. 106**), esto es una medida temporal para luego reubicarlo desde OpenSim cuando ya se tenga la nueva geometría del fémur.

```
<Thelen2003Muscle name="bifemlh_l">
        <!--The set of points defining the path-->
        <PathPointSet>
            <objects>
                <PathPoint name="bifemlh_l-P1">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/pelvis</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.13410198199644272 -0.10920006584133952 -0.073928561684923635</location>
                </PathPoint>
                <PathPoint name="bifemlh_l-P2">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/tibia_l</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.028419472661056835 -0.03399006697003476 -0.027786879748003417</location>
                </PathPoint>
                <PathPoint name="bifemlh_l-P3">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/tibia_l</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.022093543530522591 -0.053156688067026579 -0.032384980474227561</location>
                </PathPoint>
            </objects>
            <groups />
        </PathPointSet>
```

Fig. 105 Parámetros a editar dentro de un músculo

```
<Thelen2003Muscle name="bifemlh_l">
            <!--Flag indicating whether the associated Geometry is visible or hidden.-->
            <visible>true</visible>
            <!--The color, (red, green, blue), [0, 1], used to display the geometry. -->
            <color>0.80000000000000004 0.10000000000000001 0.10000000000000001</color>
        </Appearance>
        <!--The set of points defining the path-->
        <PathPointSet>
            <objects>
                <PathPoint name="bifemlh_l-P1">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/pelvis</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.13410198199644272 -0.10920006584133952 -0.073928561684923635</location>
                </PathPoint>
                <PathPoint name="bifemlh_l-P2">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/tibia_r</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.028419472661056835 -0.03399006697003476 -0.027786879748003417</location>
                </PathPoint>

            </objects>
            <groups />
        </PathPointSet>
        <!--The wrap objects that are associated with this path-->
        <PathWrapSet>
            <objects />
            <groups />
        </PathWrapSet>
    </GeometryPath>
```

Fig. 106 Parámetros a editar dentro de un músculo

Sar_l: En la **Fig. 107** se ve que este músculo tiene 5 puntos, uno en la pelvis, uno en el fémur y tres en la tibia, por lo tanto, solo es necesario eliminar los 3 puntos de la tibia, entre las etiquetas de *PathPoint*. Dado que quedan 2 puntos, es suficiente para construir el músculo y la ubicación del punto 2 se puede mover desde OpenSim.

```
<Thelen2003Muscle name="sar_l">
    <objects>
        <PathPoint name="sar_l-P1">
            <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
            <socket_parent_frame>/bodyset/pelvis</socket_parent_frame>
            <!--The fixed location of the path point expressed in its parent frame.-->
            <location>-0.016288983205347517 -0.0013840312527419456 -0.13222821660811512</location>
        </PathPoint>
        <PathPoint name="sar_l-P2">
            <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
            <socket_parent_frame>/bodyset/femur_l</socket_parent_frame>
            <!--The fixed location of the path point expressed in its parent frame.-->
            <location>-0.0026749434427062813 -0.31813994011920038 0.03753837297931148</location>
        </PathPoint>
        <PathPoint name="sar_l-P3">
            <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
            <socket_parent_frame>/bodyset/tibia_l</socket_parent_frame>
            <!--The fixed location of the path point expressed in its parent frame.-->
            <location>-0.0052873437508942954 -0.039560661279012678 0.037672324225121852</location>
        </PathPoint>
        <PathPoint name="sar_l-P4">
            <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
            <socket_parent_frame>/bodyset/tibia_l</socket_parent_frame>
            <!--The fixed location of the path point expressed in its parent frame.-->
            <location>0.0056650111616724603 -0.055611526237084645 0.036161654582009199</location>
        </PathPoint>
        <PathPoint name="sar_l-P5">
            <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
            <socket_parent_frame>/bodyset/tibia_l</socket_parent_frame>
            <!--The fixed location of the path point expressed in its parent frame.-->
            <location>0.022943295204773461 -0.079310156263414436 0.023793046879024331</location>
        </PathPoint>
    </objects>
```

Fig. 107 Parámetros a editar dentro de un músculo

grac_l: En la **Fig. 108** se ve que este músculo tiene 4 puntos, uno en la pelvis y tres en la tibia, se podría realizar el procedimiento aplicado al músculo *bifemlh_l,* sin embargo, el segundo punto es un *ConditionalPathPoint,* es decir, que este punto se usa para describir en qué rango de movimiento se activa el músculo cuando se está moviendo el ángulo de la rodilla, no obstante, la rodilla izquierda ya no existe debido a las modificaciones previas, por lo que el punto 2 no se puede mantener como un *ConditionalPathPoint,* por lo cual, se procede a eliminar los puntos 2 y 4, mientras que al punto 3 se le modifica el *socket_parent_frame* a la tibia derecha, como se muestra en la **Fig. 109** y se cambia su nombre a *P2*.

```
<Thelen2003Muscle name="grac_l">
            <objects>
                <PathPoint name="grac_l-P1">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/pelvis</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.078793963858024188 -0.12637269976959153 -0.029746025539699975</location>
                </PathPoint>
                <ConditionalPathPoint name="grac_l-P2">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/tibia_l</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.025086557760939543 -0.030118976009558578 0.035632920206919778</location>
                    <!--Path to a Component that satisfies the Socket 'coordinate' of type Coordinate (desc
                    <socket_coordinate>/jointset/knee_l/knee_angle_l</socket_coordinate>
                    <!--The minimum and maximum values that the coordinate can range between, for which the
                    <range>-0.436332 0.17453299999999999</range>
                </ConditionalPathPoint>
                <PathPoint name="grac_l-P3">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/tibia_l</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.01834519447854931 -0.048653004193496968 0.033801233264645673</location>
                </PathPoint>
                <PathPoint name="grac_l-P4">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/tibia_l</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>0.0056650111616724603 -0.078932488852636248 0.021527042414355345</location>
                </PathPoint>
```

Fig. 108 Parámetros a editar dentro de un músculo

```
<Thelen2003Muscle name="grac_l">
        </Appearance>
        <!--The set of points defining the path-->
        <PathPointSet>
            <objects>
                <PathPoint name="grac_l-P1">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/pelvis</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.078793963858024188 -0.12637269976959153 -0.029746025539699975</location>
                </PathPoint>
                <PathPoint name="grac_l-P2">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/tibia_r</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.01834519447854931 -0.048653004193496968 0.033801233264645673</location>
                </PathPoint>

            </objects>
            <groups />
        </PathPointSet>
        <!--The wrap objects that are associated with this path-->
        <PathWrapSet>
            <objects />
```

Fig. 109 Parámetros a editar dentro de un músculo

rect_fem_l: Como se presenta en la **Fig. 110**, este músculo tiene 3 puntos, sin embargo, uno es un *ConditionalPathPoint* y otro es un *MovingPathPoint,* ambos asociados a la rodilla, por lo que se deben eliminar estas condiciones, para ello se elimina el punto 3 (*MovingPathPoint*), mientras que al punto 2 se le modifican las etiquetas de *CondicionalPathPoint* por etiquetas de *PathPoint* y se elimina el *socket_coordinate* y el *range*, como se ve en la **Fig. 111.**

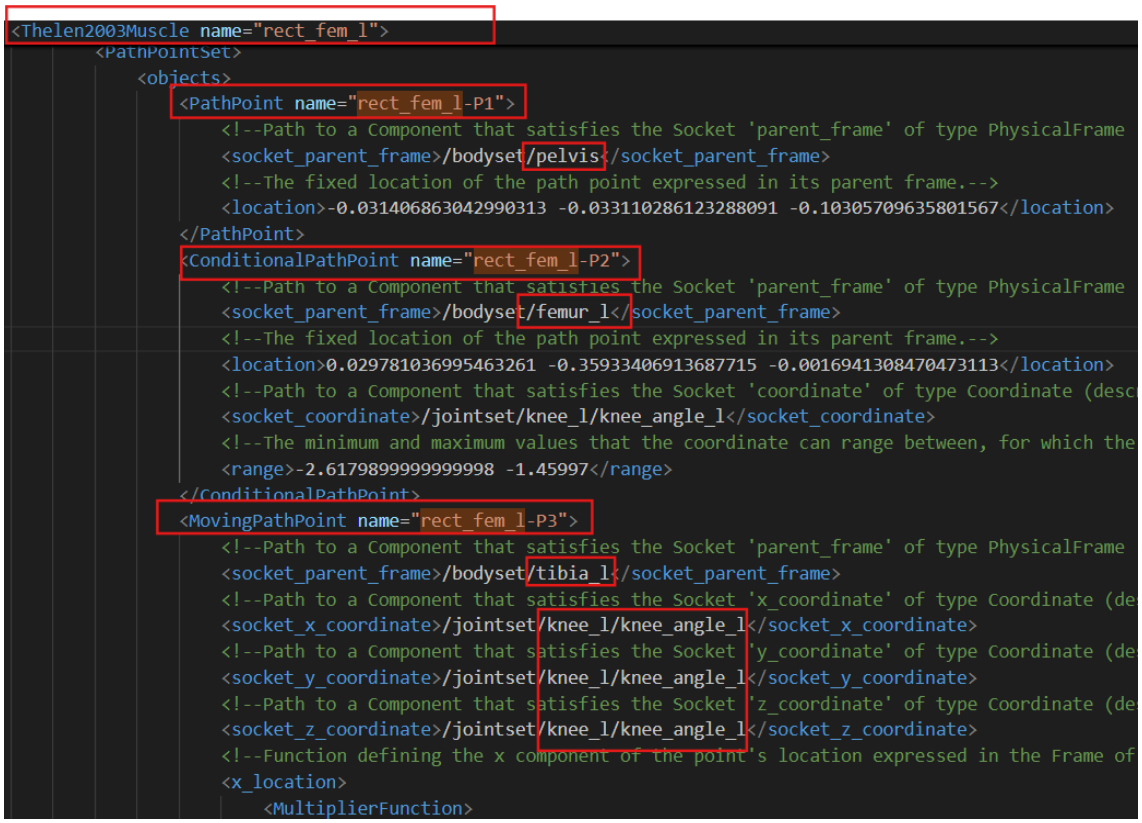


Fig. 110 Parámetros a editar dentro de un músculo

```
<Thelen2003Muscle name="rect_fem_l">
            <visible>true</visible>
            <!--The color, (red, green, blue), [0, 1], used to display the geometry. -->
            <color>0.8000000000000004 0.1000000000000001 0.1000000000000001</color>
        </Appearance>
        <!--The set of points defining the path-->
        <PathPointSet>
            <objects>
                <PathPoint name="rect_fem_l-P1">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/pelvis</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>-0.031406863042990313 -0.033110286123288091 -0.10305709635801567</location>
                </PathPoint>
                <PathPoint name="rect_fem_l-P2">
                    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
                    <socket_parent_frame>/bodyset/femur_l</socket_parent_frame>
                    <!--The fixed location of the path point expressed in its parent frame.-->
                    <location>0.029781036995463261 -0.35933406913687715 -0.0016941308470473113</location>

                </PathPoint>

            </objects>
            <groups />
        </PathPointSet>
        <!--The wrap objects that are associated with this path-->
        <PathWrapSet>
            <objects />
            <groups />
        </PathWrapSet>
```

Fig. 111 Parámetros a editar dentro de un músculo

Con los ejemplos anteriores se presentan los posibles casos de modificación para los músculos, por lo que a partir de ahí es cuestión de buscar en la sección de *ForceSet* las ocurrencias de los cuerpos óseos eliminados anteriormente y proceder a eliminar bien sea el músculo completo, algún punto dentro de este o modificar el *socket_parent_frame* de un punto.

En caso de que usted haya empleado como base el modelo de la carpeta Gait2354, y haya realizado una amputación transfemoral izquierda, la **TABLA VI** presenta una lista de los músculos eliminados o modificados, según el caso.

TABLA VI
Lista de músculos eliminados y modificados

Músculo	Operación
soleus_l	Eliminación completa
Tib_post_l	
Tib_ant_l	
med_gas_l	

bifemlh_l	Eliminación y edición de puntos en el código. Reinserción de puntos desde GUI de OpenSim
bifemsh_l	
sar_l	
tfl_l	
grac_l	
rect_fem_l	
vast_int_l	

Adicionalmente, se deben eliminar o modificar aquellos grupos musculares que contengan músculos que ya no existan en su modelo, como se presenta en la **Fig. 112.**

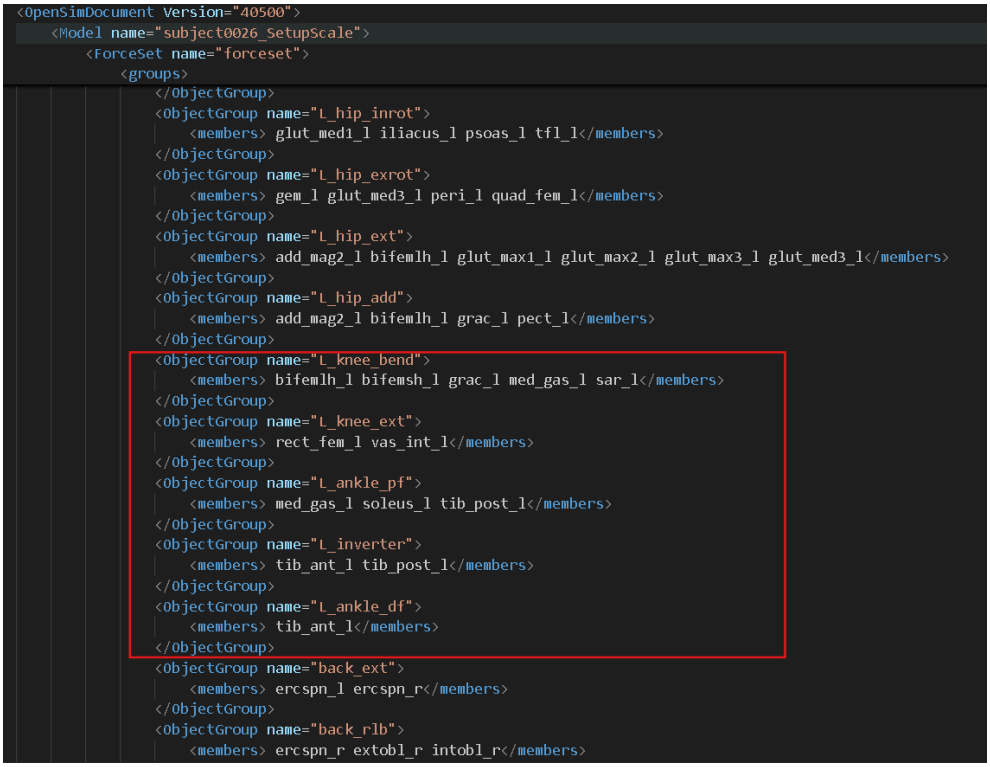


Fig. 112 Modificación de grupos musculares

9.1.3. Marcadores

Si el modelo tiene marcadores, entonces es necesario eliminar aquellos ubicados en cuerpos óseos ya eliminados, para esto se debe realizar una búsqueda en la sección de *MarkerSet*, aquellos marcadores cuyo *socket_parent_frame* sea un cuerpo que ya no existe se deben eliminar para posicionarse adecuadamente una vez se tenga la nueva geometría.

En la **Fig. 113** se muestra un ejemplo con un marcador ubicado en *L.Midfoot.Lat,* es necesario eliminar todo el marcador, es decir desde la etiqueta de inicio hasta la de cierre (*<Marker>…</Marker>*).

```
<Marker name="L.Midfoot.Lat">
    <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame
    <socket_parent_frame>/bodyset/calcn_l</socket_parent_frame>
    <!--The fixed location of the station expressed in its parent frame.-->
    <location>0.11501332883639784 -0.062298740455945856 -0.03653817596855255</location>
    <!--Flag (true or false) specifying whether the marker is fixed in its parent frame du
    <fixed>false</fixed>
</Marker>
```

Fig. 113 Modificación de marcadores dentro del modelo

Para verificar que las modificaciones anteriores se realizaron correctamente debe de ser posible abrir el modelo en OpenSim sin que se generen errores, de lo contrario es necesario verificar de nuevo que no haya quedado ningún músculo con un solo *PathPoint* y que los nombres de los cuerpos eliminados no aparezcan en ninguna parte del código. En la **Fig. 114** se muestra cómo queda el modelo al abrirse en OpenSim.

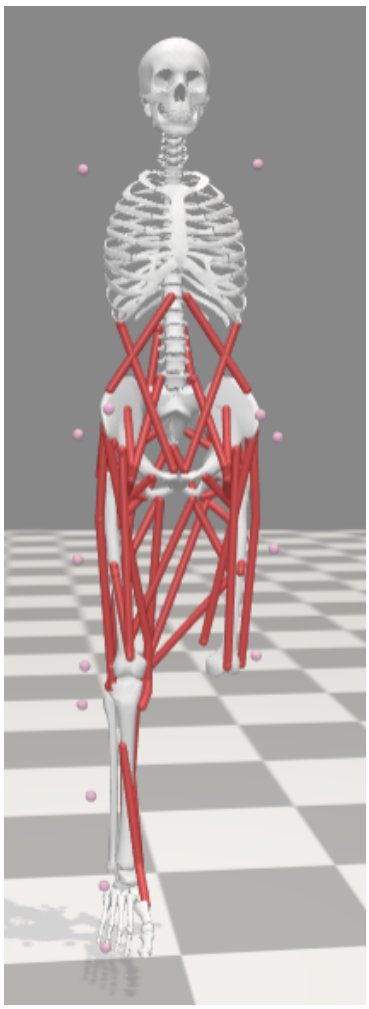


Fig. 114 Visualización del modelo amputado

9.2. Modificación femoral

Una amputación transfemoral remueve la parte distal del fémur, por lo cual, se debe modificar la geometría del modelo de OpenSim para obtener un hueso con la longitud de la amputación. Las geometrías óseas en OpenSim se encuentran en formato .vtp (Visualization Toolkit Polygonal), los cuales son archivos utilizados para almacenar datos poligonales para visualización y procesamiento de gráficos 3D. Sin embargo, no son archivos editables en un software CAD como Fusion360 o SolidWorks, por lo que hay que convertir el formato, para lo cual se usarán dos softwares libres, ParaView y MeshLab.

9.2.1. ParaView

Para recortar la longitud del fémur lo ideal es editar directamente el fémur que proporciona OpenSim, dado que su ubicación en el modelo ya está definida y si no se

modifica la ubicación espacial de esta geometría, su reubicación en el modelo será más sencilla. Por lo cual, lo primero será descargar ParaView, una vez se tenga el software instalado se puede abrir la geometría del fémur. Para abrir el archivo en el software se debe seguir la siguiente ruta *File* → *Open* (**Fig. 115**), luego se abrirá el buscador de archivos del dispositivo y para encontrar el fémur de OpenSim hay que dirigirse a la carpeta de *Geometry* que está en el directorio donde se instala OpenSim como se ve en la **Fig. 116**.

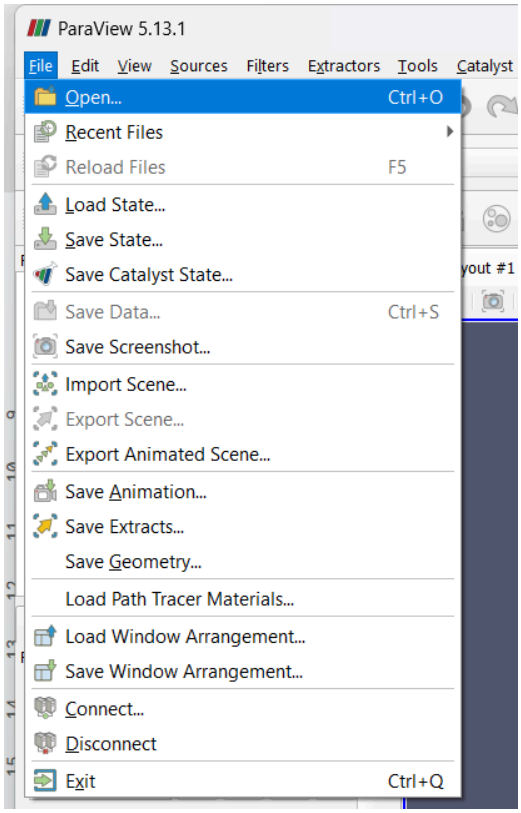


Fig. 115 Abrir archivos en ParaView

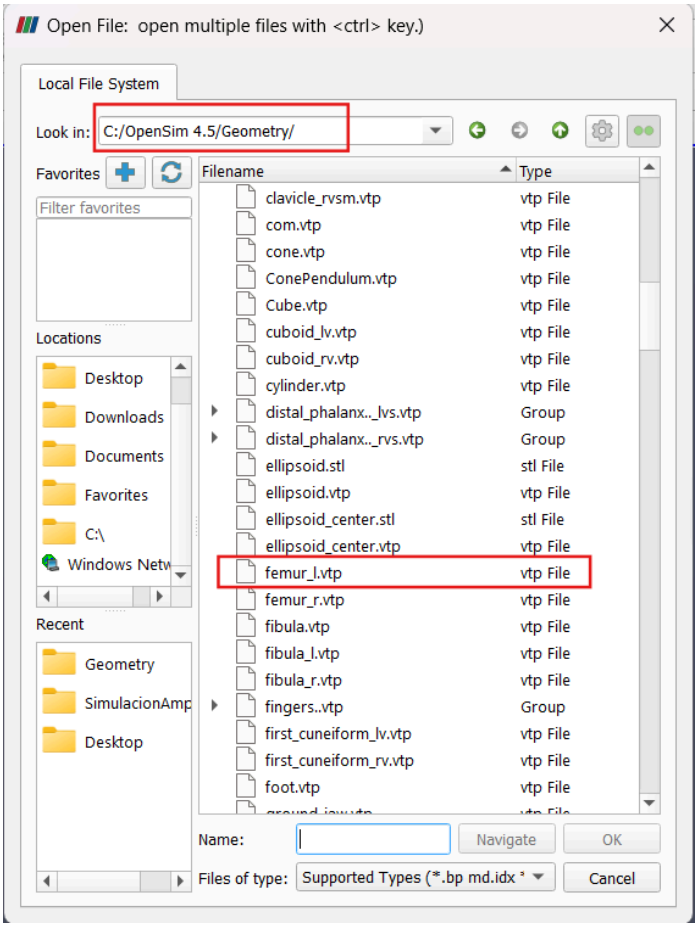


Fig. 116 Búsqueda de geometría propia de OpenSim

Una vez se abra el archivo, este se puede visualizar activando la opción marcada en la **Fig. 117**, en la ventana de *Pipeline Browser*

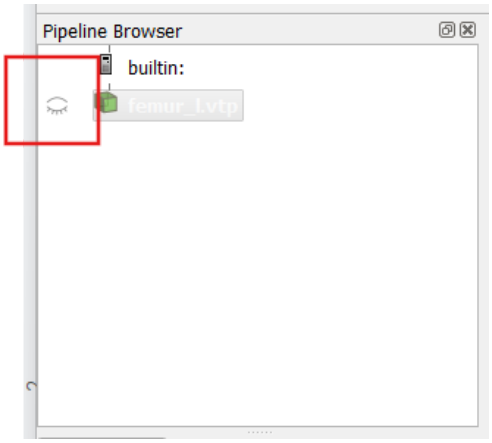
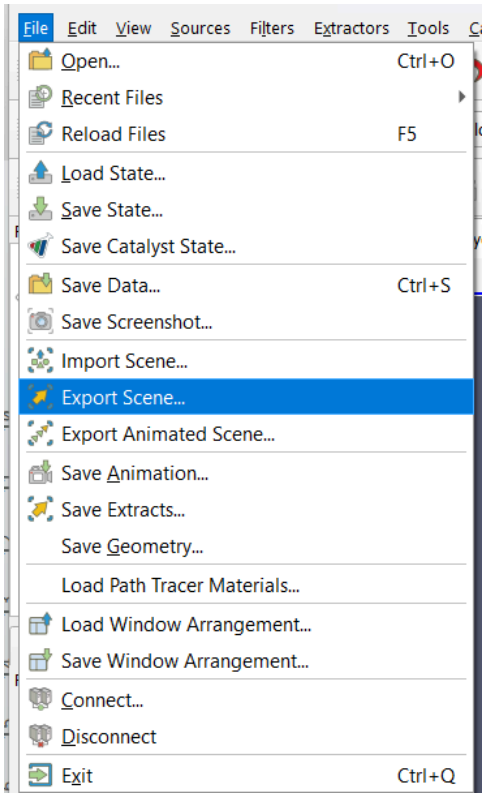
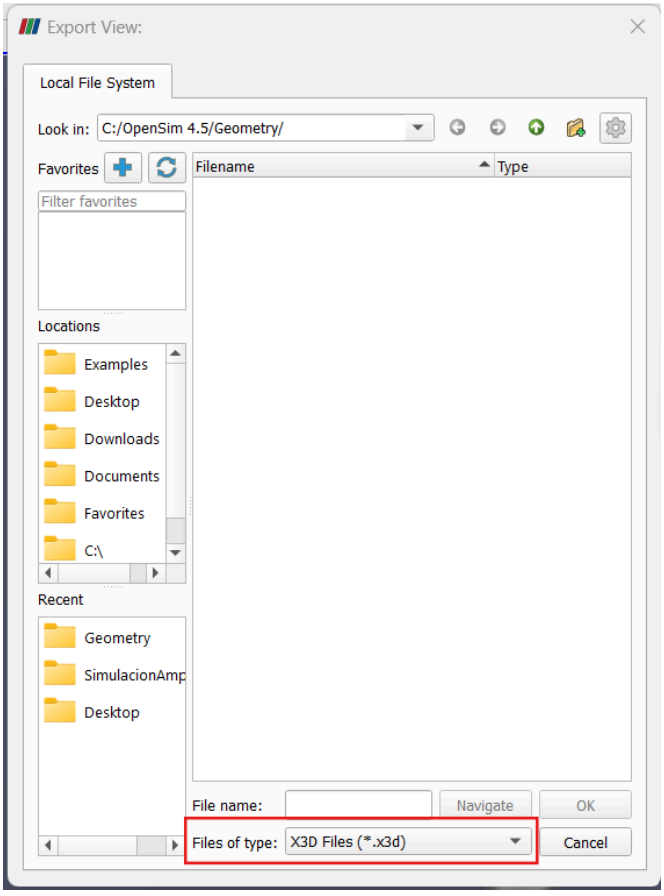


Fig. 117 Visualización de geometría en ParaView

Posteriormente solo es necesario exportar la geometría en un nuevo formato, para esto se sigue la ruta *File → Export Scene…,* en la ventana emergente seleccionar en *File Type* la opción *.x3d,* como se ve en la **Fig. 118.**



a



b

Fig. 118 Exportar geometría desde ParaView

9.2.2. MeshLab

El archivo que se exportó de ParaView, se puede abrir usando MeshLab, este software permite editar y procesar mallas 3D. Partiendo de tener MeshLab instalado, se debe importar la malla en el software, la ruta para abrir el archivo es *File → Import Mesh…* **(Fig. 119)**, esto despliega el buscador de archivos del dispositivo, y allí se busca el archivo exportado de ParaView, una vez abierto solo es necesario exportar la geometría como un *.stl*, que es un formato de malla soportado por diversos software CAD, para esto la ruta es *File → Export Mesh…*, en la ventana emergente seleccionar como tipo de archivo *.stl* **(Fig. 120)**

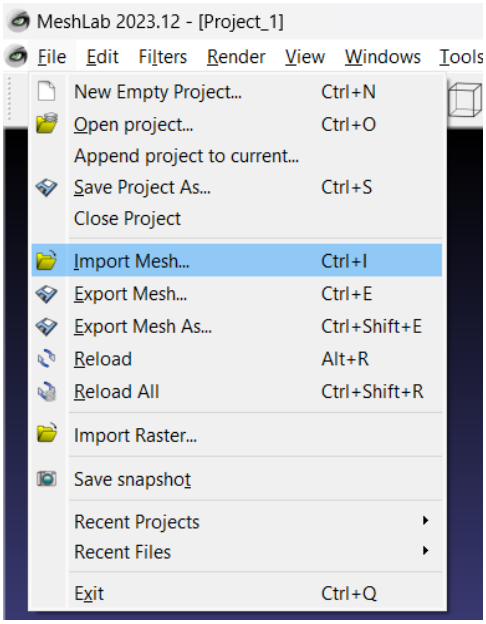


Fig. 119 Abrir archivos con MeshLab

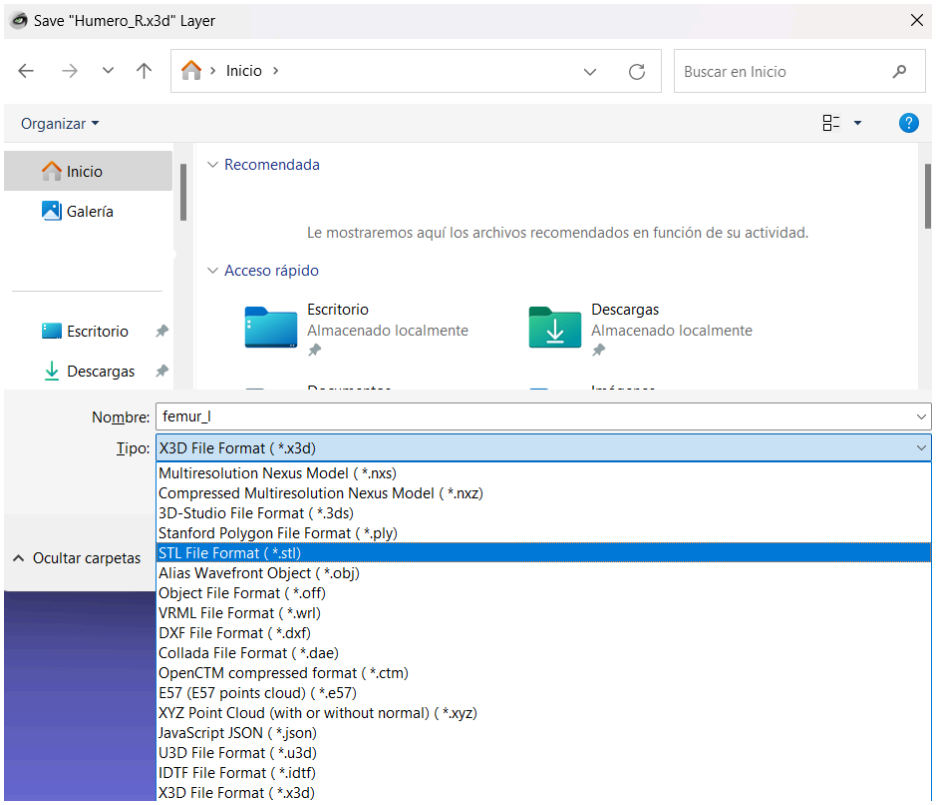


Fig. 120 Exportar archivos desde MeshLab

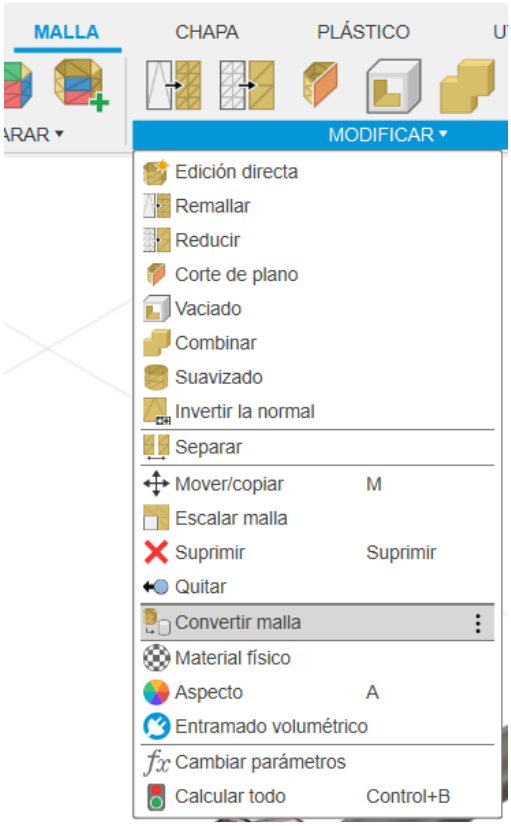
9.2.3. Software CAD

Una vez se cuenta con la geometría como .slt se puede realizar el corte para modificar la longitud ósea, en este caso se usará Fusion360, sin embargo, se puede seleccionar el software CAD de su preferencia, solo se deben tener en cuenta algunas consideraciones. Al importar la geometría en el software no se debe modificar su ubicación espacial, es decir, no se deben realizar traslaciones ni rotaciones, dado que esto representa una modificación en el modelo de OpenSim y es más complejo reposicionarlo.

Si las modificaciones se realizan en Fusion360 el proceso inicia insertando la malla (**Fig. 121a**), luego es necesario convertir la malla en un cuerpo, para esto la ruta es *Malla* → *Modificar* → *Convertir malla* (**Fig. 121b**), con la ventana emergente se selecciona el cuerpo, es decir, la malla, luego la operación y el método (**Fig. 121c**).



a



a



c

Fig. 121 Insertar malla en Fusion 360 y convertir en cuerpo

Con el nuevo cuerpo es posible modificar la longitud del hueso, para esto es suficiente con hacer un plano de desfase a la distancia en la que se realizará el corte, luego con un sketch de un círculo en este plano, cuyo diámetro sea mayor al del hueso, se realiza una extrusión de tipo corte de este para eliminar la porción del hueso deseada. Finalmente, para exportar el nuevo fémur se debe ir al ícono mostrado en la **Fig. 122** y dar clic en exportar, en tipo de archivo seleccionar .stl, dado que este formato se puede usar en OpenSim como geometría. Ahora bien, la ubicación del archivo debe ser la misma carpeta que contiene el modelo de OpenSim con el que se está trabajando, para que el software lo pueda identificar.

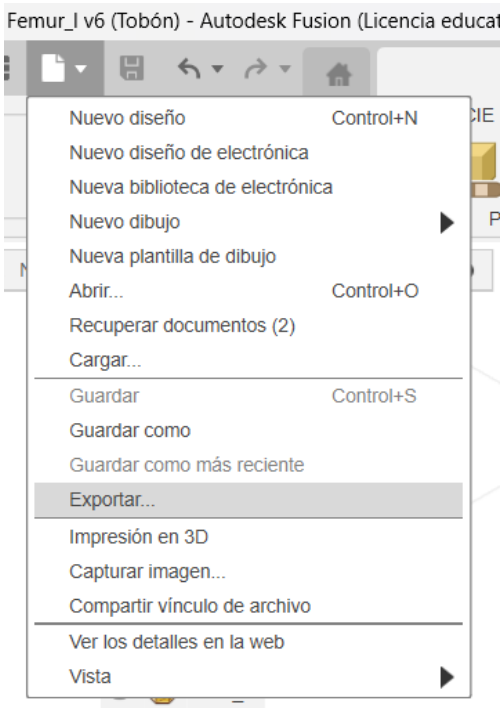


Fig. 122 Exportar geometría desde Fusion 360

9.2.4. Reubicación en OpenSim

Una vez se cuenta con el segmento modificado se puede reposicionar en OpenSim, para esto hay que buscar en la sección de *BodySet* el cuerpo modificado, en este caso el fémur izquierdo, dentro de Mesh se debe de modificar el archivo que está dentro de las etiquetas *<mesh_file> </mesh_file>*, por el nuevo archivo .stl obtenido del software CAD, como se ve en la **Fig. 123**.

```
<Body name="femur_l">
    <!--The geometry used to display the axes of this Frame.-->
    <FrameGeometry name="frame_geometry">···
    </FrameGeometry>
    <!--List of geometry attached to this Frame. Note, the geometry are treated as fixed to the frame a
    <attached_geometry>
        <Mesh name="femur_l_geom_1">
            <!--Path to a Component that satisfies the Socket 'frame' of type Frame.-->
            <socket_frame>..</socket_frame>
            <!--Scale factors in X, Y, Z directions respectively.-->
            <scale_factors>0.89164781423542716 0.89164781423542716 0.89164781423542716</scale_factors>
            <!--Default appearance attributes for this Geometry-->
            <Appearance>
                <!--The opacity used to display the geometry between 0:transparent, 1:opaque.-->
                <opacity>1</opacity>
                <!--The color, (red, green, blue), [0, 1], used to display the geometry. -->
                <color>1 1 1</color>
            </Appearance>
            <!--Name of geometry file.-->
            <mesh_file>femur_l.vtp</mesh_file>
        </Mesh>
    </attached_geometry>
    <!--Set of wrap objects fixed to this body that GeometryPaths can wrap over.This property used to b
    <WrapObjectSet name="wrapobjectset">···
    </WrapObjectSet>
```

a

```
<Body name="femur_l">
    <!--The geometry used to display the axes of this Frame.-->
    <FrameGeometry name="frame_geometry">···
    </FrameGeometry>
    <!--List of geometry attached to this Frame. Note, the geometry are treated as fixed to the frame a
    <attached_geometry>
        <Mesh name="femur_l_geom_1">
            <!--Path to a Component that satisfies the Socket 'frame' of type Frame.-->
            <socket_frame>..</socket_frame>
            <!--Scale factors in X, Y, Z directions respectively.-->
            <scale_factors>0.89164781423542716 0.89164781423542716 0.89164781423542716</scale_factors>
            <!--Default appearance attributes for this Geometry-->
            <Appearance>
                <!--The opacity used to display the geometry between 0:transparent, 1:opaque.-->
                <opacity>1</opacity>
                <!--The color, (red, green, blue), [0, 1], used to display the geometry. -->
                <color>1 1 1</color>
            </Appearance>
            <!--Name of geometry file.-->
            <mesh_file>femur_l_cut.stl</mesh_file>
        </Mesh>
    </attached_geometry>
```

b

Fig. 123 Parámetros a modificar para cargar nueva geometría en OpenSim

Después de cambiar el archivo de geometría es posible volver a abrir el modelo de OpenSim para revisar los cambios realizados, como se ve en la **Fig. 124**, a partir de este punto es necesario reposicionar la inserción de los músculos, no solo los que se movieron a la pierna derecha, sino todos aquellos que se encuentren sobre el fémur y que se deben reubicar para generar movimiento.

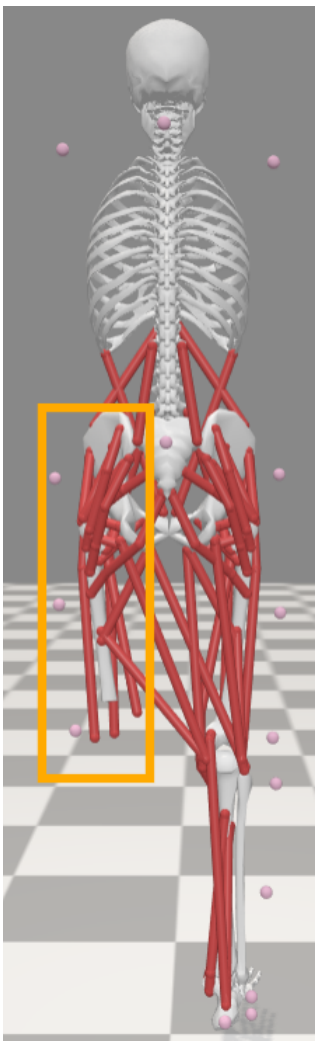


Fig. 124 Visualización de modificaciones óseas y musculares en el modelo

9.3. Modificación muscular

Para mover un punto de inserción muscular se debe desplegar todo el conjunto de músculos en la ventana *Navigator*, *Forces* → *Muscles* → *all*, allí se puede dar doble clic sobre un músculo para visualizarlo, una vez se tenga seleccionado el músculo que se desea reubicar se crea un control en el punto distal del músculo dando clic en la prominencia de la inserción como se ve marcado en amarillo en la **Fig. 125.**

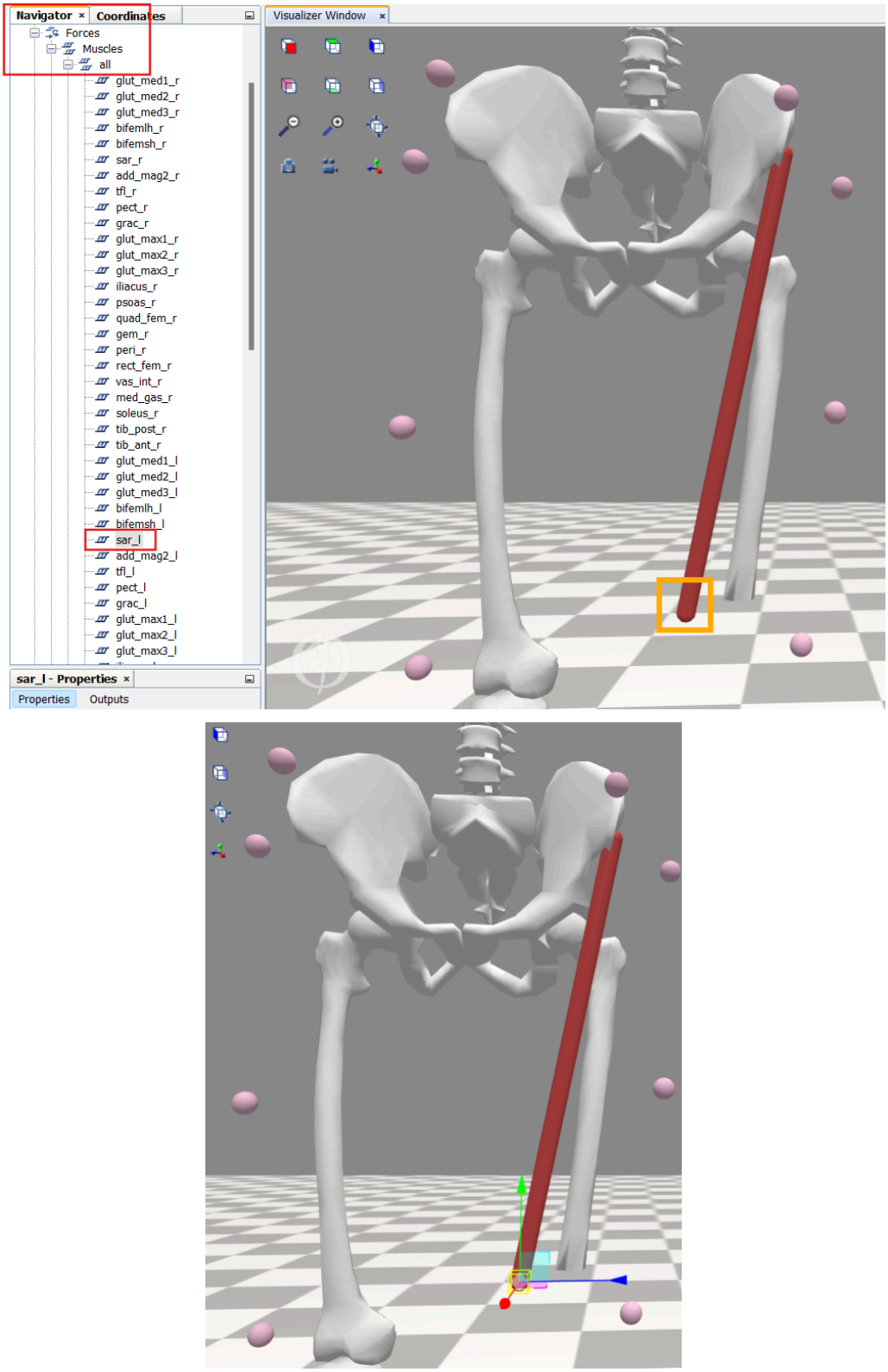


Fig. 125 Modificación de inserción muscular desde OpenSim

Para mover el punto distal se usan las guías del control ubicado en la inserción muscular, para reubicarlo se pueden usar guías quirúrgicas para conocer en qué sitio óseo se suele ubicar cada músculo luego de una amputación, este procedimiento se debe realizar con todos los músculos que quedaron sobre el miembro amputado, una vez se tengan todos los músculos ubicados adecuadamente se puede volver a activar la visualización del conjunto completo usando la siguiente ruta clic derecho sobre *Muscles* → *Display* → *Show* (**Fig. 126**), se debe guardar el archivo para poder ver los cambios reflejados en el código del modelo y continuar con el resto de procesos, para guardar el modelo la ruta es *File* → *Save Model,* como se ve en la **Fig. 127.**

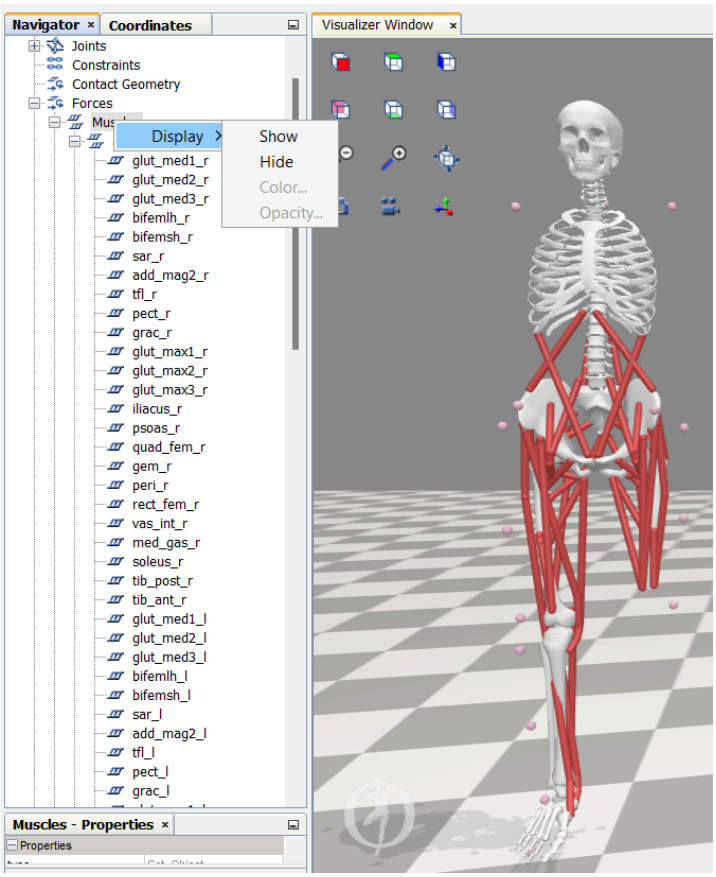


Fig. 126 Activación y desactivación de visualización muscular

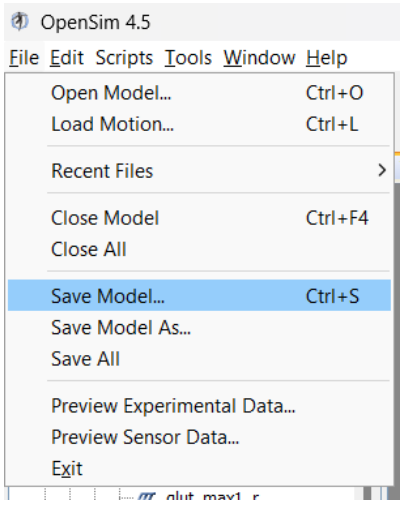


Fig. 127 Guardar un modelo de OpenSim

9.4. Carga de geometría protésica

Para cargar la geometría protésica en OpenSim hay que crear diferentes secciones en el código, sin embargo, el primer punto es posicionar la geometría adecuadamente en un software CAD, el propósito de esta guía no es explicar cómo construir una prótesis, dado que el modelo de esta es individual y depende de los objetivos de cada estudio, no obstante, la ubicación espacial de cada elemento si es relevante.

9.4.1. Ubicación espacial de componentes

En el software CAD de su preferencia es necesario ordenar los componentes de la prótesis con relación al fémur dado que este está ubicado correctamente en el modelo, por lo cual, partiendo de abrir la malla del cuerpo óseo editado, sin realizar traslaciones ni rotaciones, se deben posicionar en el plano 3D los componentes de la prótesis realizando las transformaciones necesarias para que encajen adecuadamente, como se ve en la **Fig. 128**. Posteriormente la prótesis no se puede exportar como un único cuerpo, dado que las articulaciones no tendrían movimiento en OpenSim, sino que se debe exportar cada componente por separado. Una prótesis transfemoral consta de 4 componentes de forma simplificada, encaje, rodilla, vástago y pie, como se ve en la **Fig. 129**, sin embargo, la rodilla está fija al encaje y presenta articulación con el vástago, por lo cual son tres los componentes a exportar, pie, vástago y rodilla con encaje, estas geometrías se exportan en formato .stl o .obj y se deben guardar en la misma carpeta donde está el modelo que se está editando. Posteriormente en OpenSim se crearán tres cuerpos y tres articulaciones para unir estos tres cuerpos tanto entre ellos como con el modelo.

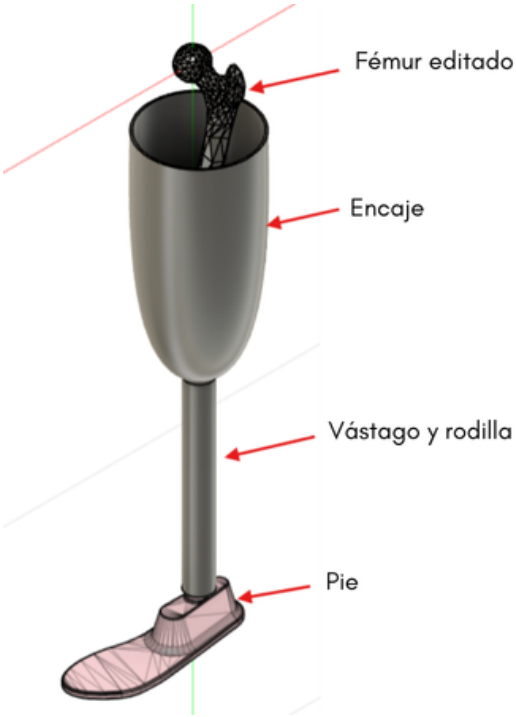


Fig. 128 Posicionamiento de componentes protésicos



Fig. 129 Componentes de una prótesis transfemoral
Nota. fuente: https://www.ottobock.com/en-ca/product/4R11

9.4.2. Creación de cuerpos

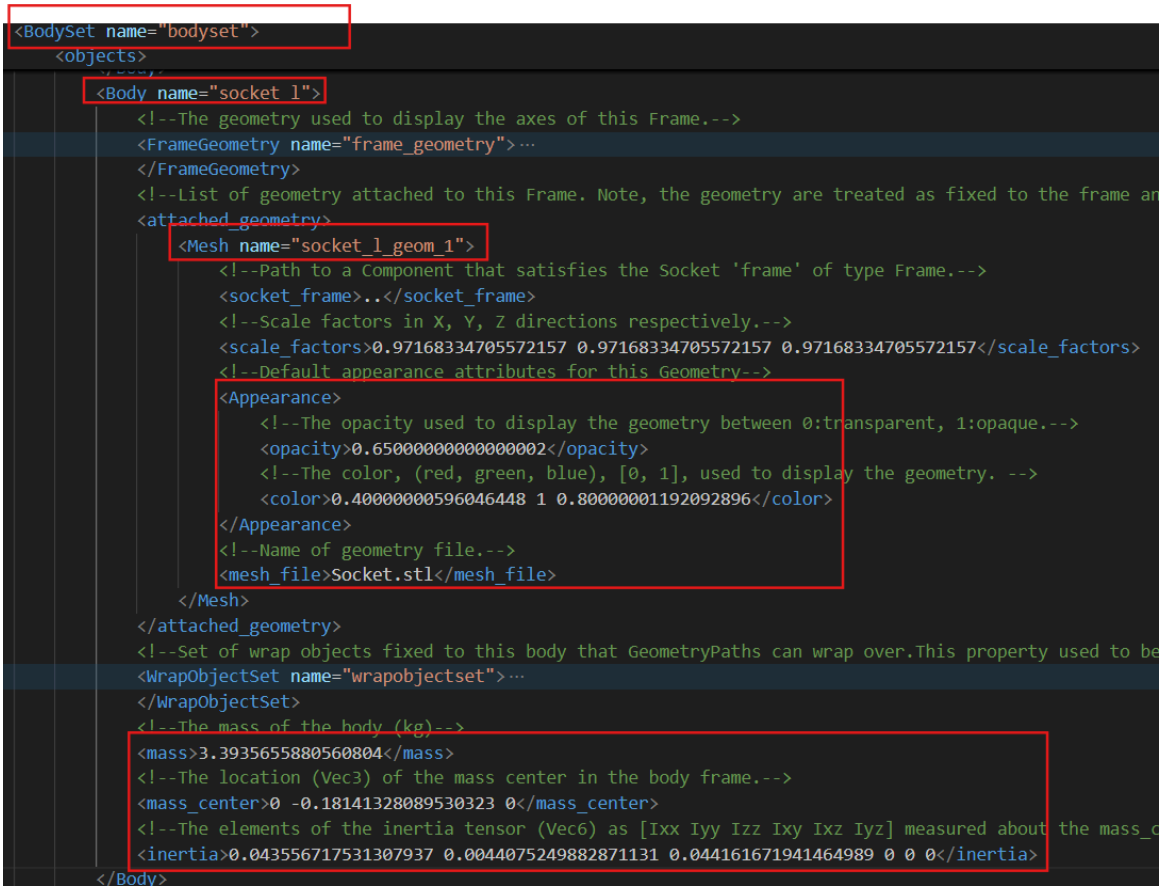
Para cargar las geometrías es necesario crear un cuerpo para cada una, para esto se debe desplegar la sección de *BodySet* en el código del modelo, la ubicación de los cuerpos dentro de esta parte del código no es relevante, no obstante, para ser ordenado se pueden posicionar debajo del fémur izquierdo que es el miembro que se va a reconstruir, debajo la etiqueta de cierre (*</Body>*), del cuerpo *femur_l,* copie y pegue este mismo cuerpo para editar los siguientes aspectos:

- Nombre → Sustituir el nombre de *femur_l* por el del nuevo cuerpo, en este caso el encaje (*socket_l*).
- Nombre del Mesh → Cambiar el prefijo de *femur_l_geom_1* por el del cuerpo nuevo, en este caso (*socket_l_geom_1*).
- Color y Opacidad → Se puede mantener el color del hueso o modificar el valor de color y opacidad por el que se desee.
- Mesh File → Cambiar el nombre del archivo de geometría por el del nuevo cuerpo, en este caso el encaje (*socket.stl*).
- Masa → Agregar el valor en kilogramos del nuevo cuerpo, esto depende de la geometría creada, se puede obtener el valor desde el software CAD o desde datos experimentales, dependiendo del origen de la parte protésica a importar.
- Centro de masa → La ubicación del vector del centro de masa se coloca en este punto.
- Inercia → Cambiar las magnitudes del tensor de inercia, es decir, las componentes *Ixx*, *Iyy*, *Izz*, *Ixy*, *Ixz* y *Iyz*, estas se deben medir alrededor del centro de masa y no del origen del cuerpo.

En la **Fig. 130a** se marcan en rojo las secciones a modificar, mientras que en la **Fig. 130b** se muestran los cambios efectuados para cargar el cuerpo del encaje. Los valores de masa, inercia y centro de masa dependen de la prótesis específica que se emplee en cada estudio y son datos que cambian con cada geometría. Si el objetivo del estudio es diseñar dónde se deberían ubicar o qué valores son ideales en estas características para evaluar los efectos sobre la marcha del paciente, se debe partir de un valor inicial y modificarse iterativamente hasta obtener los resultados esperados.

```
<Body name="femur_l">
    <!--The geometry used to display the axes of this Frame.-->
    <FrameGeometry name="frame_geometry">…
    </FrameGeometry>
    <!--List of geometry attached to this Frame. Note, the geometry are treated as fixed to the frame a
    <attached_geometry>
        <Mesh name="femur_l_geom_1">
            <!--Path to a Component that satisfies the Socket 'frame' of type Frame.-->
            <socket_frame>..</socket_frame>
            <!--Scale factors in X, Y, Z directions respectively.-->
            <scale_factors>0.89164781423542716 0.89164781423542716 0.89164781423542716</scale_factors>
            <!--Default appearance attributes for this Geometry-->
            <Appearance>
                <!--The opacity used to display the geometry between 0:transparent, 1:opaque.-->
                <opacity>1</opacity>
                <!--The color, (red, green, blue), [0, 1], used to display the geometry. -->
                <color>1 1 1</color>
            </Appearance>
            <!--Name of geometry file -->
            <mesh_file>femur_l_cut.stl</mesh_file>
        </Mesh>
    </attached_geometry>
    <!--Set of wrap objects fixed to this body that GeometryPaths can wrap over.This property used to b
    <WrapObjectSet name="wrapobjectset">…
    </WrapObjectSet>
    <!--The mass of the body (kg)-->
    <mass>8.9840382307628861</mass>
    <!--The location (Vec3) of the mass center in the body frame.-->
    <mass_center>0 -0.15158012842002264 0</mass_center>
    <!--The elements of the inertia tensor (Vec6) as [Ixx Iyy Izz Ixy Ixz Iyz] measured about the mass_
    <inertia>0.10282306506322923 0.026953618997157182 0.10842880348713939 0 0 0</inertia>
</Body>
```

a



b

Fig. 130 Parámetros a editar dentro de un cuerpo modificado en un modelo

Este proceso se debe de repetir con las tres geometrías que componen la prótesis, encaje y rodilla, vástago y pie, es decir, que se deben crear tres cuerpos nuevos, como se muestra en la **Fig. 131**, una vez se tengan todos los cuerpos nuevos es necesario crear articulaciones entre estos para poder generar movimiento.

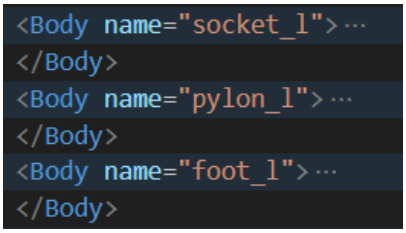


Fig. 131 Cuerpos nuevos dentro del modelo

9.4.3. Creación de articulaciones

Los cuerpos que componen un modelo de OpenSim deben de relacionarse entre sí por medio de articulaciones, por lo cual, dado que en la subsección anterior se crearon tres

cuerpos, se deben de crear tres articulaciones para unirse tanto entre ellos como con el modelo.

OpenSim tiene siete tipos de articulaciones, sin embargo, para los propósitos de esta guía se explicarán 2 tipos.

- **Weld Joint**

  Este tipo de articulación no produce movimiento, su propósito como tal es unir dos cuerpos rígidos sin permitir movimiento relativo entre ellos, el encaje protésico es un cuerpo que se coloca sobre el miembro residual para unir el resto de la prótesis al usuario, por lo cual no hay movimiento entre este y el muñón, es por esto que *WeldJoint* es el tipo de articulación adecuada para unir el fémur residual con el encaje. Como se mencionó anteriormente OpenSim tiene una estructura de Padre-Hijo, para definir la estructura jerárquica del modelo, en este caso el cuerpo Padre será el fémur, mientras que el Hijo será el encaje.

  Para crear un *WeldJoint* es necesario desplegar la sesión *JointSet* del código del modelo, y debajo de la articulación *hip_l* se puede insertar el siguiente código, para mantener el órden del archivo.

```
            <WeldJoint name="socket_l_femur">
               <!--Path to a Component that satisfies the Socket 'parent_frame' of type PhysicalFrame (description: The parent frame for the joint.).-->
               <socket_parent_frame>femur_l_offset</socket_parent_frame>
               <!--Path to a Component that satisfies the Socket 'child_frame' of type PhysicalFrame (description: The child frame for the joint.).-->
               <socket_child_frame>socket_l_offset</socket_child_frame>
               <!--Physical offset frames owned by the Joint that are typically used to satisfy the owning Joint's parent and child frame connections (sockets). PhysicalOffsetFrames are often used to describe the fixed transformation from a Body's origin to another location of interest on the Body (e.g., the joint center). When the joint is deleted, so are the PhysicalOffsetFrame components in this list.-->
               <frames>
                  <PhysicalOffsetFrame name="femur_l_offset">
                     <!--The geometry used to display the axes of this Frame.-->
                     <FrameGeometry name="frame_geometry">
                        <!--Path to a Component that satisfies the Socket 'frame' of type Frame.-->
                        <socket_frame>..</socket_frame>
                        <!--Scale factors in X, Y, Z directions respectively.-->
                        <scale_factors>0.20000000000000001 0.20000000000000001 0.20000000000000001</scale_factors>
                     </FrameGeometry>
                     <!--Path to a Component that satisfies the Socket 'parent' of type C (description: The parent frame to this frame.).-->
                     <socket_parent>/bodyset/femur_l</socket_parent>
                     <!--Translational offset (in meters) of this frame's origin from the parent frame's origin, expressed in the parent frame.-->
```

```
                    <translation>0 0 0</translation>
                      <!--Orientation offset (in radians) of this frame in its parent frame, expressed as a
frame-fixed x-y-z rotation sequence.-->
                    <orientation>0 0 0</orientation>
                  </PhysicalOffsetFrame>
                  <PhysicalOffsetFrame name="socket_l_offset">
                    <!--The geometry used to display the axes of this Frame.-->
                    <FrameGeometry name="frame_geometry">
                      <!--Path to a Component that satisfies the Socket 'frame' of type Frame.-->
                      <socket_frame>..</socket_frame>
                      <!--Scale factors in X, Y, Z directions respectively.-->
                                    <scale_factors>0.20000000000000001  0.20000000000000001
0.20000000000000001</scale_factors>
                    </FrameGeometry>
                      <!--Path to a Component that satisfies the Socket 'parent' of type C (description:
The parent frame to this frame.).-->
                    <socket_parent>/bodyset/socket_l</socket_parent>
                      <!--Translational offset (in meters) of this frame's origin from the parent frame's
origin, expressed in the parent frame.-->
                    <translation>0 0 0</translation>
                      <!--Orientation offset (in radians) of this frame in its parent frame, expressed as a
frame-fixed x-y-z rotation sequence.-->
                    <orientation>0 0 0</orientation>
                  </PhysicalOffsetFrame>
                </frames>
              </WeldJoint>
```

Como se marca en la **Fig. 132,** es necesario hacer las siguientes modificaciones para que la articulación quede configurada adecuadamente:

- *WeldJoint name*: Escriba el nombre que se asignará a la articulación.
- *socket_parent_frame:* Escriba el nombre del cuerpo padre, en este caso fémur izquierdo, tal cual nombró el cuerpo en la sección *BodySet.*
- *socket_child_frame:* Escriba el nombre del cuerpo hijo, en este caso el encaje, tal cual nombró el cuerpo en la sección *BodySet.*
- *PhysicalOffsetFrame name="femur_l_offset:* Cada cuerpo de la articulación debe tener un Marco (*Frame*) para que se escale y ajuste adecuadamente en el modelo, el primer marco será el del cuerpo Padre y el segundo el del cuerpo hijo, sustituya el prefijo *femur_l* por el nombre del cuerpo padre de su modelo, pero mantenga el sufijo *_offset*.
- *<socket_parent>/bodyset/femur_l</socket_parent>*: En esta sección reemplace *femur_l* por el nombre del cuerpo padre de su modelo.
- *<PhysicalOffsetFrame name="socket_l_offset"*: El segundo marco es el del cuerpo hijo, sustituya el prefijo *socket_l* por el nombre del cuerpo hijo de su modelo, pero mantenga el sufijo *_offset*.

- *<socket_parent>/bodyset/socket_l</socket_parent>:* En esta sección reemplace *socket_l* por el nombre del cuerpo hijo de su modelo.



Fig. 132 Parámetros a editar dentro de una articulación del modelo

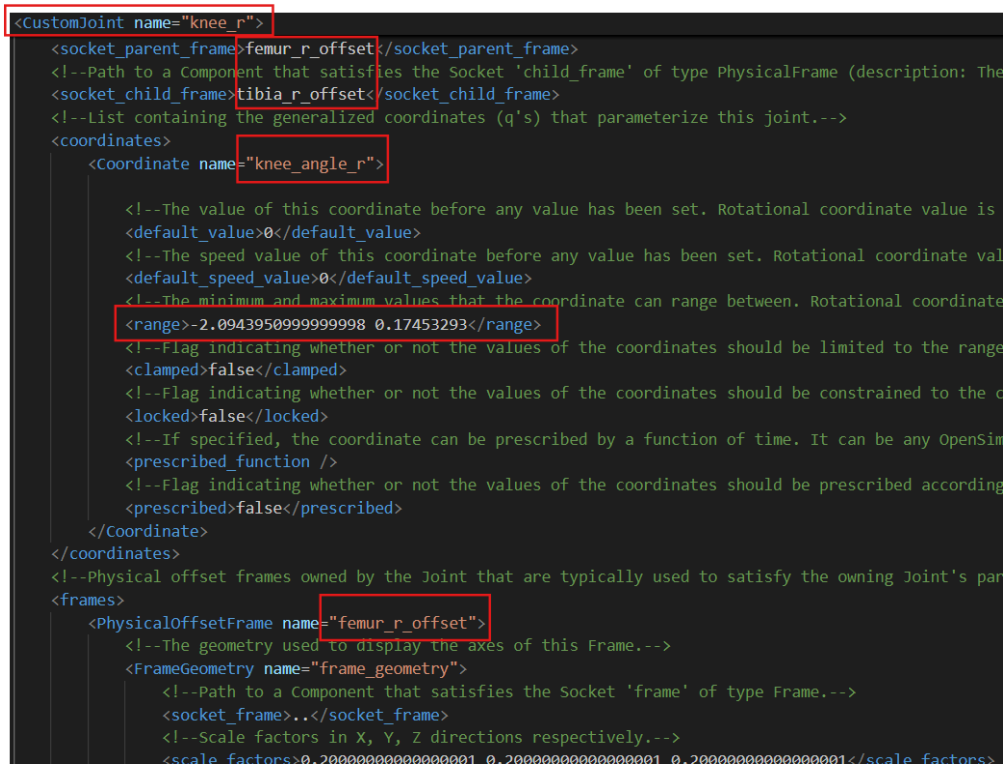
- **Custom Joint**

Las articulaciones de la rodilla y el tobillo en los modelos de OpenSim suelen ser de tipo *CustomJoint* dado que permiten configurar diferentes movimientos, según el número de grados de libertad que se deseen agregar a la articulación, por lo que para recrear el movimiento de la rodilla protésica y el tobillo se usarán este tipo de articulaciones.

Para crear la articulación de la rodilla se realizará una copia de la rodilla de la extremidad sana, en este caso se usará la rodilla derecha, dado que se realizó la simulación de amputación en el miembro inferior izquierdo, para encontrarla se debe desplegar la sección *JointSet,* copie y pegue la porción de código entre las etiquetas *CustomJoint* debajo de la articulación que se creó entre el fémur y el encaje protésico para mantener el órden del modelo.

Como se marca en la **Fig. 133**, es necesario hacer las siguientes modificaciones para que la articulación quede configurada adecuadamente:

- *CustomJoint name*: Escriba el nombre que se asignará a la articulación, en este caso, se sugiere cambiar *knee_r* por *knee_l*.
- *socket_parent_frame:* Escriba el nombre del cuerpo padre, en este caso fémur izquierdo, tal cual nombró el cuerpo en la sección *BodySet.*
- *socket_child_frame:* Escriba el nombre del cuerpo hijo, en este caso el vástago (*pylon_l*), tal cual nombró el cuerpo en la sección *BodySet.*
- *Coordinate name="knee_angle_r":* Reemplace el nombre *knee_angle_r* por el de su articulación y el lado de la misma, si lo desea puede mantener la palabra *angle* en el medio, esta será la coordenada mediante la cual se parametrice la articulación.
- *<range>-2.0943950999999998 0.17453293</range>:* En esta sección se describe en radianes el rango de rotación de la articulación, debe modificarlo según las especificaciones de su diseño prostético.
- *PhysicalOffsetFrame name="femur_l_offset:* Cada cuerpo de la articulación debe tener un Marco (*Frame*) para que se escale y ajuste adecuadamente en el modelo, el primer marco será el del cuerpo Padre y el segundo el del cuerpo hijo, sustituya el prefijo *femur_l* por el nombre del cuerpo padre de su modelo, pero mantenga el sufijo *_offset*.
- *<socket_parent>/bodyset/femur_r</socket_parent>*: En esta sección reemplace *femur_r* por el nombre del cuerpo padre de su modelo.
- *<PhysicalOffsetFrame name="tibia_r_offset"*: El segundo marco es el del cuerpo hijo, sustituya el prefijo *tibia_r* por el nombre del cuerpo hijo de su modelo, es decir, el vástago, pero mantenga el sufijo *_offset*.
- *<socket_parent>/bodyset/tibia_r</socket_parent>:* En esta sección reemplace *tibia_r* por el nombre del cuerpo hijo de su modelo, es decir, el vástago.



a

b

Fig. 133 Parámetros a editar dentro de una articulación tipo *CustomJoint*

En la **Fig. 133b** se han marcado en amarillo las etiquetas *translation* del frame hijo de la articulación, estas servirán para reposicionar la geometría en caso de ser

necesario una vez se pueda abrir el modelo en OpenSim, sin embargo, no las modifique aún, pues primero debe terminar de configurar algunos parámetros y no siempre se requiere realizar traslaciones de la misma.

Como se presenta en la **Fig. 134**, entre las etiquetas de *SpatialTransform* hay 6 secciones de *TransformAxis,* cada una de ellas hace referencia a un tipo de movimiento en un eje específico, en este caso 3 rotaciones y 3 traslaciones, dado que la rodilla cuenta con 6 grados de libertad, según su modelo protésico usted deberá considerar qué movimientos permite y cuáles debe eliminar.

```
<CustomJoint name="knee_r">
            <FrameGeometry name="frame_geometry">
                <!--Path to a Component that satisfies the Socket 'frame' of type Frame.-->
                <socket_frame>..</socket_frame>
                <!--Scale factors in X, Y, Z directions respectively.-->
                <scale_factors>0.20000000000000001 0.20000000000000001 0.20000000000000001</scale_factors>
            </FrameGeometry>
            <!--Path to a Component that satisfies the Socket 'parent' of type C (description: The parent f
            <socket_parent>/bodyset/tibia_r</socket_parent>
            <!--Translational offset (in meters) of this frame's origin from the parent frame's origin, exp
            <translation>0 0 0</translation>
            <!--Orientation offset (in radians) of this frame in its parent frame, expressed as a frame-fix
            <orientation>0 0 0</orientation>
        </PhysicalOffsetFrame>
    </frames>
    <!--Defines how the child body moves with respect to the parent as a function of the generalized coordi
    <SpatialTransform>
        <!--3 Axes for rotations are listed first.-->
        <TransformAxis name="rotation1">···
        </TransformAxis>
        <TransformAxis name="rotation2">···
        </TransformAxis>
        <TransformAxis name="rotation3">···
        </TransformAxis>
        <!--3 Axes for translations are listed next.-->
        <TransformAxis name="translation1">···
        </TransformAxis>
        <TransformAxis name="translation2">···
        </TransformAxis>
        <TransformAxis name="translation3">···
        </TransformAxis>
    </SpatialTransform>
```

Fig. 134 Secciones *TransformAxis* dentro de una articulación

En la **Fig. 135** se muestran las componentes de la rotación 1 y 2 de la rodilla, en la rotación 1 se tiene la coordenada de rotación, esta se debe de modificar por la misma que se renombró entre las etiquetas, *Coordinate name="knee_angle_r":,* en la parte superior del código que modifica la articulación, adicionalmente se define el eje de rotación Z, entre las etiquetas *axis*, los ejes se activan y desactivan con valores lógicos, y su orden es XYZ. Finalmente se define la función de rotación, en este caso es una función lineal, es decir que satisface la ecuación y=mx+b, donde, m=1 significa que la coordenada se escala directamente sin cambios y b=0 representa que no hay desplazamiento adicional. Por otro lado, en la rotación 2 el eje Y se encuentra activado, pero su función no es lineal sino una

constante cuyo valor es cero, lo que significa que no hay rotación efectiva en este eje, lo mismo sucede para la rotación 3, la cual se da en el eje X. Debe considerar los puntos previamente mencionados en caso de que las características de su prótesis sean diferentes.

```
<SpatialTransform>
    <!--3 Axes for rotations are listed first.-->
    <TransformAxis name="rotation1">
        <!--Names of the coordinates that serve as the independent variable
        <coordinates>knee_angle_r</coordinates>
        <!--Rotation or translation axis for the transform.-->
        <axis>0 0 1</axis>
        <!--Transform function of the generalized coordinates used to
        <LinearFunction name="function">
            <coefficients> 1 0</coefficients>
        </LinearFunction>
    </TransformAxis>
    <TransformAxis name="rotation2">
        <!--Names of the coordinates that serve as the independent variable
        <coordinates></coordinates>
        <!--Rotation or translation axis for the transform.-->
        <axis>0 1 0</axis>
        <!--Transform function of the generalized coordinates used to
        <Constant name="function">
            <value>0</value>
        </Constant>
    </TransformAxis>
```

Fig. 135 Parámetros dentro de *TransformAxis*

Finalmente, respecto a los movimientos de traslación, estos indican cómo se mueve la extremidad según el ángulo de la articulación, en la **Fig. 136** se presentan las traslaciones 1 y 3, en la traslación 1 se define la coordenada, que como se explicó previamente corresponde al mismo nombre que se usó en la definición inicial de la articulación, luego se selecciona el eje, en este caso el eje X, posteriormente se define una función lineal en la cual se describen valores para X y Y, en X se agregan los ángulos en radianes que van desde el valor mínimo hasta el valor máximo que se había definido en la parte inicial de la articulación, mientras que en Y se escriben las traslaciones en metros que se permiten durante cada ángulo de la articulación, es decir, en un ángulo de -2.094 radianes (máxima flexión) se permite un desplazamiento en el eje X de -0.0032 metros, estos valores se pueden mantener o modificar según las condiciones del dispositivo. En la sección de traslación 2 la dinámica es la misma, sin embargo, en eje de traslación ahora es en Y, por lo que los valores de desplazamiento cambian, mientras que en la traslación 3 se presenta de nuevo una constante de valor 0, lo que significa que no hay desplazamiento en este sentido.

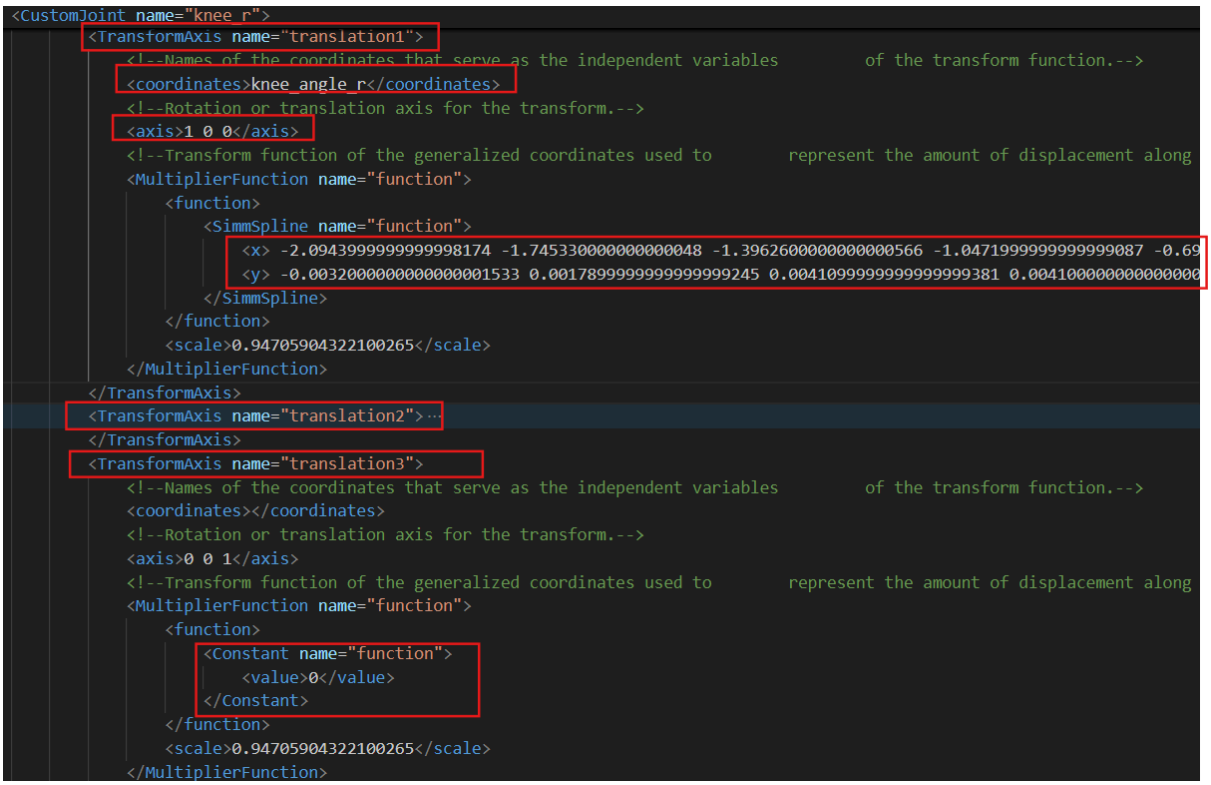


Fig. 136 Traslaciones dentro de una articulación tipo *CustomJoint*

Ahora bien, la articulación del tobillo puede seguir el mismo proceso anterior para implementarse, empleando una articulación de tipo *CustomJoint* en caso de que su dispositivo permita movimiento relativo entre el vástago y el pie, para ello puede partir de copiar y pegar la articulación del tobillo del miembro sano. Sin embargo, si su prótesis no tiene articulación en el tobillo entonces se recomienda usar una articulación de tipo *WeldJoint*, cuyas características se explicaron previamente.

Una vez realizadas todas las transformaciones anteriores, puede volver a abrir su modelo en OpenSim para revisar tanto el movimiento como la ubicación espacial de la geometría, **Fig. 137.**

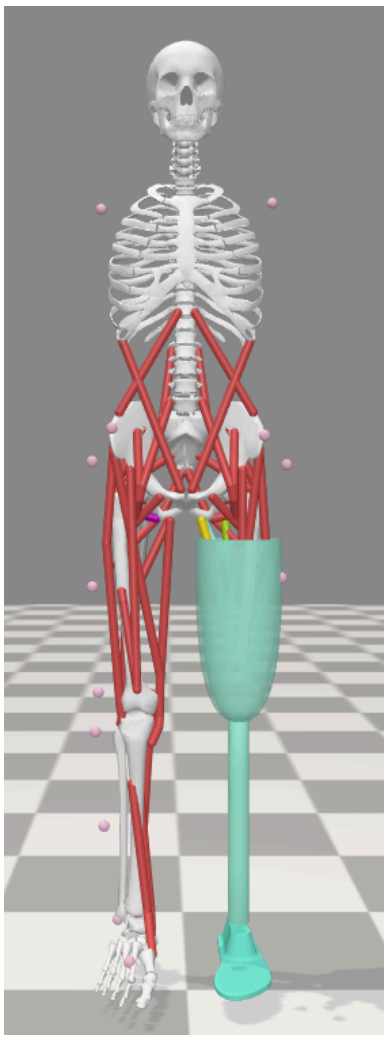


Fig. 137 Visualización de amputación transfemoral y geometría protésica

En las secciones anteriores se explica como realizar posicionamiento de marcadores, así como escalado, cinemática y dinámica inversa y RRA, todas estas operaciones se pueden aplicar sobre este este modelo, sin embargo, debe considerar los nuevos nombres asignados tanto a las articulaciones como a los cuerpos, para realizar las modificaciones pertinentes.

Referencias

[1] SimTK, “What is SimTK?” Disponible en: https://simtk.org/whatIsSimtk.php

[2] M. Haro., “Laboratorio de análisis de marcha y movimiento,” *Revista médica Clínica Las Condes*, vol. 25, no. 2, pp. 237-247, 2014, doi: 10.1016/S0716-8640(14)70034-3

[3] J. Hicks, “Collecting Experimental Data” 2018. Disponible en:https://shorturl.at/AWVnm

[4] Mokka, “Motion Kinematic and kinetic analyzer” Disponible en: https://biomechanical-toolkit.github.io/mokka/

[5] J. Hicks, “How Inverse Kinematics Works.” 2018. Disponible en: https://shorturl.at/EmmD6

[6] J. Hicks, “Inverse Kinematics Best Practices”, 2012. Disponible en: https://shorturl.at/GmFmA

[7] J. Hicks, “Getting Started with Inverse Dynamics.” 2018. Disponible en: https://shorturl.at/nu4Au

[8] J. Hicks, “How RRA Works.” 2018. Disponible en: https://shorturl.at/Eig4y

[9] J. Hicks, “Getting Started with Static Optimization.” 2018. Disponible en: https://acortar.link/h91zGn

[10] J. Hicks, “Getting Started with CMC” 2018. Disponible en: https://acortar.link/LkOGXJ

Programa de Bioingeniería

Desarrollo de una guía metodológica para el registro y análisis de marcha protésica en usuarios de prótesis de miembro inferior utilizando OpenSim

UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería

PRACTICANTE: María Paulina Tobón Palacio

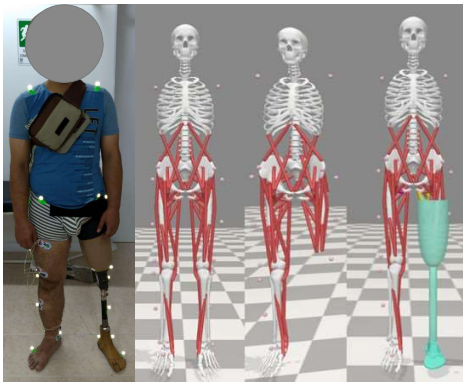
ASESORES: Juliana Uribe Pérez

PROGRAMA: Bionigeniería

Modalidad de la práctica: Trabajo de grado

Este trabajo tuvo como objetivo desarrollar una guía para el registro y análisis de la marcha protésica en usuarios de prótesis de miembro inferior utilizando OpenSim. A partir de una revisión bibliográfica, se abordaron aspectos clave del procesamiento de datos de marcha y el uso de herramientas del software como el escalado, la cinemática y dinámica inversa, la optimización estática y la simulación de amputación transfemoral.

Estos procesos se documentaron en una guía metodológica ilustrada y validada mediante una evaluación que arrojó una calificación promedio de 4.78 sobre 5. Además, se complementó el material con un video explicativo, concluyendo que fue posible establecer una metodología clara y detallada para este tipo de análisis biomecánico.



Introducción

La pérdida de una extremidad inferior afecta gravemente la movilidad e independencia de las personas. En Colombia entre el 4.3% y 6.3% de la población mayor de 5 años tiene alguna discapacidad, de las cuales el 31% presenta dificultades de movimiento. Esto resalta la necesidad de generar estrategias para estudiar la marcha en personas con discapacidad. Aunque herramientas como OpenSim permiten simular y analizar la marcha protésica, aún no existe una guía clara para su implementación, por lo que se propone desarrollar un protocolo que oriente a los investigadores en este proceso.

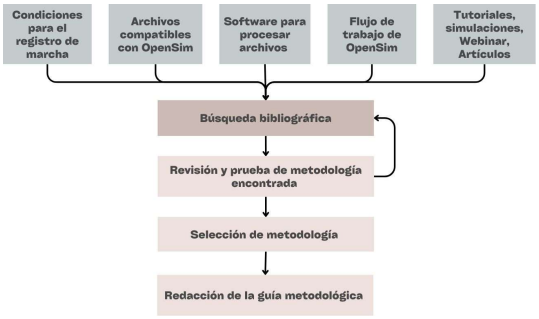
Objetivos

Desarrollar una metodología para el registro de la marcha en usuarios de prótesis de miembro inferior y el análisis de la marcha protésica utilizando OpenSim.

- Implementar un protocolo estandarizado del registro e importación de datos de marcha protésica en el software OpenSim.
- Desarrollar un flujo de trabajo para el análisis de marcha protésica en OpenSim.
- Evaluar la calidad de los protocolos propuestos a través de la retroalimentación de usuarios con conocimientos en el manejo de OpenSim.

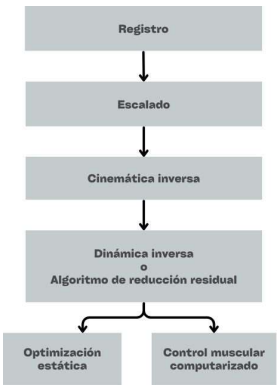
Metodología

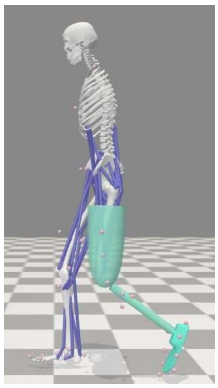
A partir de registros de marcha de usuarios de prótesis transfemoral y de revisiones bibliográficas se seleccionaron los métodos más claros para incluirse en una guía metodológica detallada que guíe al usuario a través del flujo de trabajo de OpenSim para realizar análisis de marcha y simulación de amputación.



Resultados

- Se desarrolló una guía metodológica que consta de nueve secciones, una para cada parte del flujo de trabajo de OpenSim, además del registro de marcha y la simulación de amputación.
- No se obtuvieron resultados adecuados para algunas simulaciones debido al origen de los datos.
- Fue posible realizar la simulación de amputación transfemoral y añadir el procedimiento a la guía.
- Se obtuvo una calificación de 4.78/5 en nivel de claridad.





Conclusiones

- Los modelos generados en OpenSim son sensibles a la calidad de los datos utilizados, la precisión de los resultados depende de la correcta captura y procesamiento de la información experimental.
- Se logró establecer una metodología detallada para el registro y el análisis de la marcha protésica, así como la simulación de amputación transfemoral, utilizando OpenSim.
- Aunque OpenSim es útil para el análisis biomecánico, presenta limitaciones como la necesidad de ajustes manuales en el escalado, la dependencia de datos experimentales de alta calidad y una curva de aprendizaje que puede dificultar su uso para principiantes.

Referencias

DATOS DE CONTACTO DEL AUTOR:

3052409172 mpaulinatp@gmail.com www.linkedin.com/in/maria-paulina-tobon-palacio