UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

Analisis de movimiento en ejercicios deportivos y de rehabilitación con el Sistema Optitrack

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

AUTOR/A: Linares Muñoz, David

Tutor/a: Mossi García, José Manuel

CURSO ACADÉMICO: 2021/2022

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

Agradecimientos

Han sido cinco años duros, con muchos altibajos y dudas, pero gracias a la gente que me rodea y que me ha apoyado he conseguido esas ganas y esa ambición que me faltaba para llegar donde estoy ahora mismo.

Gracias a mi tutor José Manuel García Mossi por esa propuesta de participar en un proyecto que generó un gran interés en mi persona; gracias por todo el tiempo invertido en hacer que disfrute elaborando un proyecto que sin ti ni se me hubiera pasado por la cabeza; gracias por apoyarme y ayudarme con todas las dudas y situaciones difíciles que han ocurrido durante el desarrollo.

Gracias a mis amigos por todo el apoyo que me han brindado a lo largo de los años y por confiar en mi y no bajarse de este barco, mi barco. Gracias Carlos e Iván por haber estado estos cinco años conmigo, por aguantarme y por animarme a seguir a pesar de todos los problemas que haya tenido, sois mi segunda familia. Gracias Joan, nos conocimos este último año, pero has sido mi compañero de aventuras durante este último curso y parte de este proyecto también se ha podido realizar gracias a ti. Gracias Bea por aparecer este año en mi vida, nunca hubiera pensado que mi compañera de piso, a la que he conocido este año, se convertiría en una persona tan importante para mí, que me ha apoyado y animado a darlo todo para terminar este largo camino.

Por último, agradecer a mi familia todo el apoyo y confianza depositada en mí. Gracias a mis abuelos y yayos por sentirse tan orgullosos de mí incluso sin tener ni idea de lo que conlleva haber estudiado esta carrera, ellos saben que su nieto va a ser ingeniero y con eso les es más que suficiente. Gracias a mi hermano que mi primer año de carrera se hiciera más ameno, ayudándome en todo lo posible para que estuviera cómodo; gracias por confiar en mí, aunque no me lo digas a menudo. Gracias a mis padres, sin vosotros no hubiera llegado tan lejos. Gracias por vuestros consejos, por apoyarme cuando más lo necesitaba, por proporcionarme todos los medios necesarios para que estuviera bien tanto física como mentalmente. Gracias mamá por tu apoyo incondicional, por tus esfuerzos en hacerme entender que no todo es blanco o negro y por aguantar mis cambios de humor por la frustración que he sentido muchas veces a lo largo del camino. Gracias papá por confiar en mí a pesar de mis resultados en muchas ocasiones, no es fácil levantarse cuando todo va mal, pero siempre has estado ahí para todo.

Resumen

Este proyecto parte de la idea de ayudar a los entrenadores deportivos y a los fisioterapeutas rehabilitadores con su trabajo mediante el análisis de movimientos y actividades deportivas. Dicho proyecto se compone de dos fases.

La primera fase consiste en grabar las actividades y ejercicios llevados a cabo por una persona con un sistema de captura de movimiento a través de unos marcadores especiales adheridos a su cuerpo o a las herramientas deportivas que se desean seguir con el programa. De esta manera se obtiene el fichero de datos que se utilizará en la siguiente parte.

La segunda fase consiste en el análisis de datos mediante Matlab (sistema de cómputo numérico) con el fin de obtener los parámetros de velocidad, aceleración, fuerza y potencia del ejercicio llevado a cabo. Asimismo, se obtendrá una gráfica 3D de los marcadores para visualizar el ejercicio sin necesidad del software.

Resum

Aquest projecte comença de la idea de ajudar al entrenadors esportius i al fisioterapeutes de rehabilitació amb el seu treball mitjançant l'anàlisi de moviments i activitats esportives. Aquest projecte es compon de dues parts.

La primera part consisteix en la gravació d'activitats y exercicis fets per una persona amb un sistema de moviment mitjançant uns marcadors especial adherits al seu cos o a les eines esportives que es volen seguir amb el programa. D'aquesta manera s'obté el fitxer de dades que s'utilitzarà en la següent part.

La segona part consisteix en l'anàlisi de dades mitjançant Matlab (sistema de còmput numèric) amb la finalitat d'obtindre els paràmetres de velocitat, acceleració, força i potencia del exercici realitzat. Així mateix, s'obtindrà una gràfica 3D dels marcadors per a visualitzar el exercici sense necessitat del software.

Abstract

This project is based on the idea of helping sport coaches and rehabilitation physiotherapists with their work through the analysis of movements and sport activities. It is divided into two phases.

The first phase consists of recording an individual's activities and exercises with a motion capture system through special markers attached to the body or to the sport tools to be followed. In this way the data file that will be used in the next part is obtained.

The second phase consists of data analysis using Matlab (numerical computing system) to obtain the parameters of speed, acceleration, strength and power of the exercise carried out. Likewise, a 3D graph of the markers will be obtained to visualize the exercise, so that the professional avoids the software.

Índice General

Contenido

Índice de ilustraciones

Índice de fragmentos de código

Capítulo 1. Introducción

La detección de movimiento mediante marcadores es una tecnología que se usa desde hace tiempo en la industria del cine y que, actualmente, se está extendiendo a otros campos tales como la realidad virtual o la rehabilitación.

Las salas que utilizan esta tecnología son conocidas como salas MoCap *"Motion Capture"*, que traducido al castellano significa captura de movimiento.

Estas salas ofrecen diversos beneficios si se comparan con otros métodos de captura de movimiento utilizados. Uno de ellos es la detección de personas u objetos que tengan los marcadores de seguimiento adheridos, de este modo se ahorran recursos tecnológicos, ya que no se debe procesar un vídeo entero, si no que se analiza específicamente la entidad marcada.

Esta tecnología aplicada a la práctica deportiva podría ayudar a entrenadores deportivos y fisioterapeutas rehabilitadores a mejorar sus técnicas. A su vez, contribuiría a que los usuarios aprendan a realizar los ejercicios de una manera correcta y óptima para evitar lesiones, ya que la realización de estas actividades de fuerza puede causar daño e inflamación en el tejido muscular debido a la contracción de estos.

En el ámbito deportivo existen dos tipos de contracciones musculares: la isométrica y la isotónica. La isométrica es aquella en la que no varía la distancia de los extremos del músculo ejercitado, mientras que la isotónica es en la que sí varía esta distancia a la hora de realizar el ejercicio. [1]

Dentro de este último tipo, se pueden encontrar dos maneras diferentes de contraer el músculo: la primera es conocida como contracción concéntrica, que consiste en que los extremos del músculo ejercitado se acercan venciendo la resistencia de carga; y la segunda, conocida como contracción excéntrica, es aquella donde los extremos de los músculos se alejan a la vez que la fuerza externa supera a la generada por el ejercicio. De estos dos tipos, la contracción excéntrica es la que produce una mayor inflamación y daño a nivel muscular, más aún cuando la velocidad de ejecución es elevada. Sin embargo, su correcta ejecución mediante el método *"Repeated Bout Effect"*, cuya traducción es *"Efecto de Intentos Repetidos"*, que consiste en la realización de una sesión de ejercicios únicamente excéntricos, protegería al músculo de futuras lesiones ocasionadas por el deporte. [2]

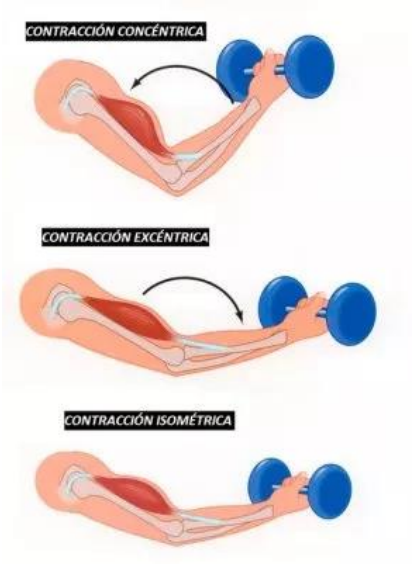


Ilustración 1. Diferentes tipos de contracciones musculares.

Este trabajo forma parte de un proyecto de mayor alcance en el que la Universitat Politècnica de València (UPV) colabora con la Universidad de Alicante (UA). Los departamentos que trabajan en dicho proyecto son el Departamento de Comunicaciones, cuyo responsable es el Dr. D. José Manuel García Mossi, por parte de la UPV y el Departamento de Educación Física, cuyo responsable es Basilio Pueo Ortega, por parte de la UA.

Capítulo 2. Objetivos

2.1 Objetivo principal

El objetivo que se persigue en este Trabajo Fin de Grado es la grabación de actividades y ejercicios deportivos mediante un conjunto de cámaras para obtener datos de coordenadas del sujeto u objeto a analizar y, posteriormente, procesar dichos datos y obtener los parámetros de velocidad, aceleración, fuerza y potencia.

2.2 Objetivos secundarios

- Adquirir conocimientos del software a utilizar.
- Realizar grabaciones de ejercicios deportivos y exportar los datos.
- Elaborar un programa que analice los ficheros de datos.
- Calcular los parámetros mediante el programa diseñado.
- Obtener gráficas en base a los parámetros obtenidos.
- Programar un script que represente en 3D los marcadores en movimiento de una persona.

Capítulo 3. Estado del arte

3.1 Antecedentes de la tecnología MoCap

Las tecnologías de captura de movimiento han evolucionado de manera considerable durante los últimos años, aumentando así el número de campos en los que se utiliza dicha técnica. Actualmente, su uso es muy frecuente en la industria del cine y de los videojuegos, donde se utiliza para generar entornos y personajes ficticios. A pesar de que su uso es cada vez más extendido, hay estudios cinematográficos, como Pixar, que rehúsan de dicha tecnología y prefieren seguir utilizando técnicas más tradicionales. Por contrapartida a Pixar, existen muchas otras empresas que sí han optado por utilizar esta técnica para crear personajes animados y entornos 3D, algunos ejemplos serían "El Señor de los Anillos" (2001), "Polar Express" (2004) o "Avatar" (2009). [3]

Ilustración 2. A la izquierda "Gollum" creado gracias a la técnica MoCap, a la derecha el actor que interpreta los movimientos y gestos con sensores en la cara y en el traje.

Del mismo modo que se aplica al cine, se hace uso también en los videojuegos haciendo que los personajes o elementos sean mucho más realistas tanto en apariencia como en comportamiento. Un ejemplo de ello sería "The Last of Us" (2013). [4]

Ilustración 3. A la izquierda "Ellie" creado con MoCap, a la derecha la actriz que la interpreta.

3.2 Actualidad de la tecnología MoCap

Los avances y el desarrollo tecnológico, que se han llevado a cabo durante estos últimos años, han provocado que las técnicas MoCap se utilicen en nuevos campos como las aplicaciones móviles o la rehabilitación.

Dentro de las aplicaciones móviles podemos encontrar infinidad de ellas dedicadas a monitorizar pasos, calorías e incluso haciendo uso de dispositivos externos se analiza el sueño, las pulsaciones, estrés, etc.

Así como estas aplicaciones son cada vez más frecuentes en los móviles de los usuarios, también ha crecido el uso de aplicaciones dedicadas al seguimiento y análisis de ejercicios llevados a cabo con pesas. Dichas aplicaciones permiten analizar el ejercicio y obtener diferentes parámetros como la velocidad a la que se realiza, el espacio recorrido o la fuerza ejercida.

En este proyecto se han tomado como ejemplo de funcionalidad las aplicaciones WL Analysis [5] y Metric VBT (Velocity Based Training) [6] que realizan el análisis del ejercicio de pesas, pero mediante una única cámara por lo que los datos podrían ser imprecisos en algunos casos debido a que no ofrecen profundidad.

Ilustración 4. Aplicación WL Analysis para iPhone.

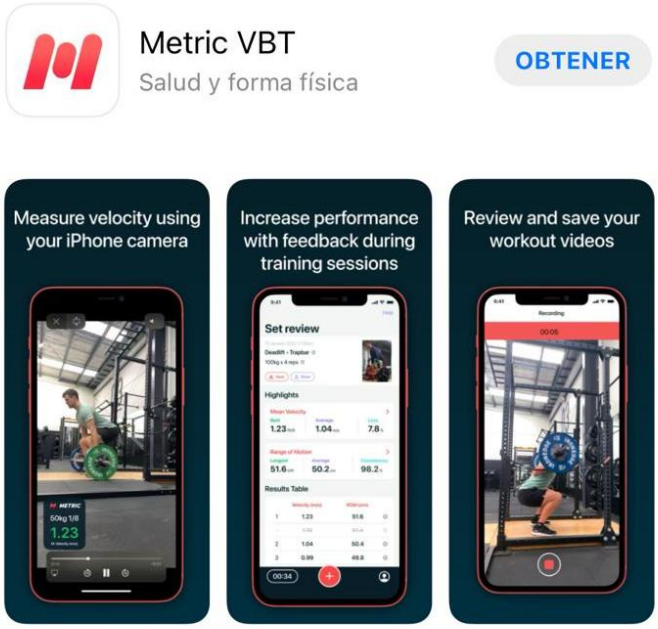


Ilustración 5. Aplicación Metric VBT para iPhone.

3.3 Captura óptica de movimiento

Estos sistemas utilizan los datos de sensores, que son captados por un conjunto de cámaras, para localizar la posición o movimiento de un objeto o persona. Para la utilización de esta técnica, se suele hacer uso de indicadores adheridos al elemento o individuo a localizar o seguir. Los datos obtenidos del seguimiento de cada indicador se componen de las coordenadas x, y, z de su posición en el espacio. Asimismo, si se define un cuerpo rígido a partir de 3 o más marcadores se pueden obtener otros tres valores correspondientes a los ángulos de la orientación de dicho cuerpo rígido.

Estos métodos de captura de movimiento son más baratos que los no ópticos y más cómodos en caso de que los tenga que llevar una persona. Además, son sistemas muy fiables para capturar ciertos movimientos y permiten la grabación en tiempo real. Esto último es muy favorable en algunos casos, pero también tiene limitaciones como podrían ser la cantidad de cámaras a utilizar, el número de marcadores o de individuos a seguir. [7]

Ilustración 6. A la izquierda un sistema de captura óptico, a la derecha uno mecánico (no óptico).

Los sistemas ópticos más habituales se componen de un conjunto de cámaras CCD (*"Charged-Coupled Device"*, cuyo significado en castellano es *"Dispositivo de Carga Acoplada"*) que mandan la señal digital a un único ordenador. Estos dispositivos crean representaciones digitales de la imagen con resoluciones que pueden variar. Cuanto mayor sea la resolución, mejor será la calidad de la imagen, pero también apareceran otros factores limitantes como el número de FPS (Fotogramas Por Segundo) que puede captar el dispositivo. Las cámaras suelen tener unas velocidades de captura de FPS entre 30 y 1000.

El número de cámaras del que se suele hacer uso en este tipo de técnicas está comprendido entre 4 y 32. Cabe destacar que con dos cámaras sería suficiente para obtener la localización de cualquier indicador, aunque es conveniente hacer uso de un mayor número para mantener siempre contacto visual con el marcador con al menos dos de ellas. Esto es debido a que si solo una cámara detectara el marcador se perdería la profundidad y solamente se obtendrían datos en 2D. A pesar de ello, hacer uso de un elevado número de cámaras complicaría el procesamiento de datos.

Ilustración 7. Cámara PrimeX 41 de la marca Optitrack.

En este tipo de sistemas se debe llevar a cabo una calibración de los dispositivos para minimizar el error de localización de marcadores y obtener el origen de coordenadas del sistema. Esto se realiza mediante dos objetos dotados de marcadores y un software que reconozca la forma y distancia predefinida entre ellos. De esta manera se consigue una identificación del espacio precisa y se consigue que las cámaras reduzcan al mínimo posible la probabilidad y la distancia de error a la hora de detectar los indicadores así como se establece el nivel del suelo y el origen del sistema de coordenadas. [8]

Ilustración 8. Vara de calibración espacial de la marca Optitrack.

Ilustración 9. Calibrador de pie CS-200 de la marca Optitrack.

3.3.1 Triangulación

La triangulación es un proceso mediante el cual dos o más cámaras determinan un punto en un espacio 3D dada su proyección en las imágenes de dichos dispositivos. Para poderlo llevar a cabo, se necesitan conocer los parámetros de proyección 3D a 2D de las cámaras.

Un punto de una imagen pertenece a una línea en un espacio 3D, es decir, desde el objetivo de la cámara hasta el punto de la imagen se genera una recta denominada rayo de proyección. En ésta, a parte del punto que se quiere localizar, también se encuentran otros puntos que, cuando se convierte dicha recta a 2D, se proyectarían en el mismo lugar que el buscado, esto es conocido como puntos correspondientes. Con una sola imagen, no se puede determinar la localización de ese punto en el espacio 3D, pero cuando se utilizan más cámaras, los puntos correspondientes no son los mismos en el resto de los dispositivos y, el punto de interés se corresponde con la intersección de las rectas de las distintas cámaras en las que aparece.

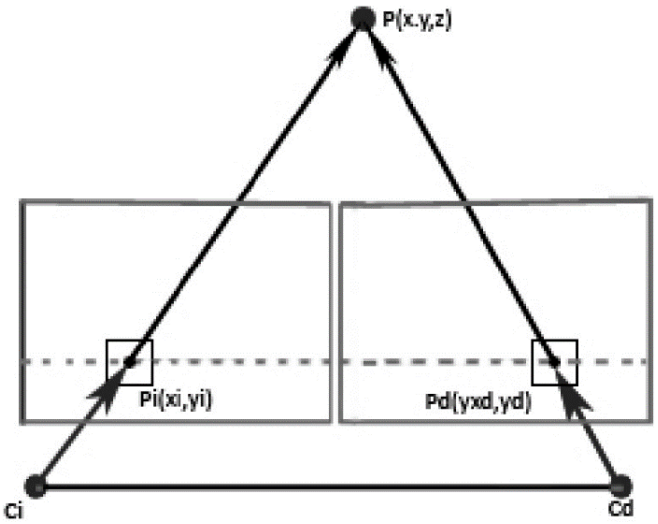


Ilustración 10. Triangulación del punto P de dos cámaras paralelas.

La resolución de este problema a nivel teórico resulta sencilla, pero a nivel práctico no es tan fácil como se plantea. Esto se debe a que la precisión de medida no es arbitraria debido a factores tales como el error geométrico por distorsión de la lente. Por consiguiente, los rayos proyectados que se generan a partir de las imágenes de las cámaras podrían no intersecar en el espacio. [9][10]

3.3.2 Geometría Epipolar

La geometría básica de un sistema de imagen estéreo es conocido como geometría epipolar. Esta geometría combina dos cámaras que ven un espacio 3D desde posiciones diferentes y, por tanto, existe una relación entre los puntos 3D y su proyección 2D en las imágenes que genera restricciones entre los puntos correspondientes.

Cada cámara tiene un centro de proyección diferente y su correspondiente plano de proyección. Conociendo esto, se deduce que un punto situado en un espacio físico tiene una proyección en cada uno de los planos. Asimismo, uniendo los centros de proyección de las cámaras, se crea en cada plano un punto conocido como epipolar, que es la imagen del centro de proyección de una cámara en el plano de la otra.

Al conjunto de la proyección del punto a localizar y el punto epipolar se le conoce como plano epipolar. A la recta resultante de la intersección del plano epipolar y el plano de la imagen derecha se le conoce como recta epipolar (indicada en rojo en la ilustración 11).

Dado un punto de una imagen, su visión correspondiente en la otra imagen debe encontrarse a lo largo de la línea epipolar correspondiente, esto se conoce como restricción epipolar. Esta restricción genera que la búsqueda bidimensional que se debería de realizar para encontrar el punto coincidente se convierte en una búsqueda unidimensional a lo largo de las líneas epipolares. Esto nos genera un ahorro computacional muy importante a la hora de procesarlo y también permite reducir en gran medida la cantidad de puntos que podrían ser posibles correspondencias.

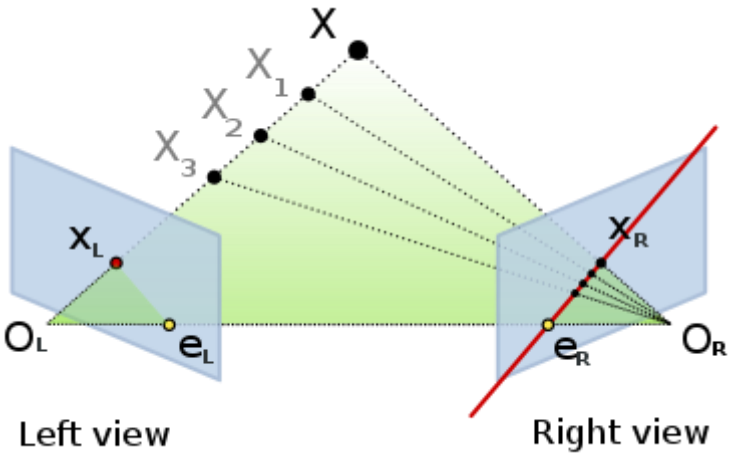


Ilustración 11. Geometría epipolar de dos cámaras.

Como se aprecia en la ilustración 11, el problema mencionado anteriormente de que varios puntos de una imagen 3D se proyectan como el mismo punto en el plano 2D, se soluciona aplicando la geometría epipolar. Los puntos que se proyectan sobre X_L y que el centro de proyección O_L podría identificar como idénticos, se proyectan en posiciones distintas en el plano de la derecha, de esta manera se identifica la localización exacta del punto X. [10]

3.4 Tipos de marcadores

3.4.1 Marcadores pasivos

Los marcadores pasivos son bolas de goma recubiertas de un material reflectante y suelen ir adheridos a la persona u objetivo que se quiere seguir, estos indicadores se colocan en lugares determinados (algunos predefinidos dependiendo del software y la entidad a seguir) para facilitar su detección. Están recubiertos de ese material para que la luz se refleje en ellos y las cámaras sean capaces de detectarlos. Además, las cámaras se pueden configurar para detectar un umbral de luz de tal manera que solo detecten la reflejada por estos marcadores.

Los marcadores deben ser detectados por al menos dos cámaras, de esta manera se puede determinar su ubicación en el espacio. Cuando se hace uso de estos marcadores, el usuario tiende a pensar que el software que los detecta podría llegar a confundir un marcador con otro, pero ese problema lo soluciona automáticamente el software mediante la triangulación y la geometría epipolar. [8]

Ilustración 12. Marcador pasivo de 19mm modelo M4 de la marca Optitrack.

3.4.2 Marcadores activos

Los marcadores activos emiten luz propia mediante leds, de esta manera se aumenta la distancia de detección de estos y, por tanto, se amplía el rango de actuación de las personas o de localización de objetos.

Los sistemas ópticos que hacen uso de este tipo de marcadores triangulan su posición iluminando un led de manera muy rápida, y es el software el encargado de identificarlos a través de sus posiciones relativas. Los indicadores se iluminan mediante baterías, esto provoca que usarlas en personas sea más incómodo debido a que tendrá que cargar con el traje y con las baterías. Otro factor limitante es la frecuencia de muestreo, ya que, al solo poder iluminarse un led a la vez, esta queda reducida al número de leds que puedan encenderse y apagarse en un fotograma.

En otros sistemas se han realizado pruebas con leds de diferentes colores. Estos se asignan a puntos específicos del cuerpo y de esa manera el software debe realizar menos cálculos para diferenciarlos. [8]

Ilustración 13. Traje de marcadores activos.

3.4.3 Marcadores activos modulados en el tiempo

Estos marcadores son una mejora respecto a los marcadores activos. Estos pueden iluminarse de manera simultánea mediante luz estroboscópica y la manera de que el software los diferencie y los identifique se realiza mediante la frecuencia de iluminación de estos. Esto mejora considerablemente la frecuencia de muestreo con respecto a los marcadores anteriores, pero a cambio, se incrementa la carga computacional.

Con este tipo de marcadores resulta más sencillo grabar al aire libre bajo la luz del sol porque, como se ha dicho antes, la detección de los marcadores se realiza mediante el parpadeo de las luces. Asimismo, se puede realizar el seguimiento de los movimientos del actor y ver los resultados sobre el personaje animado en tiempo real.

Los marcadores modulados se procesan en tiempo real mediante las cámaras, esto permite que se alcancen altas frecuencias de muestreo y mayor precisión en la detección de la posición y rotación del marcador. [8]

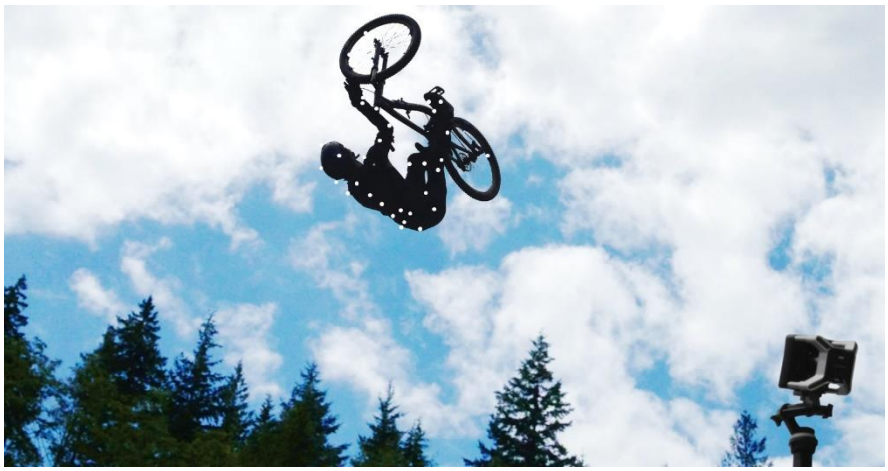


Ilustración 14. Traje de marcadores activos modulados en el tiempo.

3.5 Contexto tecnológico del software utilizado

3.5.1 Optitrack Motive

En este proyecto se ha utilizado el sistema MoCap de la marca Optitrack disponible en la ETSIT de la UPV.

Motive es el entorno de detección, captura y exportación de vídeos de la marca Optitrack. La detección se realiza a través de las cámaras que detectan los marcadores pasivos de diferentes tipos y tamaños. [11]

Este entorno contiene diferentes funciones que permiten al usuario realizar diversas modificaciones y llevar a cabo procesos tales como:

- Calibración de cámaras.
- Creación de objetos sólidos y esqueletos mediante marcadores.
- Configuración de los parámetros de las cámaras (Exposición, Activa/Inactiva, número de FPS a capturar, diferentes vistas…).

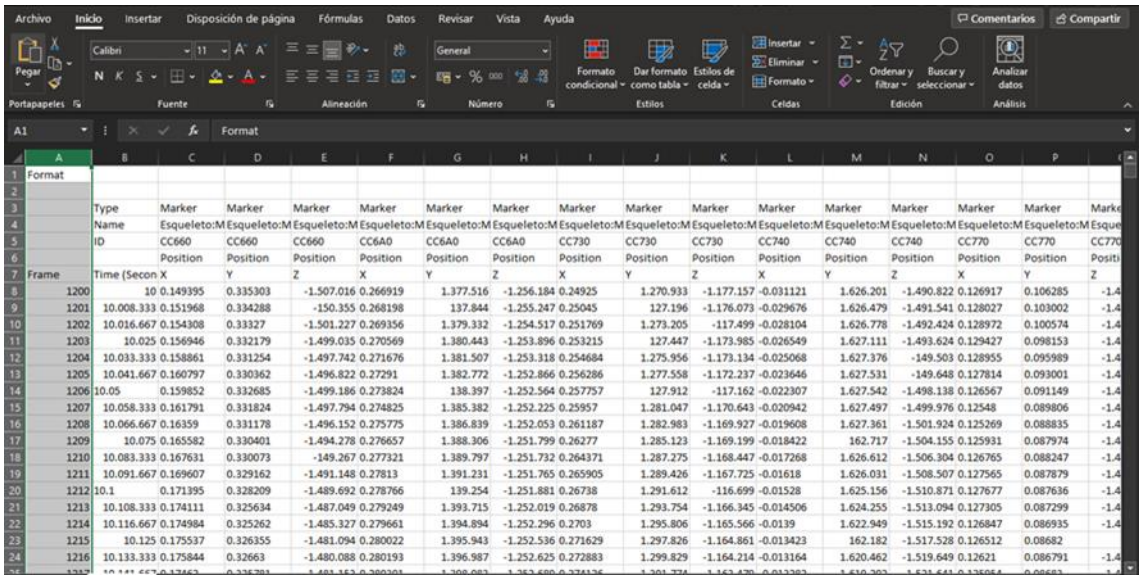
- Grabación de vídeos.
- Visualización y edición de los vídeos grabados.
- Exportación de ficheros de datos tanto de vídeos como de los objetos y esqueletos creados (ficheros .csv, .3d, .cal, etc.).

Ilustración 15. Software Motive.

3.5.2 Excel

Excel es un programa de Microsoft basado en hojas de cálculo que permite realizar diferentes operaciones matemáticas. En este proyecto se ha utilizado para la observación de datos mediante la carga de un fichero .csv (*"Coma Separated Values"*, en castellano *"Valores Separados por Coma"*) ya que tiene una función que divide dichos valores en columnas a través del análisis de un carácter. Esto facilita la comprobación de datos porque resulta más sencillo analizar datos por columnas que analizarlos en una misma línea, con diferentes longitudes y separados por una coma. [12]

Ilustración 16. Software Excel.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Format															
2																
3		Type	Marker	Marker	Marker	Marker	Marker	Marker	Marker	Marker	Marker	Marker	Marker	Marker	Marker	Marker	Marke
4		Name	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esqueleto:M	Esque
5		ID	CC660	CC660	CC660	CC6A0	CC6A0	CC6A0	CC730	CC730	CC730	CC740	CC740	CC740	CC770	CC770	CC770
6			Position	Position	Position	Position	Position	Position	Position	Position	Position	Position	Position	Position	Position	Position	Positi
7	Frame	Time (Secon	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
8	1200	10	0.149395	0.335303	-1.507.016	0.266919	1.377.516	-1.256.184	0.24925	1.270.933	-1.177.157	-0.031121	1.626.201	-1.490.822	0.126917	0.106285	-1.4
9	1201	10.008.333	0.151968	0.334288	-150.355	0.268198	137.844	-1.255.247	0.25045	127.196	-1.176.073	-0.029676	1.626.479	-1.491.541	0.128027	0.103002	-1.4
10	1202	10.016.667	0.154308	0.33327	-1.501.227	0.269356	1.379.332	-1.254.517	0.251769	1.273.205	-117.499	-0.028104	1.626.778	-1.492.424	0.128972	0.100574	-1.4
11	1203	10.025	0.156946	0.332179	-1.499.035	0.270569	1.380.443	-1.253.896	0.253215	127.447	-1.173.985	-0.026549	1.627.111	-1.493.624	0.129427	0.098153	-1.4
12	1204	10.033.333	0.158861	0.331254	-1.497.742	0.271676	1.381.507	-1.253.318	0.254684	1.275.956	-1.173.134	-0.025068	1.627.376	-149.503	0.128955	0.095989	-1.4
13	1205	10.041.667	0.160797	0.330362	-1.496.822	0.27291	1.382.772	-1.252.866	0.256286	1.277.558	-1.172.237	-0.023646	1.627.531	-149.648	0.127814	0.093001	-1.4
14	1206	10.05	0.159852	0.332685	-1.499.186	0.273824	138.397	-1.252.564	0.257757	127.912	-117.162	-0.022307	1.627.542	-1.498.138	0.126567	0.091149	-1.4
15	1207	10.058.333	0.161791	0.331824	-1.497.794	0.274825	1.385.382	-1.252.225	0.25957	1.281.047	-1.170.643	-0.020942	1.627.497	-1.499.976	0.12548	0.089806	-1.4
16	1208	10.066.667	0.16359	0.331178	-1.496.152	0.275775	1.386.839	-1.252.053	0.261187	1.282.983	-1.169.927	-0.019608	1.627.361	-1.501.924	0.125269	0.088835	-1.4
17	1209	10.075	0.165582	0.330401	-1.494.278	0.276657	1.388.306	-1.251.799	0.26277	1.285.123	-1.169.199	-0.018422	162.717	-1.504.155	0.125931	0.087974	-1.4
18	1210	10.083.333	0.167631	0.330073	-149.267	0.277321	1.389.797	-1.251.732	0.264371	1.287.275	-1.168.447	-0.017268	1.626.612	-1.506.304	0.126765	0.088247	-1.4
19	1211	10.091.667	0.169607	0.329162	-1.491.148	0.27813	1.391.231	-1.251.765	0.265905	1.289.426	-1.167.725	-0.01618	1.626.031	-1.508.507	0.127565	0.087879	-1.4
20	1212	10.1	0.171395	0.328209	-1.489.692	0.278766	139.254	-1.251.881	0.26738	1.291.612	-116.699	-0.01528	1.625.156	-1.510.871	0.127677	0.087636	-1.4
21	1213	10.108.333	0.174111	0.325634	-1.487.049	0.279249	1.393.715	-1.252.019	0.26878	1.293.754	-1.166.345	-0.014506	1.624.255	-1.513.094	0.127305	0.087299	-1.4
22	1214	10.116.667	0.174984	0.325262	-1.485.327	0.279661	1.394.894	-1.252.296	0.2703	1.295.806	-1.165.566	-0.0139	1.622.949	-1.515.192	0.126847	0.086935	-1.4
23	1215	10.125	0.175537	0.326355	-1.481.094	0.280022	1.395.943	-1.252.536	0.271629	1.297.826	-1.164.861	-0.013423	162.182	-1.517.528	0.126512	0.08682	
24	1216	10.133.333	0.175844	0.32663	-1.480.088	0.280193	1.396.987	-1.252.625	0.272883	1.299.829	-1.164.214	-0.013164	1.620.462	-1.519.649	0.12621	0.086791	-1.4

Ilustración 17. Fichero .csv separado en columnas con la herramienta Excel.

3.5.3 Matlab

Matlab es una plataforma de computación numérica y de programación que ofrece un entorno de desarrollo integrado (IDE), cuyo lenguaje de programación es propio (lenguaje M). [13]

Ilustración 18. Software Matlab.

Matlab posee una gran cantidad de opciones para llevar a cabo el análisis y el procesamiento de datos. Entre esas opciones se encuentran operaciones con vectores y matrices, programación orientada a objetos, desarrollo de algoritmos, etc. La aplicación se compone de una ventana de programación instantánea, en la que el código que se escribe se ejecuta cuando se presiona la tecla E*nter*, y de un espacio de trabajo que será donde se muestren las variables almacenadas. También se pueden crear *Scripts* donde se puede programar como se haría en un entorno de programación habitual y ejecutarlo todo mediante la opción *Ejecutar*.

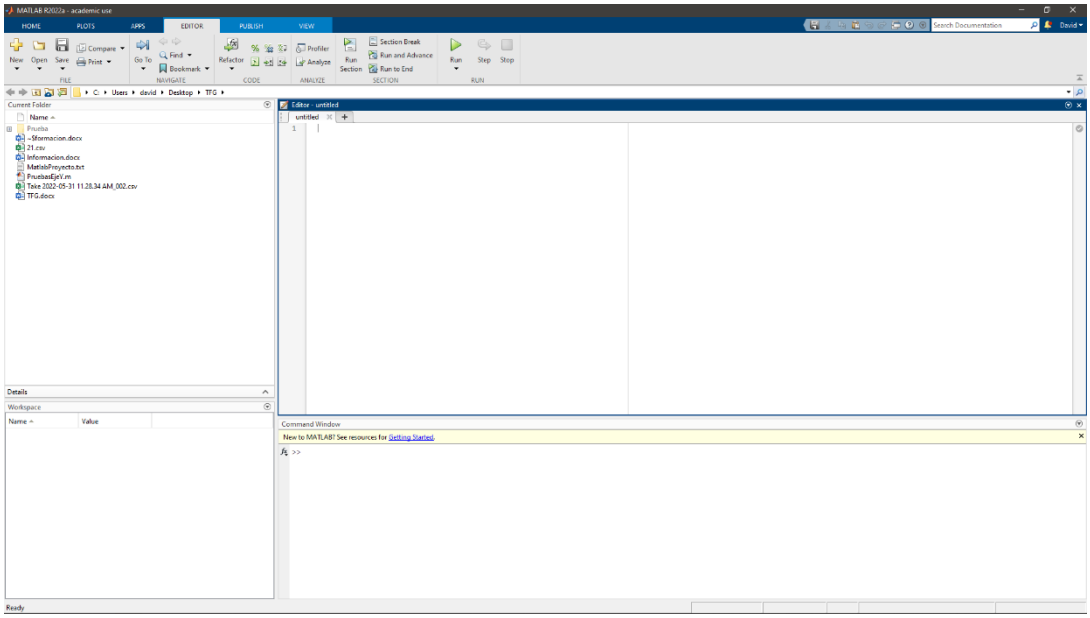


Ilustración 19. Matlab con el espacio de trabajo a la izquierda, el script en la parte superior y la ventana de comandos en la parte inferior.

Asimismo, existe una opción llamada LiveScript, donde se puede programar, ejecutar y visualizar en tiempo real el código completo o por secciones. Este método resulta cómodo para ir analizando poco a poco el código y los resultados obtenidos.

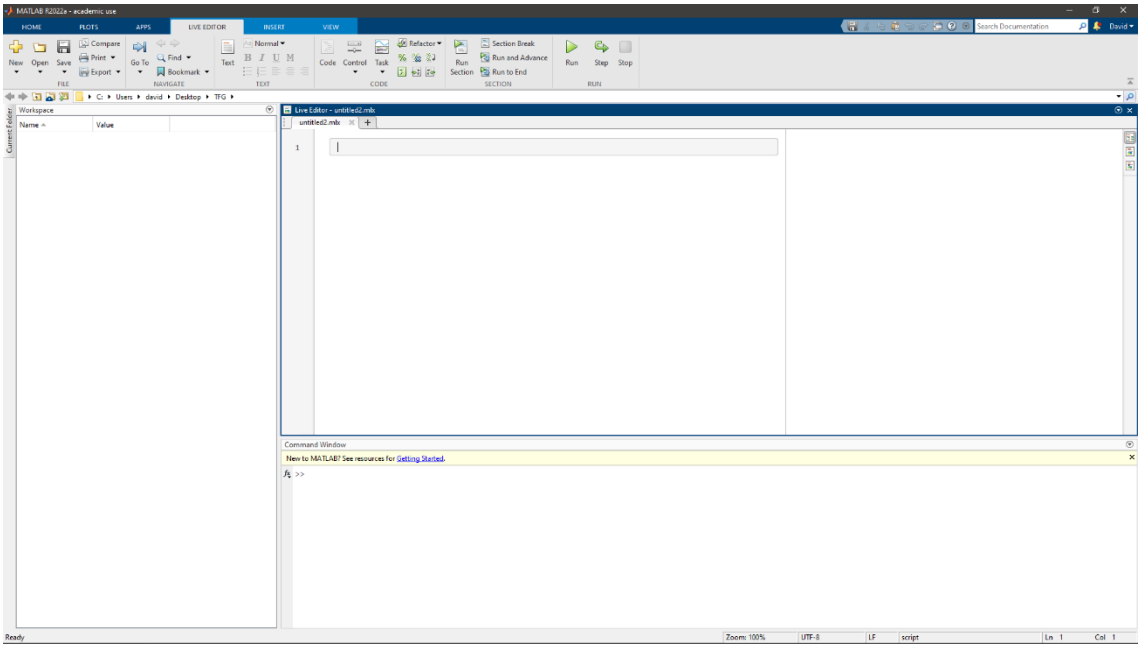


Ilustración 20. Matlab con el espacio de trabajo a la izquierda, el LiveScript en la parte superior donde la parte izquierda es el código y la parte derecha son los resultados y la ventana de comandos en la parte inferior.

3.6 Contexto tecnológico de la sala MoCap utilizada

Para la realización del proyecto se ha tenido acceso a la Sala MoCap situada en el edificio 4D de la Escuela Técnica Superior de Ingeniería de Telecomunicación de la UPV. La sala está dotada de un array de cámaras, accesorios de calibración y seguimiento y un equipo informático con el software instalado.

3.6.1 Posicionamiento de cámaras

La sala contiene un array de cámaras compuesto por un total de 12 dispositivos colocados en la parte alta de las paredes o colgando del techo, conformando así un rectángulo cuyo interior será el espacio de grabación.

Ilustración 21. Posicionamiento de cámaras en la sala.

Las cámaras utilizadas son las Flex 13 de la marca Optitrack. Su resolución es de 1280x1024 px, son capaces de grabar con una velocidad de 120 FPS y tienen un ángulo de visión de 56º. Además, en aplicaciones de seguimiento tienen un fallo máximo de 0.10 milímetros, siendo este muy bajo y proporcionando posiciones altamente precisas.

Algunas de las características de estos dispositivos se pueden modificar desde la aplicación UI (*"User Interface"*, cuya traducción al castellano es *"Interfaz de Usuario"*) como por ejemplo el control de exposición, la tasa de captura de fotogramas, la resolución y el modo de procesamiento.

Ilustración 22. Cámara Optitrack modelo Flex 13 utilizada en la sala

3.6.2 Material de seguimiento

Para realizar el seguimiento mediante las cámaras, se ha hecho uso de marcadores de tipo pasivo. Se han utilizado los marcadores M3 de 9.5mm para el seguimiento de objetos y marcadores X-base de 14mm con velcro para la identificación del esqueleto. Los marcadores X-base se colocaron sobre un traje hecho de material adherente de velcro, el modelo utilizado fue el Motion Capture Suit.

Ilustración 23. Materiales Utilizados para el seguimiento.

Capítulo 4. Uso de sala MoCap y software Motive

4.1 Puesta en marcha del equipo

Con el ordenador en funcionamiento, el primer paso es abrir el software Motive. Una vez el programa se encuentra en ejecución, las cámaras se encienden y comienzan a transmitir la señal al equipo y se visualiza a través del programa.

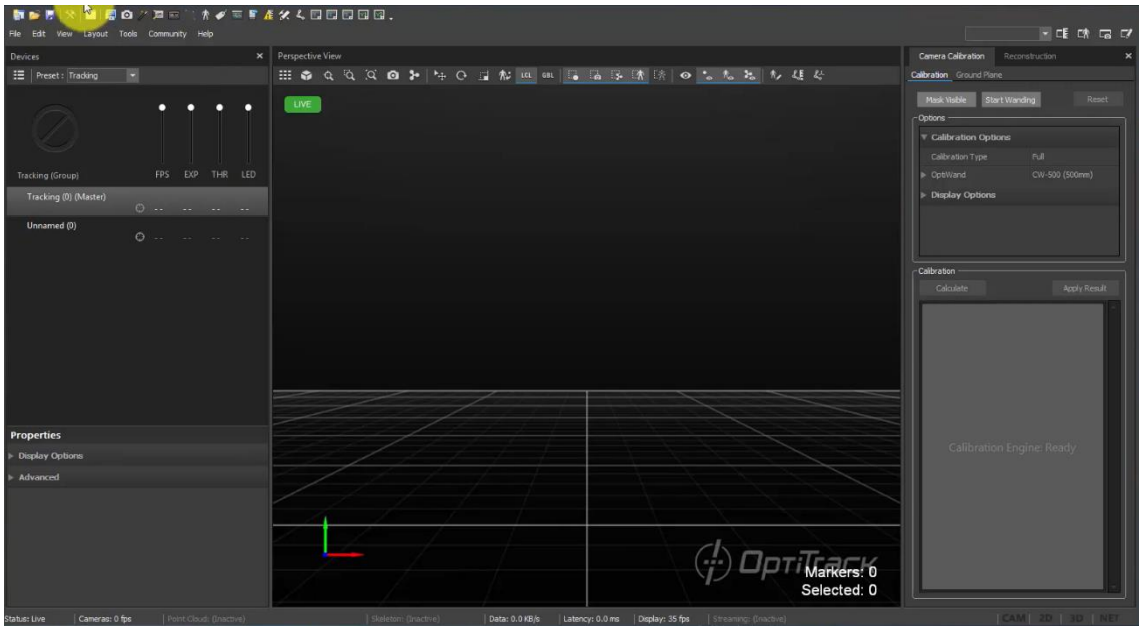


Ilustración 24. Pantalla inicial de Optitrack.

4.2 Calibración

Tras iniciar el software y encender las cámaras, se debe realizar una correcta calibración de estas. En la sala MoCap, como se aprecia en la ilustración 24, se encuentran varias mesas y sillas con superficies mínimamente reflectantes que pueden causar confusión en las cámaras.

Para reducir los reflejos lo máximo posible las persianas se deben cerrar, para minimizar la entrada de luz natural, y se han de encender todas las luces de la sala para crear una iluminación lo más homogénea posible. Hecho esto, las cámaras pueden detectar reflejos y lo comunican a través del software marcando la zona en blanco.

Para marcar esos reflejos y que no sean identificados como marcadores, hay que abrir el desplegable de *"Camera Calibration"* y seleccionar la opción *"Mask Visible"*. De esta manera, se aplica una máscara, marcando los reflejos en rojo, indicando a las cámaras que son zonas de no interés y no se toman en cuenta a la hora de calibrar.

Ilustración 25. Reflejo en la parte inferior izquierda generado por el flash del móvil.

Ilustración 26. Máscara aplicada al reflejo.

Una vez las cámaras ya detectan los reflejos marcados, se procede a calibrar las mismas para minimizar la distancia de error. Para ello, en las opciones de calibración se debe seleccionar el modelo de la vara a utilizar, en este caso la CW-500. Es importante seleccionar de manera adecuada la vara de calibración ya que los marcadores tienen unas distancias predefinidas que el software conoce y permite que

la calibración y la identificación del espacio sea la correcta. Hecho esto, es momento de empezar a calibrar, para ello hay que pulsar en el botón *"Start Wanding"*.

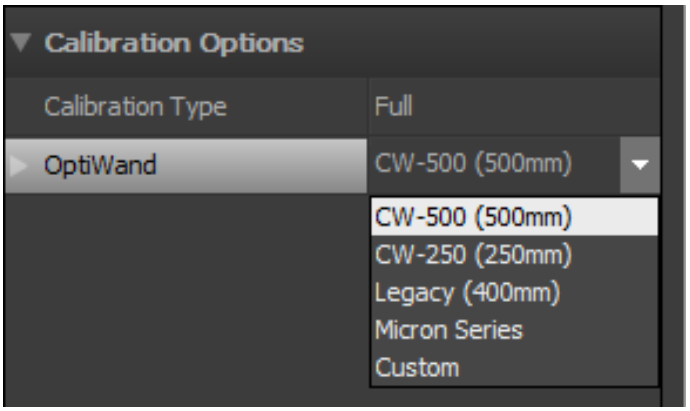


Ilustración 27. Selección de modelo de la vara de calibración.

Ilustración 28. Vara de calibración modelo CW-500

Para una correcta calibración, se recomienda que cada cámara recopile aproximadamente entre 2500 y 6000 muestras, por tanto, hay que mover la vara alrededor del espacio para que las cámaras detecten el movimiento y la posición en cada instante de tiempo de los marcadores. Una vez se ha alcanzado el número de muestras deseado se ejecuta la opción *"Calculate"* y se mostrarán los resultados de la calibración.

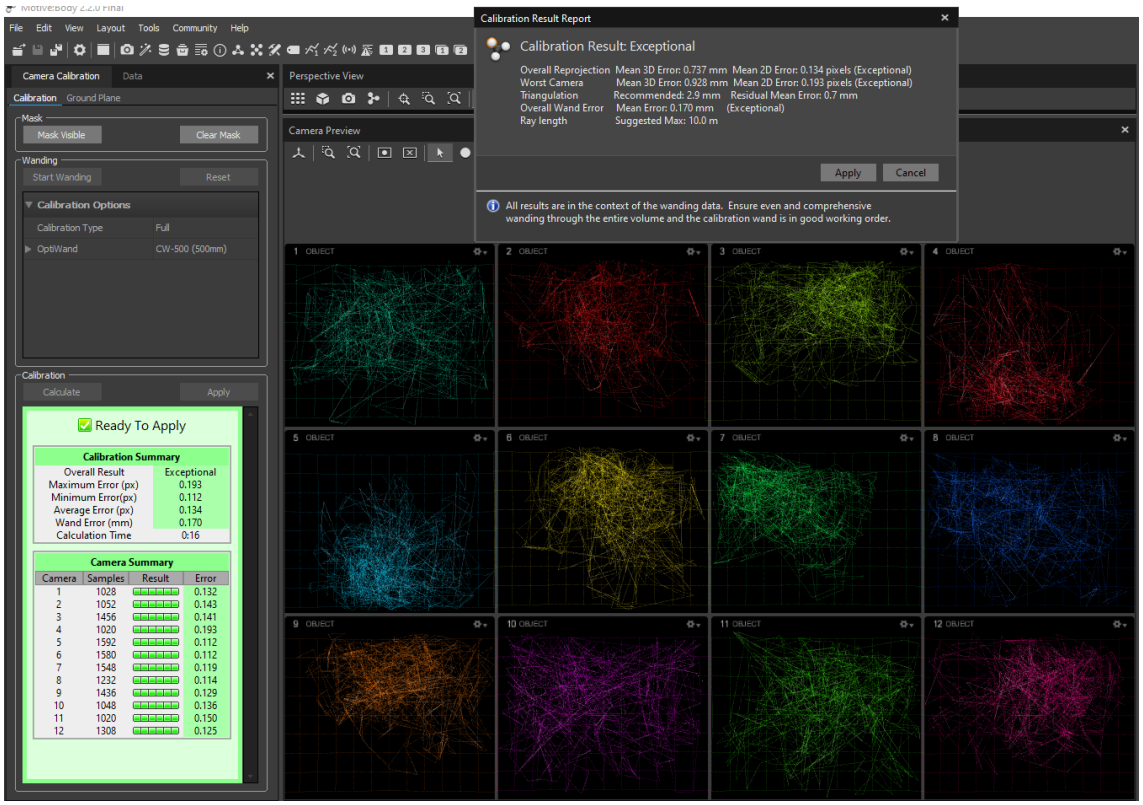


Ilustración 29. Resultados de calibración.

En los resultados de dicha calibración, se indica el error máximo (en píxeles), el error mínimo (en píxeles), el error medio (en píxeles) y el error de la vara (en milímetros). Además, debajo de estos datos, aparece de manera detallada el error medio de cada cámara, así como la cantidad de muestras que han detectado durante la calibración. Una vez hecho esto se le da a *"Apply"* y se procede con la calibración del suelo y del origen de coordenadas.

Esta calibración se realiza para que las cámaras sean capaces de detectar donde está el nivel del suelo y para establecer un origen de coordenadas. La parte larga del pie de calibración indicará el eje Z positivo y la parte corta indicará el eje X positivo, por lo que el eje Y positivo irá hacia arriba. Tras comprobar que el pie está correctamente colocado, en las opciones del software se debe indicar la distancia vertical que hay desde los marcadores hasta el suelo (en milímetros), de tal manera que el software pueda calcular de manera correcta el origen y no parezca que los objetos e individuos se encuentran en el aire.

Ilustración 30. Pie de calibración

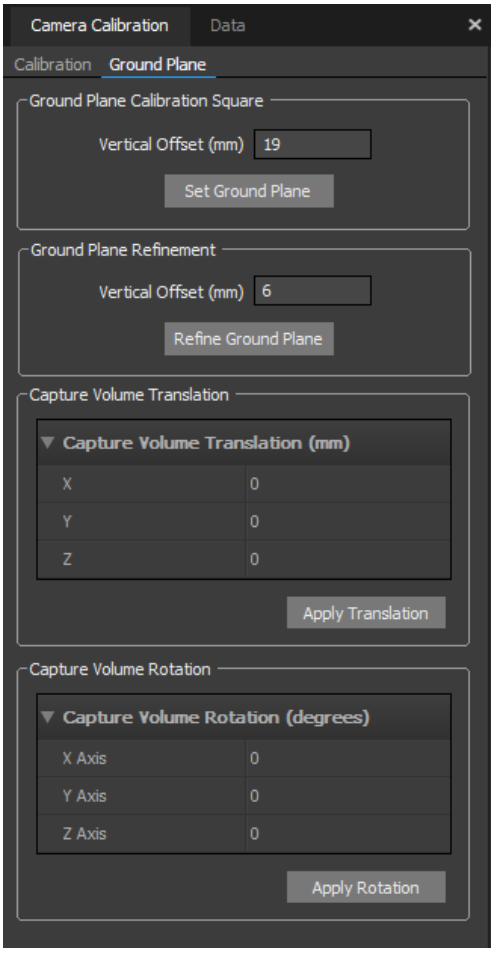


Ilustración 31. Opciones de calibración del pie.

4.3 Objetos sólidos y esqueletos

Completada la calibración está completa, lo siguiente que se debe hacer es crear un objeto sólido o un esqueleto mediante marcadores para realizar el seguimiento. Este también se puede realizar sin crear ninguna entidad, pero el análisis de los datos puede ser más complejo. En este proyecto se ha trabajado en dos modelos diferentes, uno de ellos es un esqueleto y el otro una pesa, en ambos casos se ha generado la forma correspondiente.

Para la pesa, se colocaron 6 marcadores en uno de los discos y se creó el objeto sólido. Para ello, se debe abrir el desplegable de *"Assets"*, opción *"Rigid Body"* y se deben seleccionar los marcadores que se desean incluir en la figura y darle a *"Create"*.

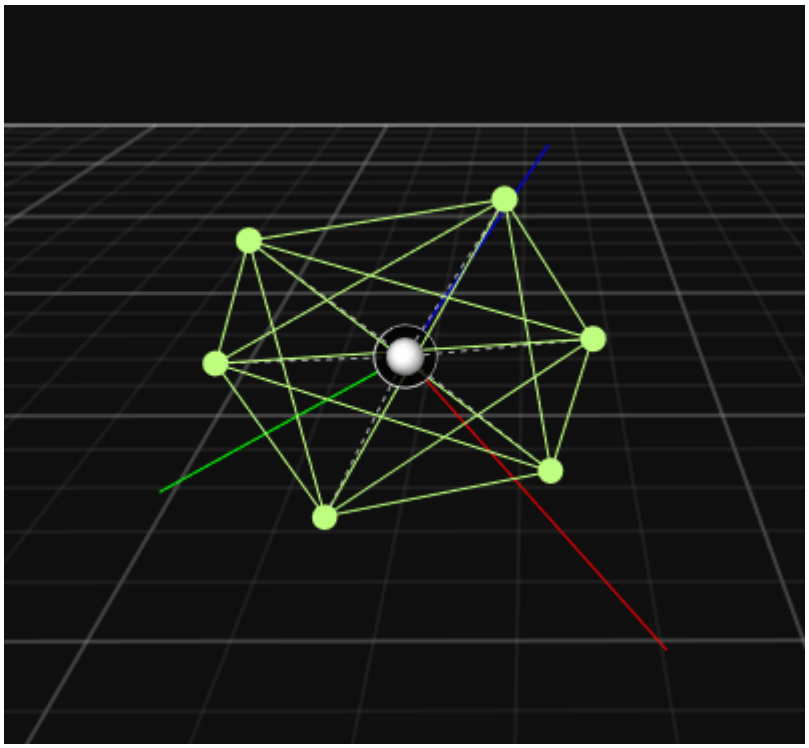


Ilustración 32. Objeto rígido formado por seis marcadores. El círculo blanco es el centro del objeto.

De esta manera, se genera un cuerpo rígido que calcula la posición central en base a todos los marcadores involucrados y la muestra en pantalla como si fuera un marcador nuevo. De esta manera se puede visualizar tanto el centro del objeto como los marcadores reales para que el usuario pueda tener conocimiento de que todo se detecta correctamente.

Para el esqueleto, se ha utilizado un traje con 39 marcadores adheridos. Para la utilización del traje es conveniente que al menos dos personas se encuentren en la sala ya que hay marcadores que se colocan en la espalda y una persona sola es difícil que

los pueda poner con precisión. Además, la persona con el traje debe estar en posición de T a la hora de crear la entidad.

Una vez todos los marcadores están colocados, en el menú de *"Assets"* se debe ir al apartado *"Skeletons"*, se selecciona el tipo de esqueleto que se desea generar (depende de la cantidad de marcadores a utilizar), y se selecciona *"Create"*.

Con el esqueleto creado, hay diferentes maneras de visualizarlo, mediante el mapa de puntos, mediante una simulación de huesos o con forma de maniquí por piezas. En cualquiera de estos casos, es posible editar el aspecto de la visualización.

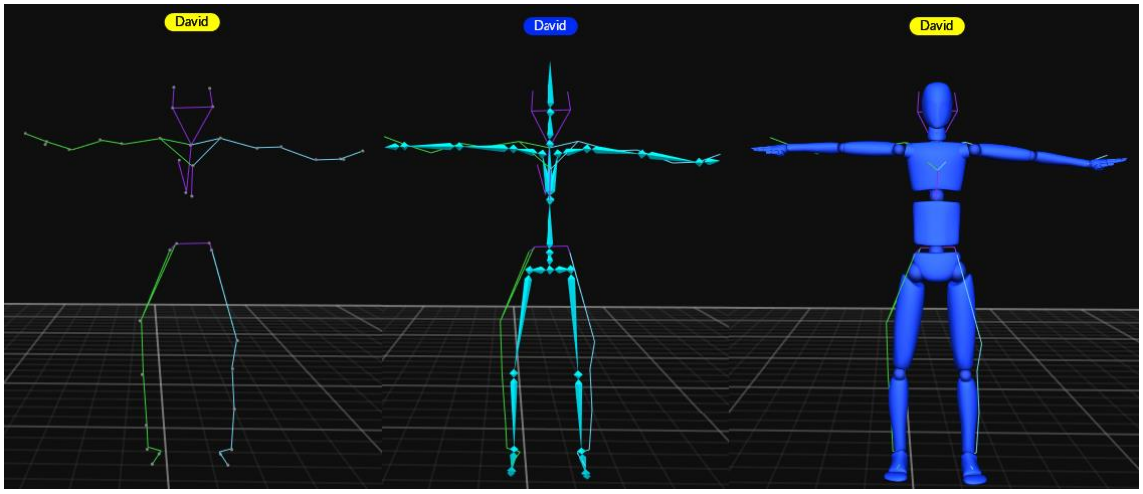


Ilustración 33. Esqueleto visualizado de las tres maneras posibles. El primero es un mapa de puntos, el siguiente la simulación de huesos y el último un maniquí.

4.4 Grabación

Con las entidades ya creadas se procede a grabar diferentes tomas usando de manera individual el cuerpo rígido y posteriormente solo el esqueleto. Cuando la grabación concluye, se abre un editor de vídeo en el cual puedes avanzar y retroceder de fotogramas y seleccionar cuáles te interesan. De esta manera, se pueden seleccionar fragmentos que sean de interés y descartar aquellos en los que el seguimiento no fuera correcto o simplemente no ocurriera nada útil.

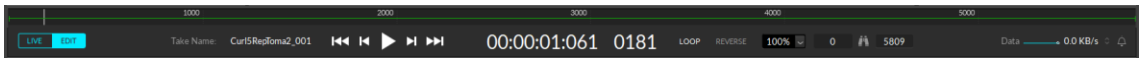


Ilustración 34. Herramientas de edición de grabaciones.

4.5 Exportación

Realizada la grabación, se procede a la exportación de los datos. En el menú de *"Data Pane"* se selecciona el vídeo que se quiere exportar y se abre una ventana emergente donde se puede seleccionar el tipo de archivo que se desea. En el caso de

este proyecto, el archivo que interesa es el .csv que contiene la información de la ubicación de los marcadores en cada fotograma. Para una mayor personalización a la hora de exportar, se puede seleccionar la longitud del vídeo (fotograma de inicio y fotograma de fin), marcadores, cuerpos rígidos, esqueletos, tipo de rotación, unidad de los datos y datos del software.

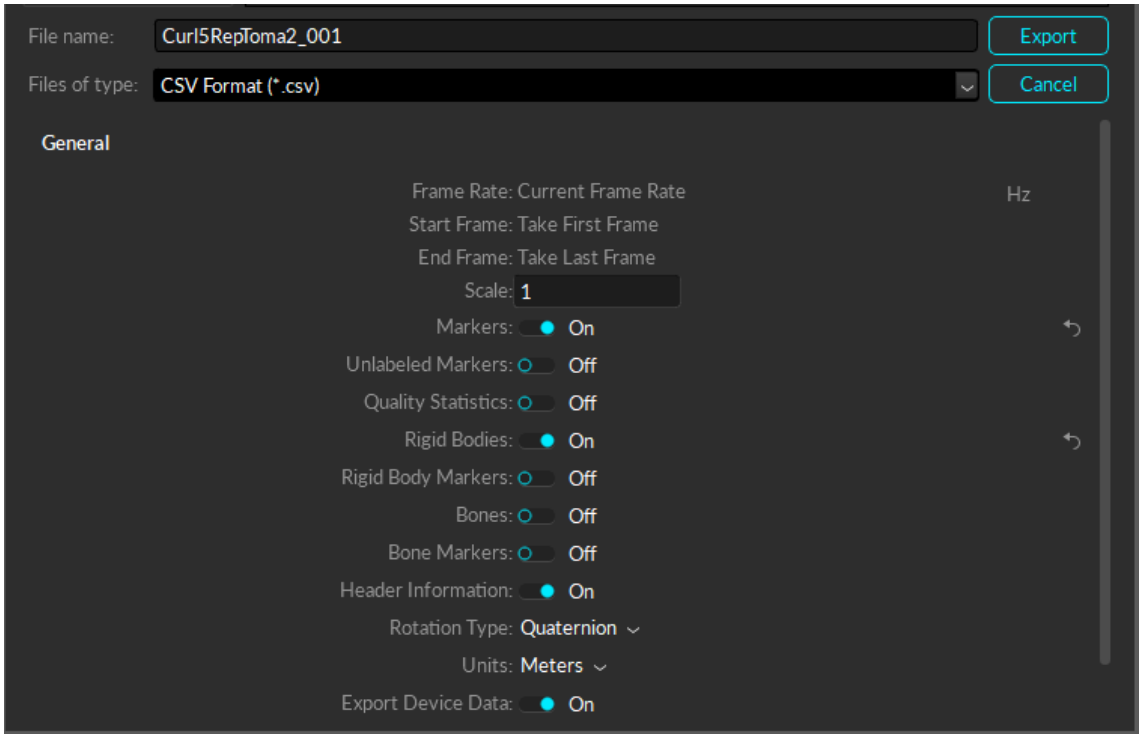


Ilustración 35. Menú de exportación de una grabación como archivo .csv.

4.6 Archivo .csv

Una vez exportada la grabación y obtenido el archivo .csv, es conveniente abrirlo para comprobar que los datos son coherentes y que los marcadores, figuras y esqueletos exportados son los correctos.

Ilustración 36. Archivo .csv obtenido tras la exportación.

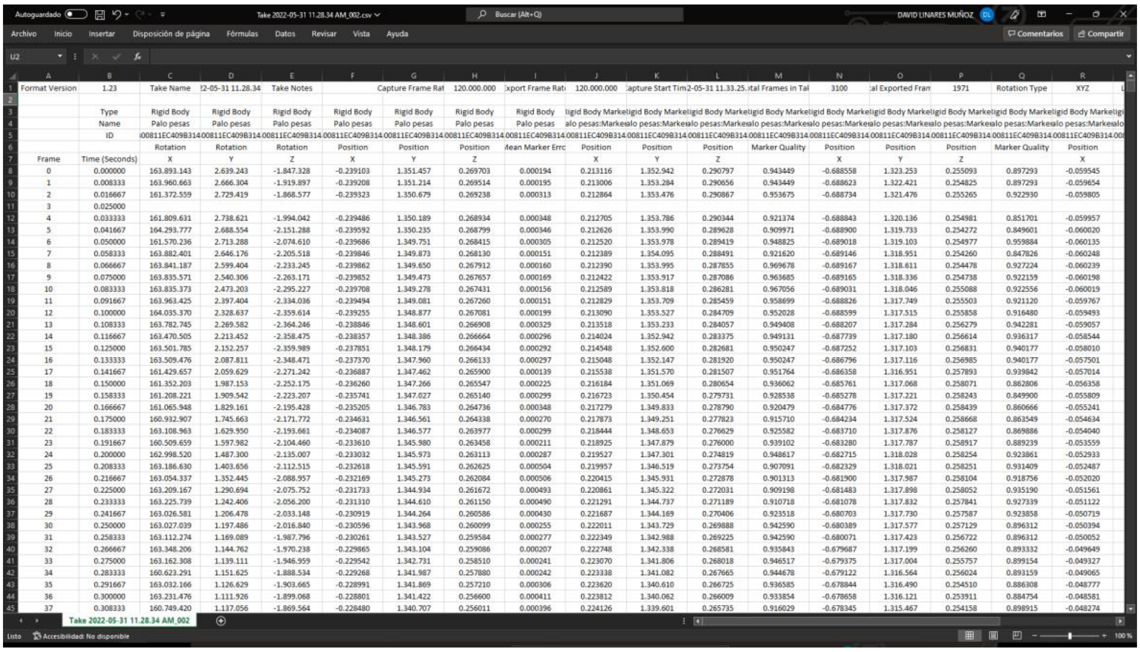


Ilustración 37. Archivo .csv visualizado en columnas.

A continuación, se va a realizar un breve análisis de la información correspondiente a las filas previas al inicio de los datos de los marcadores:

- En la primera fila se encuentra la información relativa al software, podemos encontrar datos como la velocidad de grabación y exportación de captura de fotogramas utilizada, longitud original del vídeo, longitud exportada del vídeo, tipo de rotación, tipo de coordenadas y

unidades utilizadas. La segunda fila se encuentra vacía por lo que se puede omitir.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Format Version	1.23	Take Name	Take 2022-05-31 11.28.34	Take Notes		Capture Frame Rate	120	Export Frame Rate	120	Capture Start Time	2022-05-31 11.33.25.362	Total Frames in Take	3100	Total Exported Frames	1971	Rotation Type	XYZ	Length Units	Meters	Coordinate Space	Global
2																						

Ilustración 38. Información correspondiente a la primera y segunda fila del archivo .csv.

- La tercera fila indica el tipo de datos que se muestra (marcadores sueltos, marcadores de objeto rígido o marcadores de esqueleto).
- La cuarta fila contiene el nombre asignado a cada marcador de los mencionados en la fila 3.
- La quinta fila contiene los identificadores únicos de cada marcador.
- La sexta fila indica el tipo de dato que se va a registrar en esa fila (rotación, posición o media del error del marcador).
- En la séptima fila comienza la identificación de los fotogramas, el segundo en el que suceden y los ejes correspondientes a cada dato (X, Y, Z para la posición y esos tres junto con W para la rotación. El error medio no tiene eje).
- De la fila ocho en adelante se muestran los datos del número de fotograma, tiempo (en segundos) y posición en el espacio de los datos que se mencionan arriba.

3		Type	Rigid Body	Rigid Body	Rigid Body	Rigid Body	Rigid Body	Rigid Body	Rigid Body	Rigid Body Marker	Rigid Body Marker	Rigid Body Marker	Rigid Body Marker	Rigid Body Marker	Rigid Body Marker	Rigid Body Marker	Rigid Body Marker
4		Name	Palo pesas	Palo pesas	Palo pesas	Palo pesas	Palo pesas	Palo pesas	Palo pesas	Palo pesas:Marker 1	Palo pesas:Marker 1	Palo pesas:Marker 1	Palo pesas:Marker 1	Palo pesas:Marker 2	Palo pesas:Marker 2	Palo pesas:Marker 2	Palo pesas:Marker 2
5		ID	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A	7635EBA3B00811EC409B31455EA8619A
6			Rotation	Rotation	Rotation	Position	Position	Position	Mean Marker Error	Position	Position	Position	Marker Quality	Position	Position	Position	Marker Quality
7	Frame	Time (Seconds)	X	Y	Z	X	Y	Z		X	Y	Z		X	Y	Z	
8	0	0.000000	163.893.143	2.639.243	-1.847.328	-0.239103	1.351.457	0.269703	0.000194	0.213116	1.352.942	0.290797	0.943449	-0.688558	1.323.253	0.255093	0.897293
9	1	0.008333	163.960.663	2.666.304	-1.919.897	-0.239208	1.351.214	0.269514	0.000195	0.213006	1.353.284	0.290656	0.943449	-0.688623	1.322.421	0.254825	0.897293
10	2	0.016667	161.372.559	2.729.419	-1.868.577	-0.239323	1.350.679	0.269238	0.000313	0.212864	1.353.476	0.290867	0.953675	-0.688734	1.321.476	0.255265	0.922930
11	3	0.025000															
12	4	0.033333	161.809.631	2.738.621	-1.994.042	-0.239486	1.350.189	0.268934	0.000348	0.212705	1.353.786	0.290344	0.921374	-0.688843	1.320.136	0.254981	0.851701
13	5	0.041667	164.293.777	2.688.554	-2.151.288	-0.239592	1.350.235	0.268799	0.000346	0.212626	1.353.990	0.289628	0.909971	-0.688900	1.319.733	0.254272	0.849601
14	6	0.050000	161.570.236	2.713.288	-2.074.610	-0.239686	1.349.751	0.268415	0.000305	0.212520	1.353.978	0.289419	0.948825	-0.689018	1.319.103	0.254977	0.959884
15	7	0.058333	163.882.401	2.646.176	-2.205.518	-0.239846	1.349.873	0.268130	0.000151	0.212389	1.354.095	0.288491	0.921620	-0.689146	1.318.951	0.254260	0.847826
16	8	0.066667	163.841.187	2.599.404	-2.233.245	-0.239862	1.349.650	0.267912	0.000160	0.212390	1.353.995	0.287855	0.969678	-0.689167	1.318.611	0.254478	0.927224
17	9	0.075000	163.835.571	2.540.306	-2.263.171	-0.239852	1.349.473	0.267657	0.000169	0.212422	1.353.917	0.287086	0.963685	-0.689165	1.318.336	0.254738	0.922159
18	10	0.083333	163.835.373	2.473.203	-2.295.227	-0.239708	1.349.278	0.267431	0.000156	0.212589	1.353.818	0.286281	0.967056	-0.689031	1.318.046	0.255088	0.922556
19	11	0.091667	163.963.425	2.397.404	-2.334.036	-0.239494	1.349.081	0.267260	0.000151	0.212829	1.353.709	0.285459	0.958699	-0.688826	1.317.749	0.255503	0.921120
20	12	0.100000	164.035.370	2.328.637	-2.359.614	-0.239255	1.348.877	0.267081	0.000199	0.213090	1.353.527	0.284709	0.952028	-0.688599	1.317.515	0.255858	0.916480
21	13	0.108333	163.782.745	2.269.582	-2.364.246	-0.238846	1.348.601	0.266908	0.000329	0.213518	1.353.233	0.284057	0.949408	-0.688207	1.317.284	0.256279	0.942281
22	14	0.116667	163.470.505	2.213.452	-2.358.475	-0.238357	1.348.386	0.266664	0.000296	0.214024	1.352.942	0.283375	0.949131	-0.687739	1.317.180	0.256614	0.936317
23	15	0.125000	163.501.785	2.152.257	-2.359.989	-0.237851	1.348.179	0.266434	0.000292	0.214548	1.352.600	0.282681	0.950247	-0.687252	1.317.103	0.256831	0.940177
24	16	0.133333	163.509.476	2.087.811	-2.348.471	-0.237370	1.347.960	0.266133	0.000297	0.215048	1.352.147	0.281920	0.950247	-0.686796	1.317.116	0.256985	0.940177
25	17	0.141667	161.429.657	2.059.629	-2.271.242	-0.236887	1.347.462	0.265900	0.000139	0.215538	1.351.570	0.281507	0.951764	-0.686358	1.316.951	0.257893	0.939842
26	18	0.150000	161.352.203	1.987.153	-2.252.175	-0.236260	1.347.266	0.265547	0.000225	0.216184	1.351.069	0.280654	0.936062	-0.685761	1.317.068	0.258071	0.862806
27	19	0.158333	161.208.221	1.909.542	-2.223.207	-0.235741	1.347.027	0.265140	0.000299	0.216723	1.350.454	0.279731	0.928538	-0.685278	1.317.221	0.258243	0.849900
28	20	0.166667	161.065.948	1.829.161	-2.195.428	-0.235205	1.346.783	0.264736	0.000348	0.217279	1.349.833	0.278790	0.920479	-0.684776	1.317.372	0.258439	0.860666
29	21	0.175000	160.932.907	1.745.663	-2.171.772	-0.234631	1.346.561	0.264338	0.000270	0.217873	1.349.251	0.277823	0.915710	-0.684234	1.317.524	0.258668	0.863549
30	22	0.183333	163.108.963	1.629.950	-2.193.661	-0.234087	1.346.577	0.263977	0.000299	0.218444	1.348.653	0.276629	0.925582	-0.683710	1.317.876	0.258127	0.869886

Ilustración 39. Información correspondiente desde la fila tres en adelante del archivo .csv.

Capítulo 5. Procesamiento de datos en Matlab

Este capítulo detallará la programación llevada a cabo para la obtención de los resultados.

5.1 Descripción

En este proyecto se ha desarrollado dos programas de Matlab, el primero orientado a la obtención de parámetros utilizados por fisioterapeutas y entrenadores; y el segundo orientado a la representación 3D de los datos obtenidos.

5.2 LiveScript de obtención de parámetros

El primer programa realiza un análisis de los datos obtenidos de la grabación del ejercicio. Asimismo, se obtienen parámetros como el desplazamiento, velocidad, aceleración, fuerza y potencia del ejercicio y a su vez, se obtienen gráficas representativas para que se puedan ver los resultados de manera visual.

5.2.1 Parámetros generales

En esta sección del script, se le pide al usuario que introduzca el peso en kilogramos que se ha utilizado a la hora de realizar el ejercicio. Además, se introduce la velocidad de fotogramas por segundo a la que graban las cámaras de la sala MoCap utilizada y se lee el fichero de datos obtenido de la grabación. Por último, se obtiene el número total de fotogramas que contienen los datos y se obtiene la variable *"time"* que es el tiempo en segundos que dura la grabación y será el eje X de las gráficas.

Datos generales y lectura de tablas

```
m = input("Introduce el peso (en kg) del ejercicio realizado: ");
fs = 120; %Número de FPS
T = readtable('21.csv'); %Leemos los datos
R = table2array(T); %Convertimos los datos a array
long = length(R);%Número total de FPS
time = 0:(1/fs):(long/fs)-(1/fs);
```

Código 1. Obtención de datos generales y lectura de tablas.

5.2.2 Localización y almacenamiento de la variable Y

En esta parte de la programación, se muestra por pantalla el nombre de cabecera de las columnas de la tabla creada anteriormente. Esto se hace para que el usuario tenga una respuesta visual de los títulos y a continuación pueda seleccionar

cual es el que se quiere interpretar. Una vez el usuario introduce el valor deseado, el programa almacena dicha columna en un array únicamente de datos, es decir, sin cabecera.

```
Entrada de la variable Y y almacenamiento en columna
7   T.Properties.VariableNames %Mostramos los nombres de las columnas
8   Yvariable = input("Selecciona la posición en la que se " + ...
9       "encuentra la variable Y de posicionamiento (no rotación): ");
10  Yaxis = R(:,Yvariable); %Extraemos la variable Y
11
```

Código 2. Selección la columna de datos.

5.2.3 Obtención de la gráfica de desplazamiento

Para el cálculo del desplazamiento no hace falta llevar a cabo ninguna operación ya que se puede deducir de manera trivial mediante el posicionamiento de datos. Por tanto, se puede pasar directamente a la obtención de la gráfica.

Lo primero que se ha utilizado es la función *"findpeaks"*. Esta función permite encontrar los valores máximos de una gráfica. Permite como parámetros de entrada la velocidad de fotogramas por segundo, cuya función es realizar un cálculo automático y que la representación sea en la escala de tiempos. Asimismo, se añade el argumento *"MinPeakDistance"* que se utiliza para evitar la detección de máximos muy próximos entre sí.

Se prosigue estableciendo los títulos de los ejes para que la información quede plasmada de manera más visual. El siguiente paso es suavizar la forma de la gráfica, para reducir el ruido de los datos mediante la función *"smoothdata"* y posteriormente se representa del mismo modo que al comienzo.

Por último, en este apartado, se vuelve a utilizar la función *"findpeaks"* pero en este caso no hará una representación, si no que se almacenarán los valores en las variables 'vmin' y 'tmin' que son respectivamente el valor de los mínimos y sus instantes temporales. Una vez obtenido los datos, se representan superpuestos a las gráficas y en forma de 'O' de manera que rodeen los valores mínimos.

Desplazamiento

```
12  findpeaks(Yaxis, 120, 'MinPeakDistance', 2)%Ploteamos la variable Y
13  % con frecuencia de muestro 120Hz (120frames/s) y le establecemos la
14  % distancia mínima de picos en 2 segundos (suficiente para obtener
15  % una repetición) para que las irregularidades en el movimiento no
16  % afecten a los picos.
17  xlabel('Tiempo (s)')
18  ylabel('Altura (m)')
19  title('Gráfica con máximos y mínimos de las repeticiones')
20  hold on
21  c = smoothdata(Yaxis,"SmoothingFactor",0.001);
22  findpeaks(c, fs, 'MinPeakProminence', 0.4)
23  %figure
24  [vmin, tmin] = findpeaks(-c, fs, 'MinPeakProminence', 0.4);
25  plot(tmin, c(round(tmin*fs)), 'o', 'Color', [0 0 0])
26  hold off
```

Código 3. Representación del desplazamiento.

La función *"hold on"* se utiliza para superponer gráficas, si no se utilizara (estaría activa la función *"hold off"*), las gráficas aparecerían en figuras separadas.

5.2.4 Velocidad

Una vez se tiene el array de datos de desplazamiento, es posible calcular la velocidad instantánea realizada en cada momento y en conjunto obtener una gráfica de la velocidad a la que se lleva a cabo el ejercicio.

La velocidad es la derivada de la posición respecto al tiempo, es decir, el límite del incremento del espacio partido el incremento de tiempo cuando este último tiende a cero. En un sistema discreto la aproximamos tomando como incremento de espacio la diferencia entre la posición siguiente al instante actual y la posición anterior al instante actual y como incremento de tiempo, el tiempo transcurrido entre estas dos posiciones, que es dos veces el periodo de muestreo.

$$v \equiv \frac{dr}{dt} = \lim_{\Delta t \to 0} \frac{\Delta r}{\Delta t} \tag{1}$$

$$v(i) = \frac{r(i+1) - r(i-1)}{2 * \frac{1}{fs}} \tag{2}$$

En este caso concreto, se está calculando la velocidad como el espacio que se recorre alrededor del instante concreto que queremos calcular entre el tiempo.

Para aplicar en Matlab la ecuación a todo el vector de posiciones de la serie, lo hacemos mediante la siguiente instrucción:

$$v = \frac{r(3{:}end) - r(1{:}end-2)}{2 * \frac{1}{f_s}} \quad (3)$$

El primer array de desplazamiento se recorre desde la tercera posición hasta el final y el segundo array desde la primera hasta el final menos dos posiciones. Aplicando la fórmula conforme aparece, es decir, con la posición tres y la posición uno, calcularíamos cual es la posición dos, por tanto, el resto de los valores se calculan igual. Se realiza así ya que la posición uno no tiene un instante de tiempo anterior ni un posicionamiento previo por tanto es imposible de calcular. Del mismo modo sucede con la última muestra, no se puede calcular ya que no conocemos que pasará en el instante posterior al último que tenemos. Por tanto, el array de la velocidad será dos datos más pequeño que el de desplazamiento.

$$v_{ajust} = [v(1);\ v;\ v(end)] \quad (4)$$

Para disponer de un vector velocidad con el mismo número de elementos que el vector posición, se hace uso de la ecuación 4, en la que se duplica el primer valor "v(1)" y el último "v(end)", esto causa que el array de velocidad tenga las dimensiones correctas, pero tenga dos datos que son suposiciones en base a las posiciones calculadas.

Calculada la velocidad, se aplica un suavizado para reducir el ruido. Una vez obtenidas ambas velocidades, la normal y la suavizada, se muestran en una única gráfica. Como en el caso anterior, se le pone título a la gráfica y nombre y unidades a los ejes para tener una información completa y visual.

Velocidad

```
27  v = (r(3:end) - r(1:end-2)) / (2*(1/fs)); %Cálculo de la velocidad
28  vajust = [v(1); v; v(end)]; %Ajuste de datos
29  vsmooth = smoothdata(vajust,"SmoothingFactor",0.00001);
30  figure, plot(time,vajust,'b'), hold on, plot(time,vsmooth,'r')
31  xlabel('Tiempo (s)')
32  ylabel('Velocidad (m/s)')
33  title('Gráfica de la velocidad en el eje Y')
```

Código 4. Obtención de la velocidad y representación.

5.2.5 Aceleración

De manera análoga al cálculo de la velocidad, se puede calcular la aceleración. La diferencia en el cálculo reside en que la aceleración es la variación de la velocidad por unidad de tiempo, el resto es prácticamente igual.

$$a \equiv \frac{dv}{dt} = \lim_{\Delta t \to 0} \frac{\Delta v}{\Delta t} \quad (5)$$

Teniendo esta ecuación, y sabiendo que el cálculo es muy similar al de la velocidad, podemos obtener la ecuación que se usará para calcular el array.

$$a = \frac{v_{smooth}(3{:}end) - v_{smooth}(1{:}end-2)}{2 * \frac{1}{f_s}} \quad (6)$$

Al igual que en el cálculo de la velocidad, aquí también se aplica la regla del posterior y el anterior. Asimismo, se divide por el tiempo multiplicado por dos ya que se recorren dos intervalos temporales.

$$a_{ajustada} = [a(1);\ a; a(end)] \quad (7)$$

Por último, se ajusta la aceleración haciendo uso de la ecuación 7 del mismo modo que se hace con la velocidad, copiando de nuevo el primer y el último valor en sus respectivas posiciones.

Una vez calculada la aceleración, se le aplica un suavizado como en el caso de la velocidad, para obtener una gráfica más definida. Una vez obtenidas ambas aceleraciones, la normal y la suavizada, se muestran en una única gráfica. Como en el caso anterior, se le pone título a la gráfica y nombre y unidades a los ejes para tener una información completa y visual.

Aceleración

```
34  a = (vsmooth(3:end) - vsmooth(1:end-2)) / (2*(1/fs)); %Cálculo de la aceleración
35  aajust = [a(1); a; a(end)]; %Ajuste de datos
36  asmooth = smoothdata(aajust,"SmoothingFactor",0.00001);
37  figure, plot(time,aajust,'b'), hold on,plot(time,asmooth,'r')
38  xlabel('Tiempo (s)')
39  ylabel('Aceleración (m/s^2)')
40  title('Gráfica de la aceleración en el eje Y')
```

Código 5. Obtención de la aceleración y su representación.

5.2.6 Fuerza

Obtenida la aceleración, es posible calcular la fuerza. Esta modifica el movimiento de un cuerpo (acelerarlo, frenarlo, cambiar el sentido, etc.) o lo puede deformar. En este caso, se modifica el movimiento ya que se levanta y se baja una pesa. La fuerza se puede expresar como:

$$F = m * a \quad (8)$$

Como es una fórmula sencilla y tenemos los datos, se calcula directamente sin necesidad de hacer operaciones extra. Como en los casos anteriores, se le pone título a la gráfica y nombre y unidades a los ejes para tener una información completa y visual.

5.2.7 Potencia

Con la fuerza calculada, la potencia viene dada por la expresión:

$$P = F * v \tag{9}$$

Para llevar a cabo este cálculo en Matlab, se debe utilizar el operador ".*", que multiplica elemento por elemento dos vectores. En el caso de la fuerza no ha hecho falta ya que la masa es un valor constante, pero la velocidad varía a cada instante, por tanto, se debe aplicar el operador para realizar el cálculo correcto.

Una vez calculada la potencia, realizamos el mismo proceso que con la fuerza, suavizamos los datos, y representamos en la misma figura las dos potencias calculadas para poder compararlas.

5.3 Scripts de representación del esqueleto

La programación se compone de dos scripts. En términos generales, uno obtiene los datos necesarios para la representación y otro se encarga de representar en 3D esos datos.

5.3.1 Obtención de datos

Para la obtención de datos se ha creado una función que necesita como parámetro de entrada el nombre del archivo .csv que se quiera representar, los datos que se obtengan se guardarán en la variable DataM. Dentro de esta función, es donde se realiza todo el procesado de datos, es decir, que se crea simplemente para facilitar su uso. Esta se encuentra dividida en secciones para diferenciar los procesos que se realizan.

Como se ha explicado anteriormente, los ficheros .csv que se exportan desde el software tienen la misma disposición. Por tanto, es posible realizar un programa concreto para la representación 3D de los esqueletos.

Conociendo esta información se puede programar la sección de obtención de datos. En ella se lee el fichero y se almacena toda la información en una única columna, tras esto, y sabiendo a partir de que posiciones comienzan los datos, se obtienen diferentes valores que se utilizan para obtener los fotogramas por segundo, los nombres de los marcadores, el número total de fotogramas y el número total de marcadores a representar.

```
%% Obtención de la tabla
FullFile = textread([FileName '.csv'],'%s','delimiter',',');
    %Coge todos los elementos del documento y genera una matriz columna con ellos.
DataStart = length(textread([FileName '.csv'],'%s','delimiter',',','headerlines',1));
    %Determina la longitud desde la fila 2 del documento hasta el final.
    %Se pone un 1 al final ya que matlab empieza a contar desde 0.
NoMarkers = length(textread([FileName '.csv'],'%s','delimiter',',','headerlines',3));
    %Determina la longitud desde la fila 4 del documento hasta el final.
    %Se pone un 3 al final ya que matlab empieza a contar desde 0.
Data = textread([FileName '.csv'],'%f','delimiter',',','headerlines',7);
    %Determina la longitud desde la fila 8 del documento hasta el final.
    %Se pone un 7 al final ya que matlab empieza a contar desde 0.

Header = FullFile(1: (length(FullFile)-DataStart) );
    %Selecciona los datos de cabecera.
FrameRate = str2double(Header(8));
    %Coge el valor 8 de Header que corresponde al valor de los FPS
MarkersL = FullFile( (length(FullFile) - DataStart + 3) : (length(FullFile)-NoMarkers) );
    %Selecciona los nombres asignados de los marcadores
NFrames = str2double(Header(16));
    %Coge el valor 16 de Header que corresponde al valor total de frames
NMarkers = length(MarkersL)/3;
    %Determina el número de marcadores utilizados, divide entre 3 ya que
    %cada marcador tiene 3 dimensiones y si no aparecerían por triplicado.
```

Código 6. Obtención de valores e información relativa a la grabación.

La siguiente sección se corresponde con la extracción de datos a una matriz. Para ello se crea una matriz de ceros donde se almacenará la información del tiempo, fotogramas y posición de cada marcador. Tras esto, se almacenan los datos dentro de la matriz y se transpone para obtenerlos en columnas. Luego se almacenan los fotogramas y el tiempo en sus respectivas variables.

```
%% Extracción de datos crudos a una matriz
Data3D = zeros((NMarkers*3)+2, NFrames);
    %Crea una matriz de ceros de (NMarkers*3)+2 filas siendo
    %NMarkers*3 para todas las dimensiones de marcadores y el +2 para
    %incluir el ' ' y 'Type' que aparecen al inicio y de NFrames siendo
    %estas el total de frames.

Data3D (1:length(Data)) = Data;
    %Asigna los datos a la matriz Data3D

Data3D = Data3D';
    %Todos los datos crudos por columnas (como en el doc original)

TFrames = Data3D(:, 1);
    %Vector de frames (0 a NFrames)
Time = Data3D(:, 2);
    %Vector de tiempos en segundos, frame a frame
```

Código 8. Extracción de las posiciones de los marcadores en cada fotograma. Se almacena todo en una única dimensión.

En la sección tres, se crea una matriz tridimensional para almacenar los datos y luego poder hacer la representación 3D. Se almacenan los datos mediante un bucle que recorre los fotogramas y otro bucle dentro del primero que recorre los marcadores de tres en tres para almacenar siempre la misma variable (x, y, z) en su correspondiente dimensión de la matriz.

```
%% Matriz 3D únicamente con marcadores

Markers = zeros(NFrames, NMarkers, 3);
    %Crea una matriz 3D con el nº de frames,
    %el nº de marcadores y las 3 dimensiones. (Para luego plotear).

for f=1:NFrames %Recorre todos los frames (filas)
    c = 1; %Contador
    for x=3:3:(NMarkers*3)+2 %Recorre todos los marcadores de 3 en 3
        Markers(f,c,1) = Data3D(f,x);
        Markers(f,c,3) = Data3D(f,x+1); %Y de Optitrack es Z
        Markers(f,c,2) = -1*Data3D(f,x+2); %Z de Optitrack es -Y
        c = c+1;
    end
end
    %Bucle para rellenar la matriz 3D con los datos correspondientes a cada
    %frame y su instante de tiempo.
```

Código 7. Organización de los datos obtenidos anteriormente en una matriz tridimensional.

En la cuarta sección se realiza la detección de las entidades. En este fragmento de código se lleva a cabo la extracción de los nombres de los marcadores y se detecta cuantas entidades diferentes aparecen. Esto se utilizará posteriormente para la identificación y representación de las posiciones de los marcadores en 3D.

```
%% Encontrar Entidades

DataL = textread([FileName '.csv'],'%s','delimiter',',','headerlines',3);
    %Coge la fila de marcadores del documento
DataL = DataL(3:(NMarkers*3)+2);
    %Los nombres de todos los marcadores, 3 veces (coord. X, Y, Z)

contNames = 0;
stringName = " ";

for NMS=1:length(DataL) %Bucle con longitud del total de marcadores
    dp = strfind(DataL{NMS},':'); %Detecta el nombre de la entidad
    SetName = DataL{NMS}(1:dp-1); %Almacena el nombre de la entidad
    Comp = strcmp(SetName,stringName); %Compara para ir guardando otras entidades
    if Comp==0 %Si los nombres son diferentes entra al if
        contNames = contNames+1; %Añade 1 a contNames para contar otros diferentes
        stringName = SetName; %Crea una variable para almacenar el nombre sin
                              %borrar el de SetName que se sigue usando en el bucle
        SetNames{contNames} = stringName; %Acumula todos los nombres
    end
end

NSets = length(SetNames); %Número de entidades diferentes

for set=1:NSets %Bucle para separar las entidades con sus marcadores
    SetN = SetNames{set}; %Guarda el nombre de la entidad
    mks = strfind(DataL,SetN); %Asigna el valor 1 a los marcadores que correspondan
                               %con el nombre de la entidad
    mks = cell2mat(mks); %Se queda solo con los marcadores de valor 1
    NMark.(sprintf('%s',char(SetN))) = sum(mks); %Total de nombres encontrados en una entidad
end

c = 1;
for l=1:3:length(DataL)
    Labels(c,1) = DataL(l); %Todas las etiquetas sin repetirse
    c = c+1;
end
```

Código 9. Localización las entidades y almacenamiento de los nombres de sus marcadores.

En la última sección se almacena la información en la variable DataM. Esta variable es una estructura que se compone de cuatro elementos, la velocidad de fotogramas por segundo, el vector tiempo, el número de fotogramas y otra estructura llamada Sets. La estructura sets almacena estructuras de entidades, esto quiere decir que si se han grabado dos esqueletos generará dos estructuras ya que el análisis de datos los habrá diferenciado por el nombre de los marcadores. Las estructuras de las entidades contienen dos parámetros, uno es el número de marcadores y otro es una estructura que contiene los marcadores con sus tres dimensiones y la posición en cada instante.

```
%% Organizar marcadores en la estructura

set_a = 0;
% mks_a = 0;

for set=1:NSets
    SetN = SetNames{set}; %Guarda el nombre de la entidad
    Labels=strrep(Labels,[SetN ':'],''); %Guarda el nombre de los marcadores
                                         %a partir de los :
    mks = NMark.(sprintf('%s',char(SetN)))/3; %Guarda el nº de marcadores sin
                                              %repetir de cada entidad
    for l=set_a+1:mks+set_a %Bucle para guardar los datos de las entidades
        for f=1:NFrames %Guarda los datos correspondientes a cada frame
            DataM.Sets.(sprintf('%s',char(SetN))).Raw.(sprintf('%s',char(Labels(l))))(f,1) = Markers(f,l,1);
            DataM.Sets.(sprintf('%s',char(SetN))).Raw.(sprintf('%s',char(Labels(l))))(f,2) = Markers(f,l,2);
            DataM.Sets.(sprintf('%s',char(SetN))).Raw.(sprintf('%s',char(Labels(l))))(f,3) = Markers(f,l,3);
        end
    end
    set_a = mks;
    DataM.Sets.(sprintf('%s',char(SetN))).NMarkers = mks;
end

DataM.FrameRate = FrameRate; %Guarda los FPS
DataM.Time = Time'; %Guarda el tiempo
DataM.Frames = NFrames; %Guarda el nº de frames
```

Código 10. Inserción de los datos en la variable DataM.

5.3.2 Representación 3D

Para la representación 3D del esqueleto se ha creado otra función que tendrá como parámetros de entrada los datos almacenados en DataM y la variable FPI que indica el número de fotogramas que se avanzan cada vez que se representa.

Para ello se deben llevar a cabo varios procesos. Primero un almacenamiento de datos mediante bucles para determinar las entidades a representar y sus respectivos marcadores con las tres dimensiones. Después se crea el entorno 3D y se establecen los límites de los ejes. Por último, se establece el bucle que representa los marcadores en la gráfica, el número de fotogramas representados variará en función del valor otorgado a FPI. Esto se hace para que se pueda variar la velocidad de representación pudiendo representar los fotogramas de uno en uno o saltando de diez en diez.

```
%% PLOTMARKERS

NSets = numel(fieldnames(DataM.Sets)); %Guarda el nº de entidades

NMarkersT = 0;
MSets = fieldnames(DataM.Sets); %Guarda el nombre de las entidades

for s=1:NSets %Bucle para coger el número de marcadores totales sin repetir
    SetName = MSets{s};
    NMarkersT = NMarkersT + DataM.Sets.(sprintf('%s',char(SetName))).NMarkers;
end

Markers1 = zeros(DataM.Frames,NMarkersT,3);
    %Crea una matriz 3D de ceros con el nº de frames, el nº de marcadores
    %sin repetir y las 3 dimensiones
counter = 0;

for s=1:NSets %Bucle
    SetName = MSets{s}; %Coge las entidades
    SetNMarkers = DataM.Sets.(sprintf('%s',char(SetName))).NMarkers;
        %Nº de marcadores de la entidad
    Labels = fieldnames(DataM.Sets.(sprintf('%s',char(SetName))).Raw);
        %Guarda el nombre de los marcadores de la entidad
    label = 1; %Contador para recorrer los marcadores de la entidad
    for M=counter+1:counter+SetNMarkers
        Markers1(:,M,1) = DataM.Sets.(sprintf('%s',char(SetName))).Raw.(sprintf('%s',char(Labels(label))))(:,1);
        Markers1(:,M,2) = DataM.Sets.(sprintf('%s',char(SetName))).Raw.(sprintf('%s',char(Labels(label))))(:,2);
        Markers1(:,M,3) = DataM.Sets.(sprintf('%s',char(SetName))).Raw.(sprintf('%s',char(Labels(label))))(:,3);
        label = label+1;
    end
    counter = M;
end
```

Código 11. Almacenamiento de datos mediante bucles en la variables a representar.

```
xmin = min(min(Markers1(:,:,1)))-0.2;
xmax = max(max(Markers1(:,:,1)))+0.2;
if xmax<0.4
    xmax = 0.4;
end
ymin = min(min(Markers1(:,:,2)))-0.2;
ymax = max(max(Markers1(:,:,2)))+0.2;
if ymax<0.4
    ymax = 0.4;
end
zmin = min(min(Markers1(:,:,3)));
if zmin > 0
    zmin = 0;
end
zmax = max(max(Markers1(:,:,3)))+0.2;

ejes = [xmin xmax ymin ymax zmin zmax];

az = -45;
el = 15;
```

Código 12. Definición de los límites del espacio 3D de representación.

```
58  for f=1:FPI:DataM.Frames
59      plot3(Markers1(f,:,1),Markers1(f,:,2),Markers1(f,:,3),'ob')
60      grid on
61      axis equal
62      axis(ejes)
63      view(az,el)
64      xlabel('X')
65      ylabel('Y')
66      zlabel('Z')
67      title(['Frames: ' num2str(f)])
68      drawnow
69      [az,el] = view;
70  end
```

Código 13. Representación de los datos. Genera una figura 3D que se actualiza continuamente mientras queden fotogramas por representar.

Capítulo 6. Resultados

Una vez explicados ambos programas, se van a mostrar los resultados obtenidos. Las pruebas se han realizado con diferentes ficheros de datos obtenidos de vídeos, pero se ha optado por utilizar los dos más representativos. Uno de ellos es un ejercicio de levantamiento de pesas y el otro es una persona realizando movimientos en el interior de la sala.

6.1 Levantamiento de pesas

El ejercicio que se ha analizado ha sido el curl de bíceps, este ejercicio trabaja tanto la parte superior como la inferior del brazo, centrándose especialmente en el bíceps, como indica el nombre del ejercicio.

Para una correcta ejecución se deben colocar los pies separados a la anchura de las caderas, los brazos deben estar relajados y las palmas mirando al frente y no se deben levantar los hombros. Una vez se adquiere una buena postura, hay que tener en cuenta como se debe realizar la respiración. Primero se inspira y se debe levantar la pesa hasta los hombros doblando los codos sin separarlos del cuerpo, una vez arriba se debe bajar la pesa poco a poco mientras se expira. De esta manera se maximiza el trabajo realizado y se lleva a cabo una correcta ejecución.

Ilustración 40. Ejercicio curl de bíceps.

Una vez se ha grabado el ejercicio y se ha obtenido el fichero de datos, se procesa con el programa desarrollado en Matlab para obtener los parámetros. Dicho programa se ejecutará por secciones que quedan determinadas por los títulos, de esa manera no hace falta ejecutar el programa completo si, por ejemplo, solo se quisiera obtener el desplazamiento.

6.1.1 Desplazamiento

Ejecutando la primera sección, el programa pide el peso que se está utilizando para la realización del ejercicio. Esto se muestra por la pantalla de comandos y el parámetro se detectará como introducido cuando el usuario presione *Enter*.

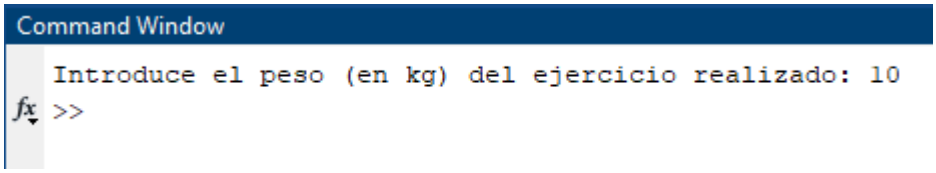


Ilustración 41. Ventana de comandos donde se le está pidiendo al usuario que introduzca el peso.

Una vez hecho esto, se habrá ejecutado al completo la primera sección y los parámetros almacenados serán los siguientes:

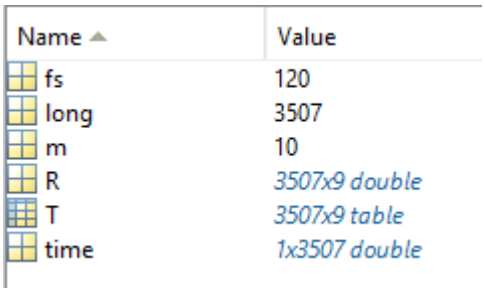


Ilustración 42. Parámetros obtenidos tras la ejecución de la primera sección.

En estos parámetros, *'m'* será el peso introducido por el usuario, *'fs'* es la frecuencia de muestreo (número de fotogramas por segundo), *'long'* será la cantidad de fotogramas y, por tanto, datos que habrá en las arrays, *'T'* contiene los datos que se convertirán en array en la variable *'R'*, por último *'time'* será el tiempo en segundos que dura el vídeo y que se usará para la representación en función del tiempo de las gráficas.

Con esos datos calculados, se ejecuta la siguiente sección, donde se mostrará por pantalla el nombre de cada columna para que el usuario pueda identificar cual es la columna de posicionamiento del eje Y.

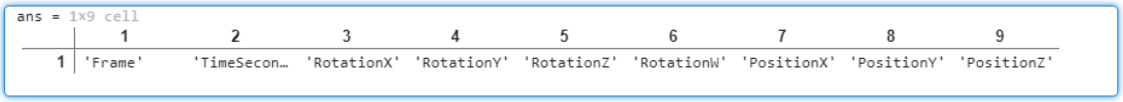


Ilustración 43. Cabecera de las columnas de las cuales el usuario debe elegir la posición Y.

Tras esto se le pide al usuario que introduzca el valor de la columna a analizar y posteriormente se almacena dicha columna en una variable nueva llamada Yaxis.

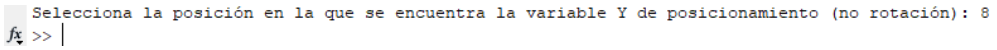
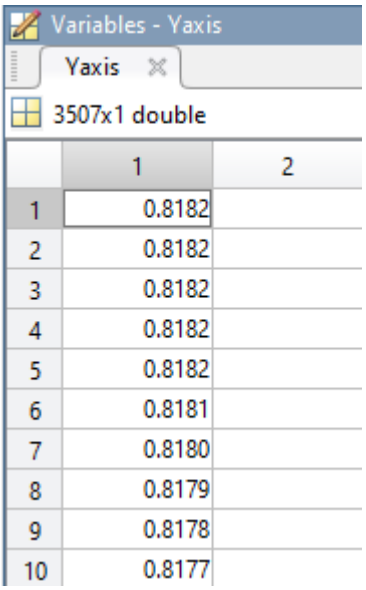


Ilustración 44. Se le pide al usuario que introduzca el número de columna a analizar.



	1	2
1	0.8182	
2	0.8182	
3	0.8182	
4	0.8182	
5	0.8182	
6	0.8181	
7	0.8180	
8	0.8179	
9	0.8178	
10	0.8177	

Ilustración 45. Valores de posición del marcador en el eje Y.

Con todos los datos primarios ya obtenidos podemos pasar al análisis y representación de las gráficas.

Al ejecutar la sección del desplazamiento obtenemos la gráfica con ambas representaciones. Como el desplazamiento y el desplazamiento suavizado son muy similares en la gráfica, a simple vista, solo se podrá ver la parte suavizada.

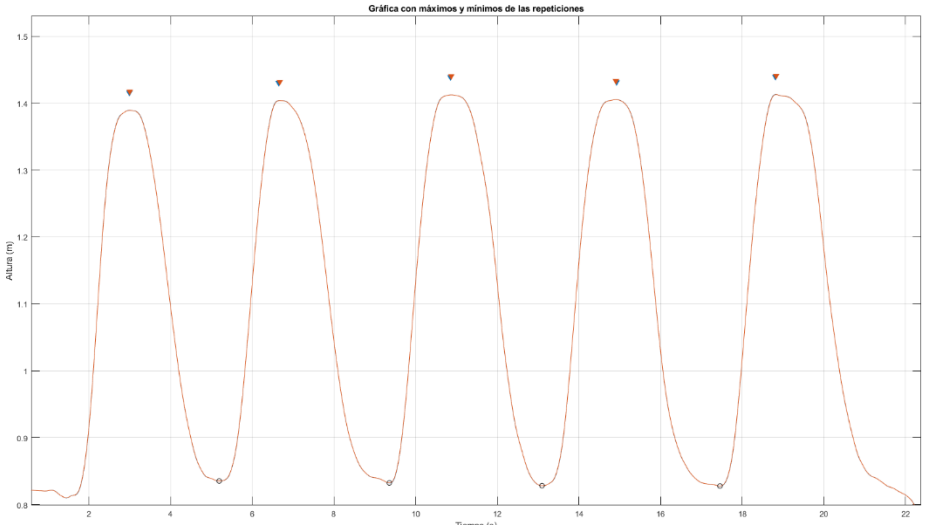


Ilustración 46. Gráfica del desplazamiento y desplazamiento suavizado. Se señalan los máximos y los mínimos en cada repetición del ejercicio.

Como se puede observar en la gráfica, durante el ejercicio se han realizado 5 repeticiones. Se aprecia como los máximos, salvo el primero, y los mínimos son muy similares lo que nos indica que la ejecución del ejercicio en cuanto a desplazamiento en el eje Y es correcta. Ahondando un poco más en los datos, cabe destacar que la escala de la altura no comienza en cero, esto se debe a que el origen de coordenadas se encuentra a nivel del suelo y la gráfica representa de manera real la posición de la pesa.

Si acercamos la gráfica y mostramos únicamente una sección pequeña, en este caso un máximo, se puede ver como la gráfica azul (desplazamiento) es mucho más irregular que la naranja (suavizada). Por tanto, para facilitar la obtención de máximos y mínimos se utiliza la suavizada, el uso del desplazamiento normal podría mostrar muchos máximos o mínimos en zonas muy cercanas debido a esos picos.

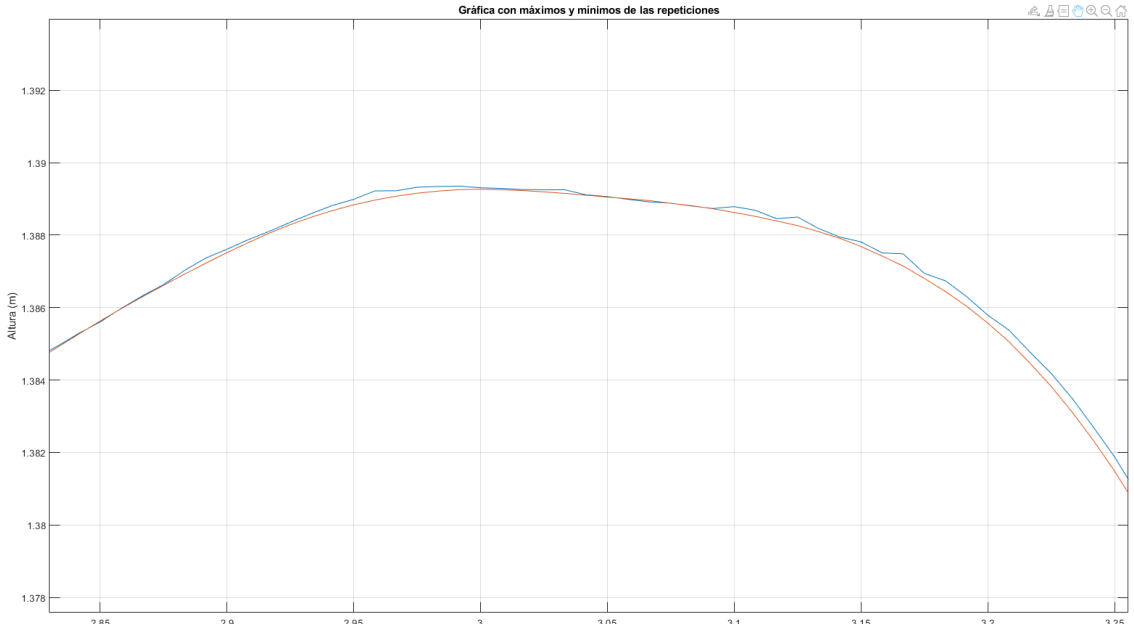


Ilustración 47. Diferencia entre el desplazamiento puro y suavizado.

6.1.2 Velocidad

Ahora se analizará la velocidad de ejecución del ejercicio. Como se mencionaba en el apartado de programación, a la hora de calcular la velocidad, se copiaba el primer y último dato con el fin de tener un array de dimensiones iguales al tiempo de ejecución. Por tanto, los valores sin ajustar y ajustados quedarían tal que:

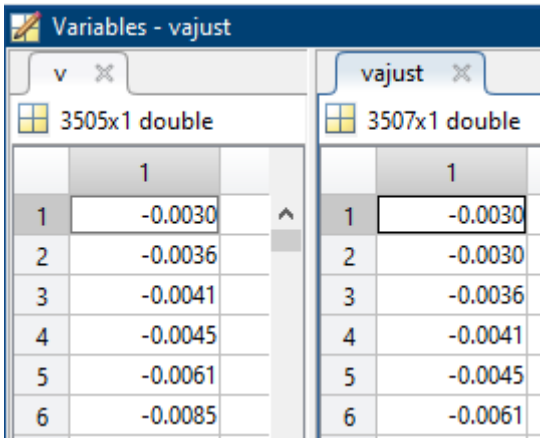


Ilustración 48. Valores sin ajustar (izquierda) y ajustados al inicio de variable (derecha).

Variables - v

v

3505x1 double

	1
3500	0.0021
3501	0.0016
3502	0.0023
3503	0.0026
3504	3.4095e-04
3505	4.5500e-04
3506	
3507	
3508	

vajust

3507x1 double

	1
3500	-7.4727e-04
3501	0.0021
3502	0.0016
3503	0.0023
3504	0.0026
3505	3.4095e-04
3506	4.5500e-04
3507	4.5500e-04
3508	

Ilustración 49. Valores sin ajustar (izquierda) y ajustados al final de la variable (derecha).

De esta manera, se puede calcular la velocidad suavizada y representar las gráficas en una misma figura obteniendo lo siguiente:

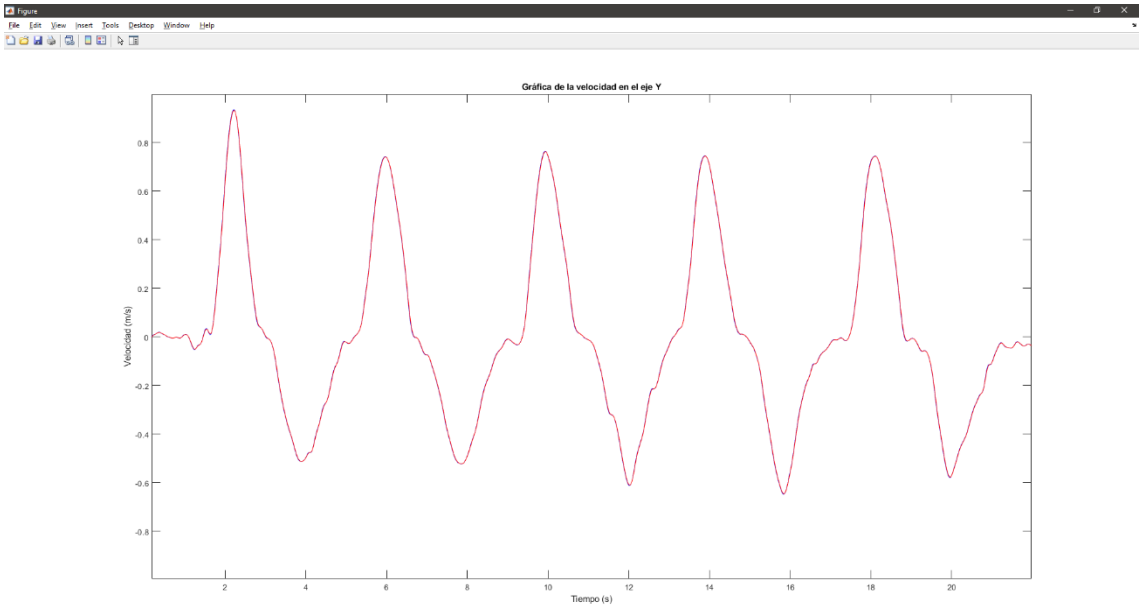


Ilustración 50. Gráfica de la velocidad de ejecución del ejercicio.

Como se puede observar en la gráfica, la velocidad de la primera repetición es mayor que el resto. Sin embargo, en repeticiones posteriores se mantiene muy estable la velocidad de subida y varía levemente la velocidad de bajada.

Para poder entender esta gráfica de manera completa, cabe destacar que tras cada subida o bajada en la ejecución del ejercicio el valor de la velocidad vuelve a cero, esto se debe a que tanto en el punto más alto como en el punto más bajo de ejecución se

realiza una parada breve. Por tanto, en una repetición, se parte de velocidad cero, se realiza la subida donde se llega al máximo, al no haber desplazamiento la velocidad vuelve a ser cero, y en la bajada sucede lo mismo, hay velocidad cuando hay movimiento.

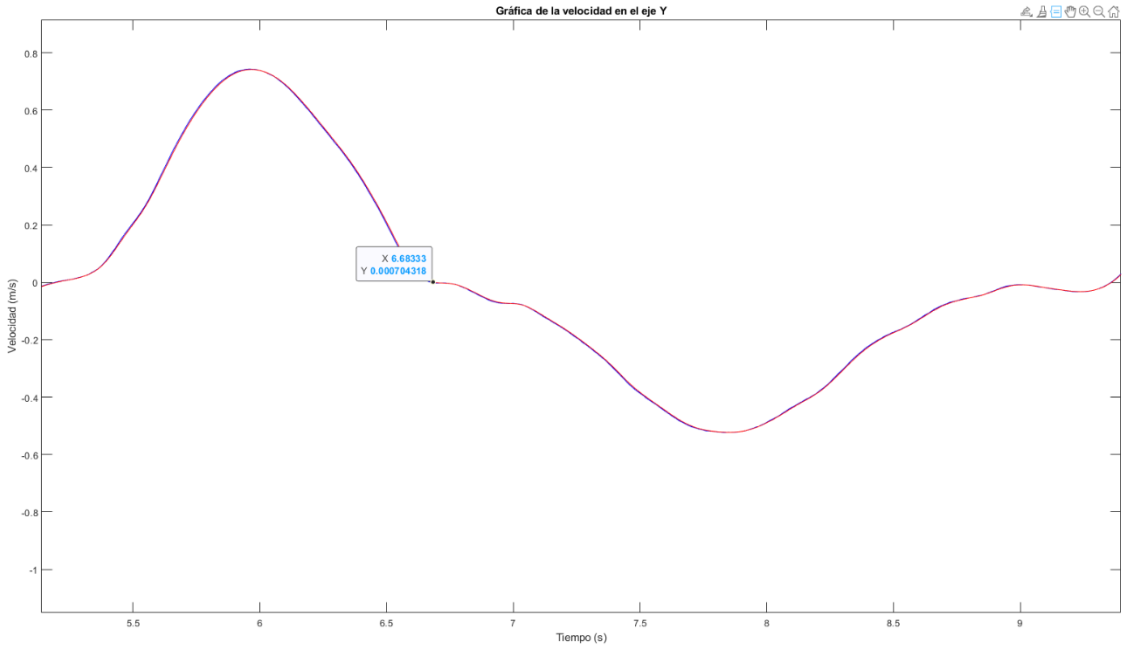
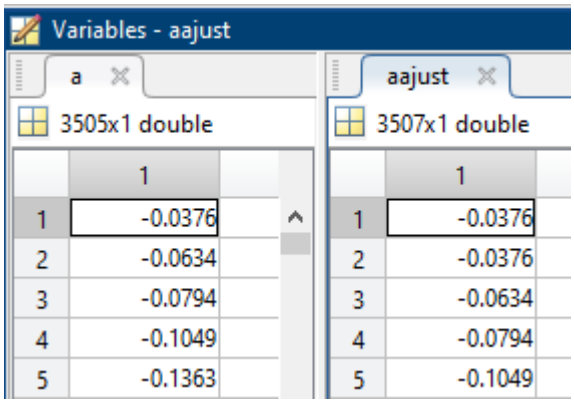


Ilustración 51. Gráfica de la velocidad de una única repetición. El punto definido indica la finalización de la subida.

En esta gráfica, observamos una única repetición (se han reducido los ejes para poder verla completa) y se aprecia lo mencionado anteriormente de la velocidad cero. Para hacerlo más gráfico se ha colocado un punto de referencia tras la subida donde la velocidad es v=0.0007m/s que es prácticamente cero.

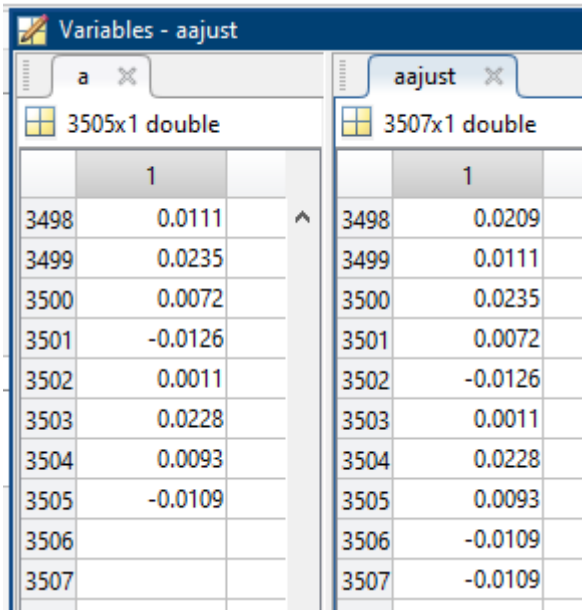
6.1.3 Aceleración

A continuación, se analizará la aceleración de ejecución. Del mismo modo que sucedía con la velocidad, en esta sección también se copian el primer y último valor para que coincida con la longitud del tiempo.



	a		aajust
	1		1
1	-0.0376	1	-0.0376
2	-0.0634	2	-0.0376
3	-0.0794	3	-0.0634
4	-0.1049	4	-0.0794
5	-0.1363	5	-0.1049

Ilustración 53. Valores sin ajustar (izquierda) y ajustados al inicio de la variable (derecha).



	a		aajust
	1		1
3498	0.0111	3498	0.0209
3499	0.0235	3499	0.0111
3500	0.0072	3500	0.0235
3501	-0.0126	3501	0.0072
3502	0.0011	3502	-0.0126
3503	0.0228	3503	0.0011
3504	0.0093	3504	0.0228
3505	-0.0109	3505	0.0093
3506		3506	-0.0109
3507		3507	-0.0109

Ilustración 52. Valores sin ajustar (izquierda) y ajustados al final de la variable (derecha).

De esta manera, se puede calcular la aceleración suavizada y representar las gráficas en una misma figura obteniendo lo siguiente:

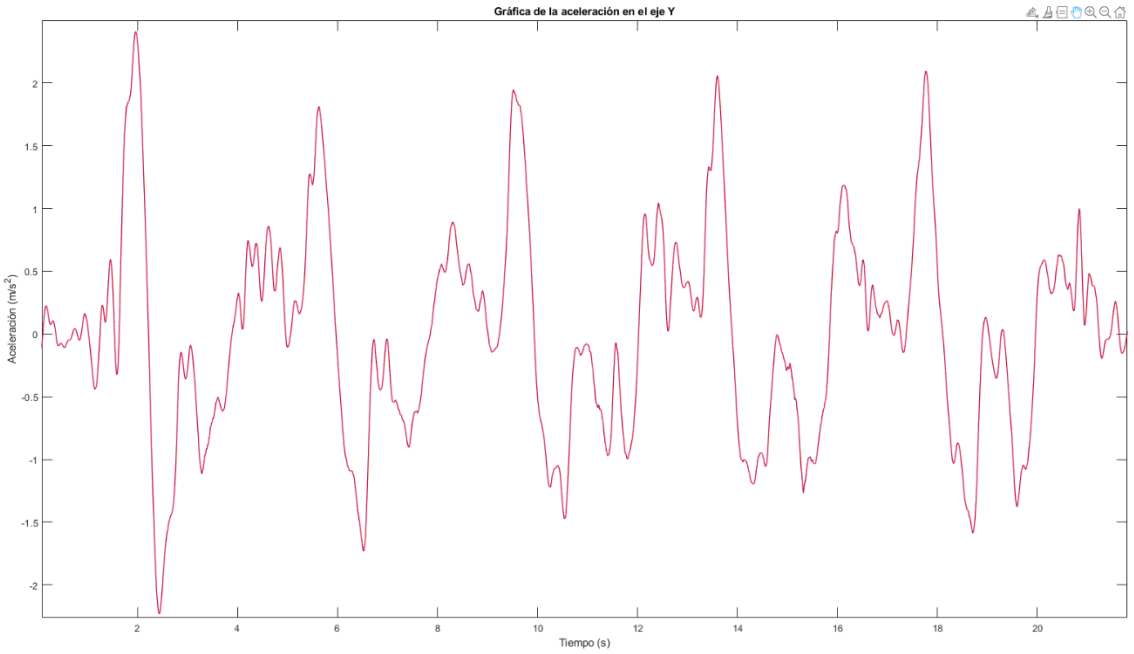


Ilustración 54. Gráfica de la aceleración.

Como la aceleración depende de la velocidad, la gráfica presenta similitud con la velocidad, siendo la primera repetición algo diferente en aceleración al resto de ellas que son más parecidas. Las irregularidades se deben a los pequeños parones que se realizan durante el ejercicio.

Si nos enfocamos únicamente en una repetición (por ejemplo la segunda), podemos diferenciar claramente la subida durante el ejercicio, una vez en el punto más alto, al no existir velocidad de movimiento, la fuerza disminuye hasta llegar a cero (alrededor del segundo 6.5) y a partir de ese momento comienza el descenso que es más irregular y lento que la subida.

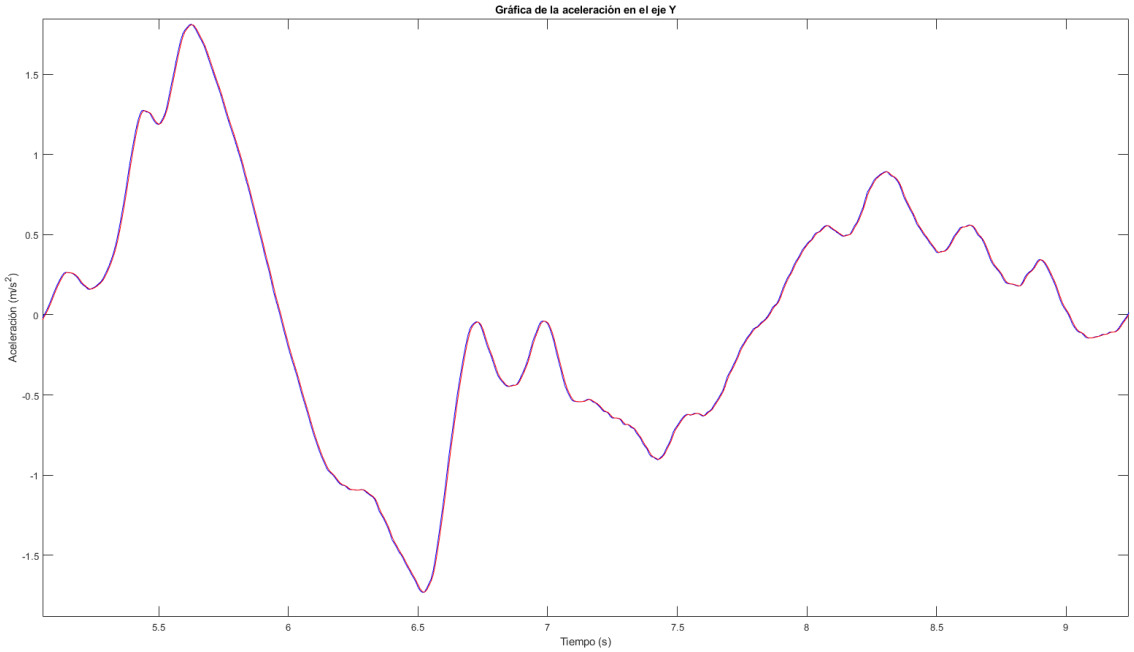


Ilustración 55. Aceleración de una única repetición.

6.1.4 Fuerza

Para la representación de la fuerza no ha sido necesario realizar ajustes, simplemente se ha multiplicado la aceleración por la masa, se ha suavizado el resultado y ambas variables se han representado en el siguiente gráfico:

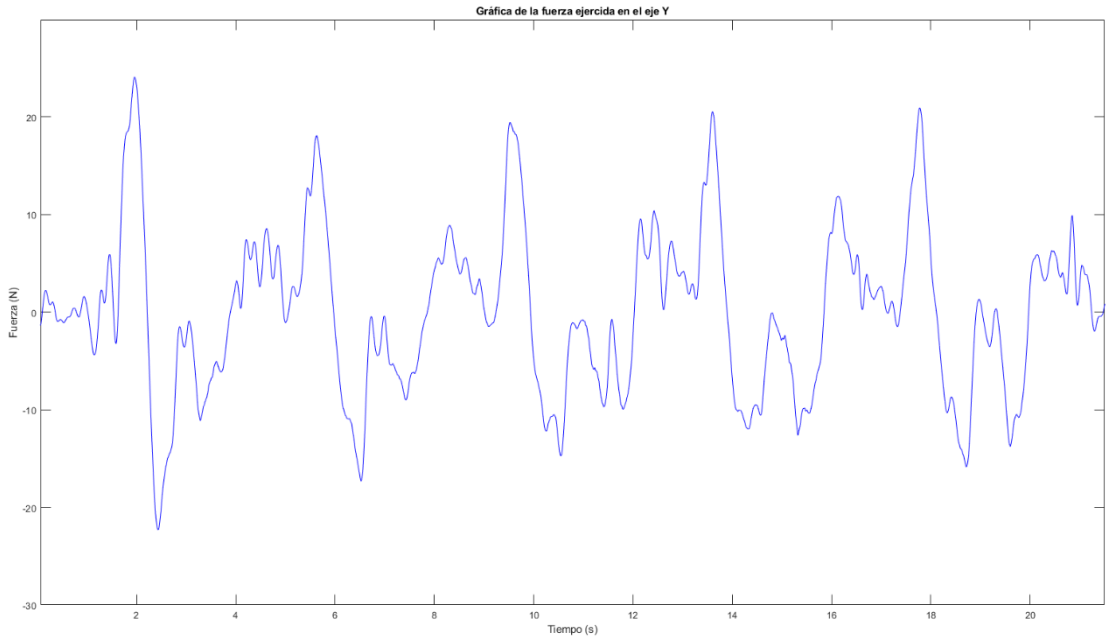


Ilustración 56. Gráfica de la fuerza ejercida.

El gráfico de fuerza es prácticamente igual al gráfico de la aceleración, la única diferencia es la magnitud del eje Y. En este caso la aceleración al ser multiplicada por la masa (10kg) los valores obtenidos serán diez veces mayores a los que se calculan en la aceleración, dando como resultado la misma gráfica escalada en diez veces.

Del mismo modo, si analizamos una única repetición, veremos que los datos obtenidos coinciden con la aceleración, pero multiplicada por un factor diez.

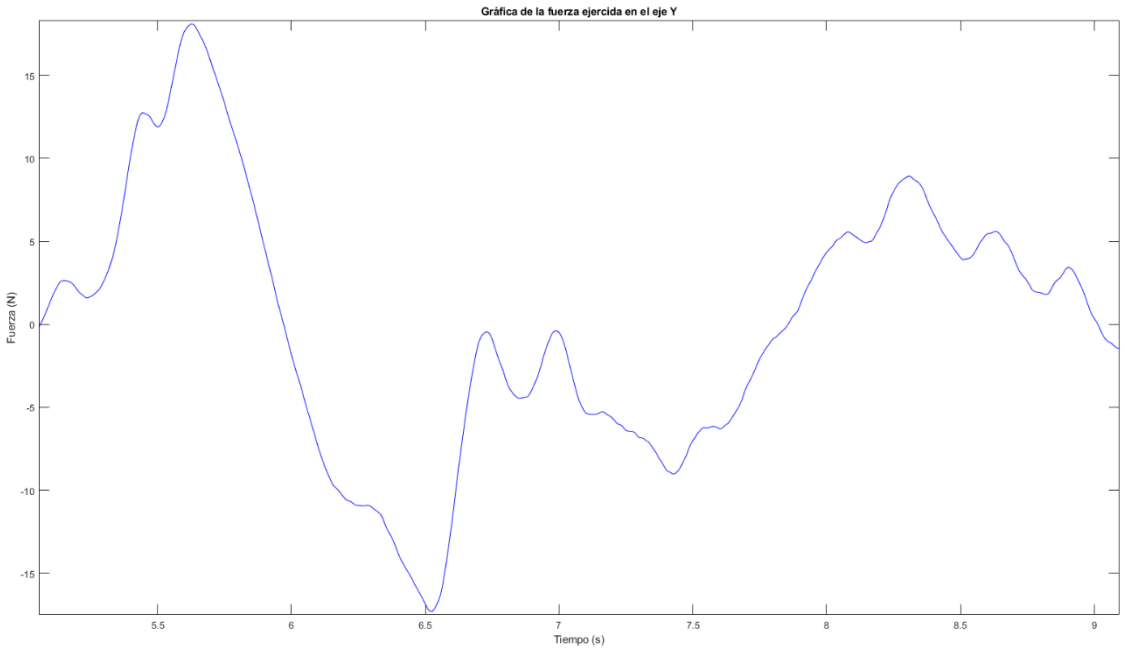


Ilustración 57. Gráfica de fuerza de una única repetición.

6.1.5 Potencia

En el caso de la potencia ocurre algo similar a la fuerza, pero esta vez en lugar de multiplicar un array por un escalar, se multiplica un array por otro array dato a dato, esto quiere decir que se multiplicará el dato en la posición uno de la fuerza con el que está en la posición uno de la velocidad y así sucesivamente. Una vez calculado esto y suavizado, se representan las gráficas en la misma figura obteniendo:

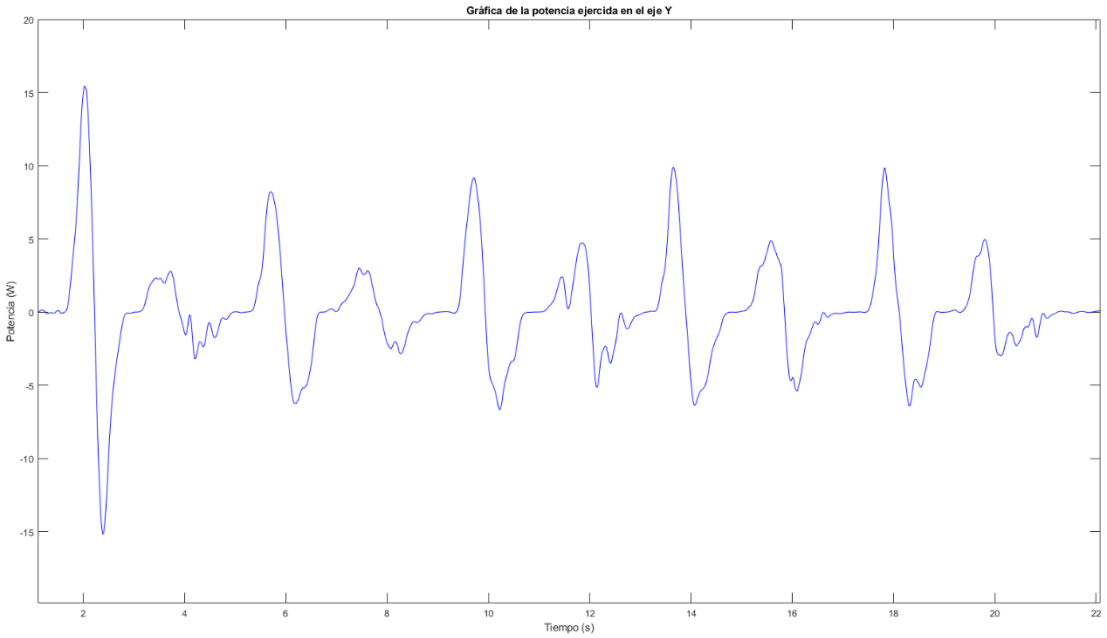


Ilustración 58. Gráfica de la potencia.

Como se aprecia en la gráfica, en la primera repetición se ejerce mucha más potencia que el resto, esto se debe a que, normalmente, cuando empiezas un ejercicio la primera repetición que se lleva a cabo suele usarse como ejemplo para las siguientes y así afianzar tanto la distancia recorrida como la velocidad a la que se quiere realizar el ejercicio. Esto se traduce en una primera repetición muy diferente de las que vienen a continuación, que como se aprecia son bastante similares.

Si se analiza una repetición, se puede observar que está perfectamente diferenciada la subida de la bajada, teniendo incluso un periodo corto de tiempo donde la potencia es cero indicando que se ha ejecutado la subida al completo. La potencia de bajada es menor ya que la velocidad a la que se realiza el ejercicio disminuye en las bajadas. Esto se debe a que, de normal, los ejercicios donde hay subida y bajada se realizan a una velocidad alta en la primera acción (de manera explosiva) y se baja lentamente controlando el peso.

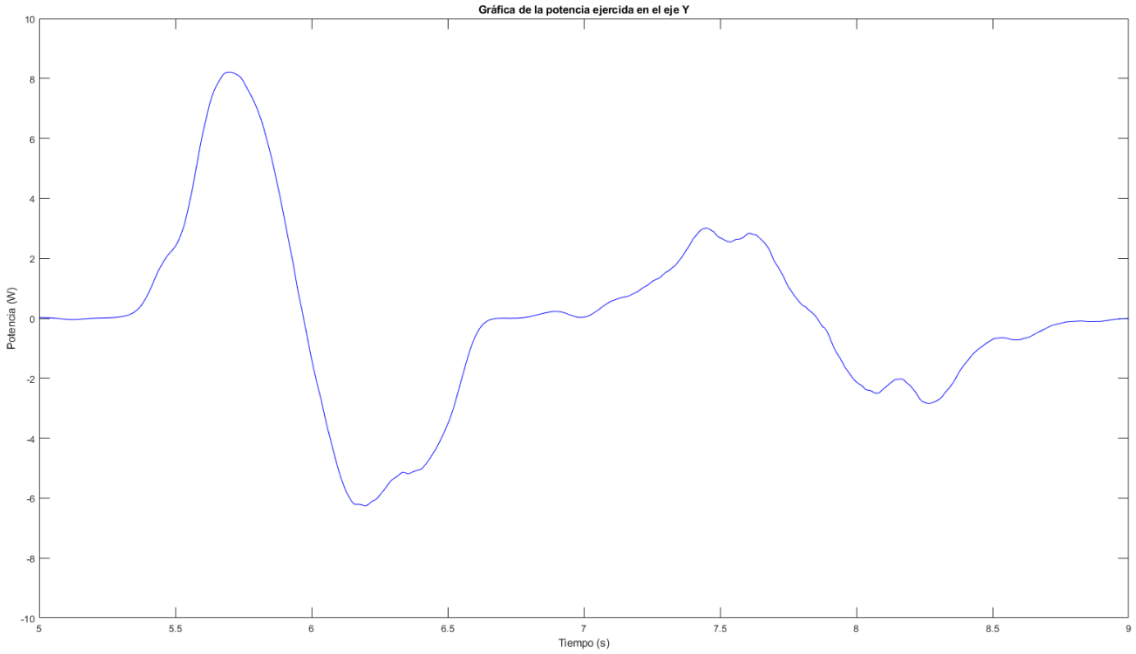


Ilustración 59. Gráfica de la potencia de una única repetición.

6.2 Movimiento del esqueleto

En este apartado se aportarán algunas imágenes de las estructuras mencionadas en la explicación del código y de los resultados de la programación para mostrar el movimiento del esqueleto por pantalla. Las imágenes del esqueleto corresponderán a diferentes fotogramas y mostrarán la posición de todos los marcadores utilizados en un espacio 3D.

Con respecto a las estructuras, cabe mencionar la llamada DataM, que contiene la información que se utilizará para representar los marcadores. Esa estructura está compuesta de los siguientes datos:

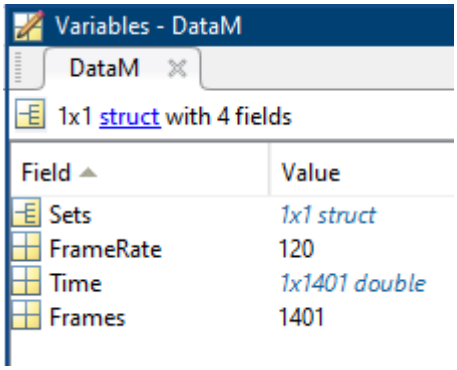


Ilustración 60. Variable DataM con los datos introducidos.

Como se aprecia contiene una estructura llamada Sets, que estará compuesta de las diferentes entidades que se quieran representar, en este caso solo aparece la entidad esqueleto que será la que representaremos.

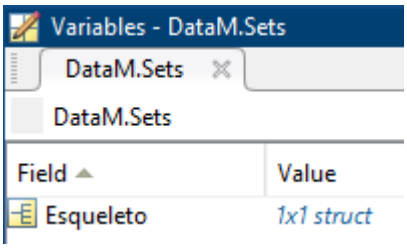


Ilustración 61. Variable *Sets* dentro de DataM.

En la estructura de las entidades encontraremos la información relacionada con el número de marcadores y otra estructura que tendrá la posición de los ejes x, y, z de cada marcador.

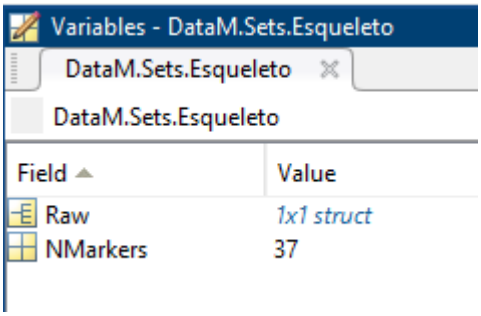


Ilustración 62. Variables dentro de la estructura *Esqueleto*.

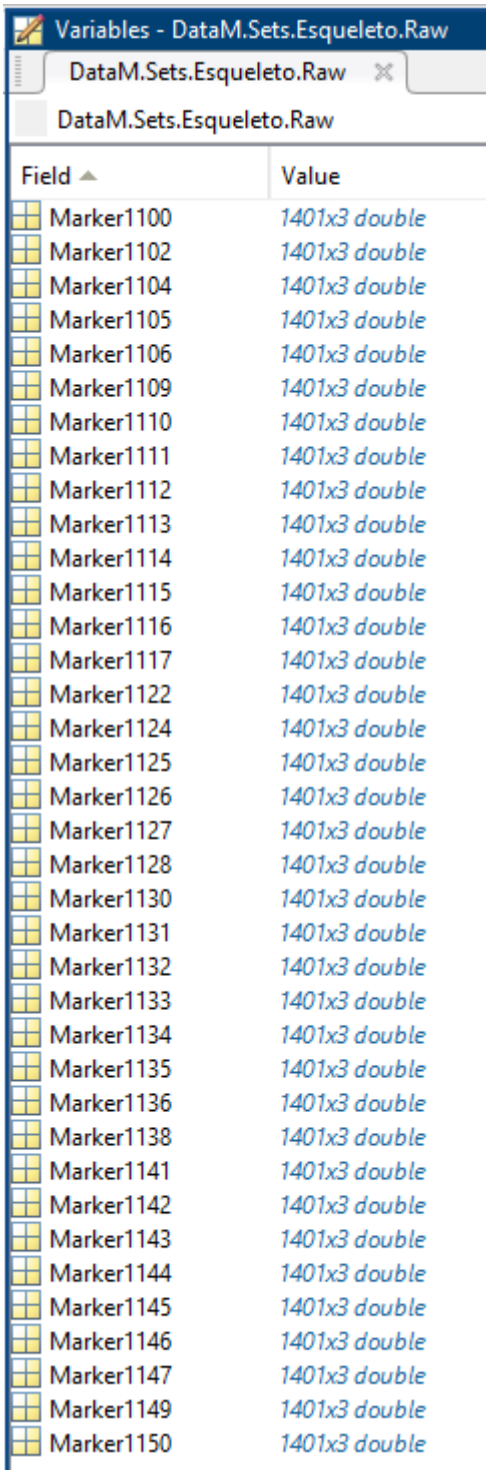


Ilustración 63. Marcadores del esqueleto (39) con los datos de las coordenadas x, y, z.

A continuación, se mostrarán las imágenes de distintos fotogramas del esqueleto en movimiento donde se aprecian todos los marcadores.

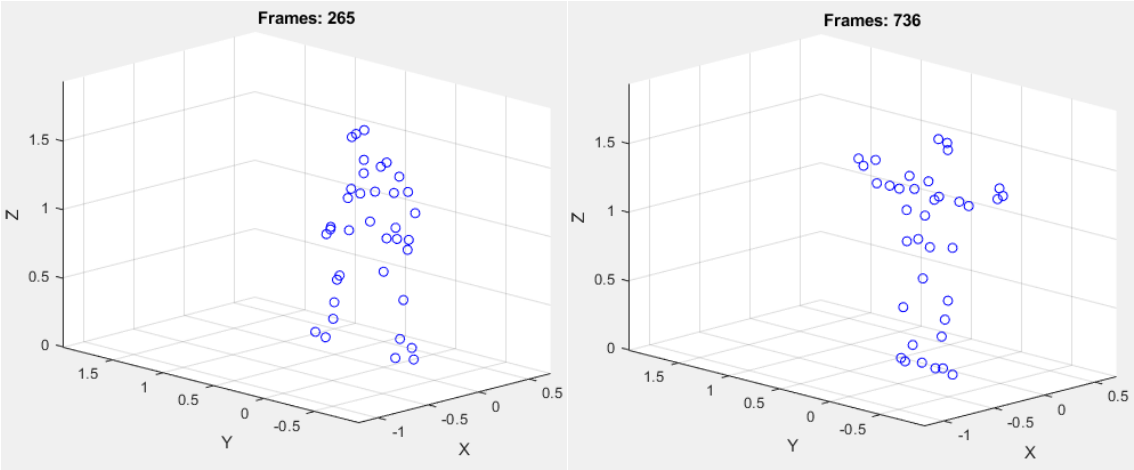


Ilustración 64. Representación de los fotogramas mediante una nube de puntos en un espacio 3D.

Asimismo, cuando se cierra la ventana de representación 3D, se nos muestra el movimiento, pero visto desde arriba. Esto podría ser especialmente útil para ejercicios donde el movimiento no sea en el eje vertical. Para que se pueda apreciar la diferencia entre los puntos de vista, se han tomado capturas de los mismos fotogramas representados en 3D, pero con la nueva representación.

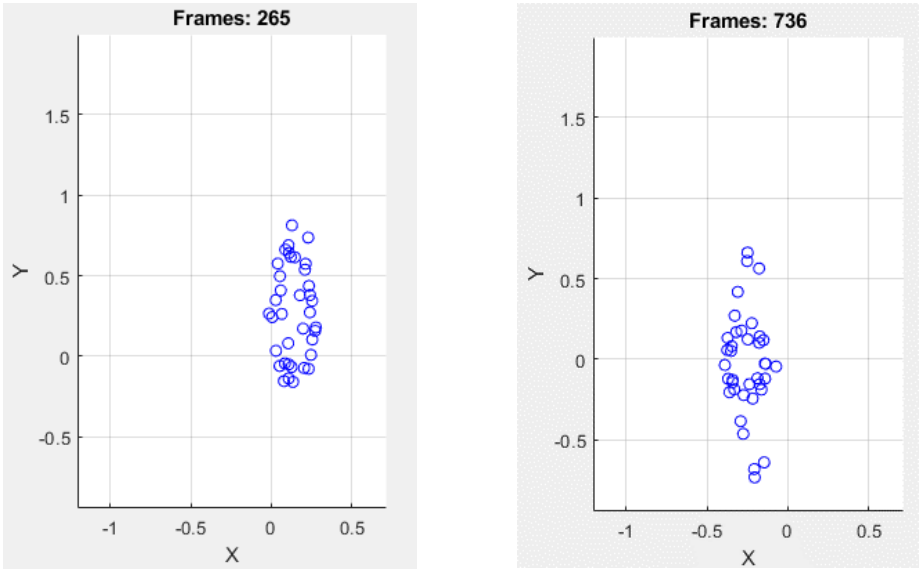


Ilustración 65. Nube de puntos vista desde arriba.

Capítulo 7. Conclusiones y proyección de futuro

7.1 Conclusiones

Tras el estudio realizado sobre el uso de los sistemas MoCap, la utilización de ellos y la programación de dos ficheros para la representación de los datos obtenidos, se ha llegado a siguientes conclusiones:

1. La realización del proyecto ha permitido adquirir conocimientos sobre las tecnologías MoCap y su uso en la actualidad.
2. El software Motive de Optitrack es una herramienta de detección y grabación muy precisa y la exportación de datos es sencilla.
3. El programa de obtención de parámetros funciona correctamente con diferentes archivos .csv y obtiene los valores planteados.
4. El programa de representación 3D funciona de manera adecuada para representar al menos dos entidades.
5. Los resultados generales obtenidos son bastante precisos y el balance general es positivo, ya que se han cumplido todos los objetivos planteados.

7.2 Desarrollo futuro del proyecto

Este proyecto sienta las bases para poder validar las aplicaciones de teléfono móvil que se están desarrollando para obtener los valores de velocidad, aceleración, fuerza y potencia durante la realización de los ejercicios a través del vídeo de la cámara.

Capítulo 8. Bibliografía

8.1 Bibliografía

[1] FisioCampus. (2017) "¿Contracción concéntrica o excéntrica? ¡Esa es la cuestión!". Disponible: https://www.fisiocampus.com/articulos/contraccion-concentrica-o-excentrica-esa-es-la-cuestion. [En línea].

[2] McHugh, M. (2003). "Recent advances in the understanding of the repeated bout effect". *Scandinavian Journal of Medicine Science in Sports*.

[3] Hernández, G. (2015) "Efectos visuales: Desarrollo y evolución a lo largo de la historia del cine", Universidad Politécnica de Valencia. Trabajo fin de carrera.

[4] Wikipedia. (2017). "Development of The Last of Us". Disponible: https://en.wikipedia.org/wiki/Development_of_The_Last_of_Us. [En línea].

[5] Karol Smolak. (2020). WL Analysis (1.0.27) [Aplicación móvil]. App Store. Disponible: https://wlanalysis.com/. [En línea].

[6] Core Advantage Pty Ltd. (2022). Metric VBT (0.5.7) [Aplicación móvil]. App Store. Disponible: https://www.metric.coach/. [En línea].

[7] Teseo. (2020). "¿Qué es y cómo funciona la captura de movimiento?" Disponible: https://teseo.es/noticias/que-es-y-como-funciona-la-captura-de-movimiento/. [En línea].

[8] Saúl Menéndez; Jonathan Rodríguez. "Protocolo de trabajo y banco de pruebas para el uso del equipo de captura de movimiento MoCap - Optitrack", Universidad de La Laguna, Tenerife, 2015. Trabajo fin de carrera.

[9] Wikipedia. (2008). "Triangulation (computer visión)". Disponible: https://en.wikipedia.org/wiki/Triangulation_(computer_vision). [En línea].

[10] Bradski, G. y Kaehler, A. (2008). "Learning OpenCV". *O'Reilly Media, Inc*.

[11] Optitrack. (2022). Motive (2.2.0) [Software ordenador]. Disponible: https://optitrack.com/software/motive/. [En línea].

[12] Microsoft. (2022). Excel (Excel 2019) [Software ordenador]. Disponible: https://www.microsoft.com/es-es/microsoft-365/excel. [En línea].

[13] MathWorks. (2022). Matlab (R2022a) [Software ordenador] Disponible: https://es.mathworks.com/products/matlab.html. [En línea].

8.2 Bibliografía de imágenes

Ilustración 1. Fuente: https://www.fisiocampus.com/articulos/contraccion-concentrica-o-excentrica-esa-es-la-cuestion

Ilustración 2. Fuente: https://indianexpress.com/article/entertainment/hollywood/lord-of-the-rings-actor-andy-serkis-has-active-sex-life-4732983/

Ilustración 3. Fuente: https://giphy.com/gifs/the-last-of-us-WR1E1OC6zxtf2

Ilustración 4. Fuente: https://apps.apple.com/es/app/wl-analysis/id1541855037

Ilustración 5. Fuente: https://apps.apple.com/au/app/metric-vbt/id1595510857

Ilustración 6. Fuentes: https://teseo.es/noticias/que-es-y-como-funciona-la-captura-de-movimiento/ y https://www.researchgate.net/publication/332350629_Desarrollo_de_dispositivo_electronico_de_bajo_costo_para_evaluar_localizacion_de_fuentes_sonoras/figures?lo=1

Ilustración 7. Fuente: https://optitrack.com/cameras/primex-41/

Ilustración 8. Fuente: https://optitrack.com/accessories/calibration-tools/

Ilustración 9. Fuente: https://optitrack.com/accessories/calibration-tools/

Ilustración 10. Fuente: https://www.redalyc.org/journal/5075/507551264001/html/#gf6

Ilustración 11. Fuente: https://hmong.es/wiki/Epipolar_geometry

Ilustración 12. Fuente: https://optitrack.com/accessories/markers/#mcm-19.0-m4-10

Ilustración 13. Fuente: https://amrproducciones.blogspot.com/2011/07/captura-de-movimiento-3d.html

Ilustración 14. Fuente: https://optitrack.com/applications/movement-sciences/

Ilustración 15. Fuente: https://optitrack.com/software/motive/

Ilustración 16. Fuente: https://www.microsoft.com/es-es/microsoft-365/excel

Ilustración 18. Fuente: https://es.mathworks.com/products/matlab.html