

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA BIOMÉDICA
TRABAJO FIN DE GRADO**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
CONTROL BASADO EN EL SISTEMA OPERATIVO
ROS PARA LA PLATAFORMA ROBÓTICA DE
REHABILITACIÓN DE LA MARCHA SWALKER**

**CARLOS BETANCORT RODRÍGUEZ
2021**

GRADO EN INGENIERÍA BIOMÉDICA

TRABAJO FIN DE GRADO

Título: Diseño e implementación de un sistema de control basado en el sistema operativo ROS para la plataforma robótica de rehabilitación de la marcha swalker.

Autor: D. Carlos Betancort Rodríguez.

Tutor: D. Eduardo Rocón de Lima.

Cotutor: D. Álvaro Gutiérrez Martín.

Departamento: Tecnología Fotónica y Bioingeniería.

MIEMBROS DEL TRIBUNAL

Presidente: D.

Vocal: D.

Secretario: D.

Suplente: D.

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de:
.....

Madrid, a de de 20...

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA BIOMÉDICA
TRABAJO FIN DE GRADO**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
CONTROL BASADO EN EL SISTEMA OPERATIVO
ROS PARA LA PLATAFORMA ROBÓTICA DE
REHABILITACIÓN DE LA MARCHA SWALKER**

**CARLOS BETANCORT RODRÍGUEZ
2021**

AGRADECIMIENTOS

Quiero expresar mi gratitud a todas las personas que han hecho posible no solo la realización de este Trabajo de Fin de Grado, sino que han contribuido a mi experiencia durante estos últimos cuatro años, ayudándome a progresar tanto académica como personalmente.

En primer lugar, agradecer a Álvaro Gutiérrez y Eduardo Rocón por su inestimable ayuda y consejo durante este último año. Gracias a ambos por vuestra supervisión y orientación a lo largo de este proyecto y por permitirme compartir esta experiencia de trabajo con vosotros. Gracias por acogerme en vuestro espacio de trabajo como uno más y ayudarme a progresar a través de vuestra profesionalidad. Gracias por los almuerzos conjuntos en la cafetería y por las conversaciones para arreglar el mundo en el taller. Por todo esto y mucho más, muchísimas gracias.

En segundo lugar, dar las gracias a los compañeros de gNEC y del CAR por formar parte de esta pequeña muestra de lo que significa trabajar en un entorno profesional como es vuestro grupo. En concreto, agradecer a Pablo y a Julio, por ser no solo compañeros, sino referentes personales y profesionales. La ayuda que me habéis prestado, no solo durante este TFG, sino de cara a mi futuro profesional y académico no se puede pagar. Gracias por las conversaciones casuales esperando el bus y las reflexiones sobre el futuro, aunque alguna incluyera correr sobre una cinta transportadora. Por todo esto y mucho más, muchísimas gracias.

En tercer lugar, agradecer a todos los compañeros de carrera que he conocido a lo largo de estos años. Algunos de vosotros habéis sido bondades efímeras en mi vida, y otros habéis constituido los cimientos de mi desarrollo personal, a lo largo de este intenso viaje. Aquellos con los que he compartido risas y alegrías, aquellos con los que he llorado hasta no poder más, aquellos con los que no mantengo la mejor de las relaciones, y aquellos que una vez estuvieron, todos formáis parte de mi vida y os estaré siempre agradecidos. Por todo esto y mucho más, muchísimas gracias.

Por último, agradecer a mi familia el apoyo incondicional que siempre me han prestado, desde el minuto uno de la carrera hasta el momento final de este proyecto. Gracias a mi madre Carmen, a mi padre Rafael y a mi hermano Pablo, habéis sido mi apoyo y habéis creído en mí, tanto en mis horas más bajas como en mis mejores éxitos. Gracias también a mis tíos y tías, que no han hecho más que apoyarme y ayudarme durante estos años, y a mis primos, quienes han aportado su granito de arena, cada uno a su manera. Por todo esto y mucho más, muchísimas gracias.

Todas y cada una de las personas mencionadas, y aquellas que no, habéis dejado una huella en mí que define quien soy el día de hoy, y por ello no puedo más que daros mi más sincera gratitud.

RESUMEN

Con las crecientes proporciones de población con edad avanzada, debido al aumento de la longevidad en las últimas décadas, el asegurar la calidad de vida de nuestros mayores se ha vuelto un reto cada vez más urgente. Entre los problemas encontrados, la fractura de cadera es una patología que afecta a una proporción creciente de los pacientes, y que tiene graves efectos en su calidad de vida, tanto clínicos como económicos. Esta clase de fracturas han demostrado beneficiarse enormemente de tratamientos de rehabilitación física inmediatamente posteriores a la fractura, pero debido al seguimiento necesario para obtener un resultado tangible, esta clase de procedimientos no se llevan a cabo, principalmente por el alto coste en personal y organización asociado.

El proyecto SWalker del Centro de Automática y Robótica (CAR) del Consejo Superior de Investigaciones Científicas (CSIC), fue diseñado como una alternativa eficiente y capaz de ofrecer un tratamiento de rehabilitación basado en la marcha completo, mientras se recoge la información necesaria para el seguimiento por parte de los profesionales sanitarios.

El objetivo de este Trabajo de Fin de Grado es el diseño e implementación de un sistema de control, utilizando la arquitectura ROS, para el SWalker, capaz de gestionar la información de la que dispone el dispositivo y asegurar un correcto funcionamiento, respetando los principios de modularidad y futuras expansiones de funcionalidad previstas para la plataforma. El sistema debe ser capaz de interpretar la información de movimiento articular generado por el paciente y generar una salida de velocidad para el sistema, de forma que se adecue a los parámetros de marcha estipulados para la terapia. Así mismo, debe proveer un sistema de coordinación para los distintos nodos que componen el circuito completo de control a nivel de software, permitiendo el intercambio de información y el establecimiento de protocolos secuenciales internos que involucren la coordinación simultánea de varios nodos.

PALABRAS CLAVE

- Rehabilitación asistida por robots
- Robotic Operating System (ROS)
- Fractura de cadera
- Sistema de control
- Rehabilitación de marcha
- Monitorización

SUMMARY

Due to the increasing demographic tendencies in the last decades, the proportion of the population above 60 years is steadily increasing. This process is deepening the need of the healthcare systems to provide viable solutions to the diseases associated with old age. Among these, hip fracture is one of the more prominent problems and it has an important effect on the quality of life of the patients, as well as a great associated economic cost. This type of fracture greatly benefits from an early rehabilitation treatment after surgery. Unfortunately, this kind of treatment is usually not viable due to the limitations in time and human resources of the healthcare systems.

The SWalker project, presented by the “Centro de Automática y Robótica” (CAR) of the “Consejo Superior de Investigaciones Científicas” (CSIC), was proposed in order to increase the efficiency and availability of this kind of treatment. This platform provides an infrastructure that facilitates the performing of gait based rehabilitation training for hip fracture patients.

The objective of this Final Year project is to design, develop and implement a control system for the SWalker platform, using the ROS architecture. This system must be able to manage the information needed in order to perform the therapies, and provide assistance for the handling of the session. On top of that, this system must be developed within the modularity and future expansion projected for the platform. The functions of this system will primarily to generate a speed profile in accordance with the therapy session, and monitor the patient's ability to follow this profile. Additionally, this system must be able to coordinate and keep record of the system status, in order to sustain the various modes of function of the system.

ÍNDICE DEL CONTENIDO

AGRADECIMIENTOS	4
RESUMEN	5
PALABRAS CLAVE	5
SUMMARY	6
ÍNDICE DEL CONTENIDO	7
INTRODUCCIÓN	8
1.1 Problema clínico: Fractura de cadera	8
1.1.1 Efectos derivados	9
1.1.2 Tratamientos rehabilitadores. Rehabilitación de Marcha	10
1.2 Plataforma Robótica SWalker	14
1.2.1 Estructura mecánica	15
1.2.2 Arquitectura de control	17
1.3 Motivación y objetivos	17
SISTEMA DE CONTROL	19
2.1 Infraestructura de control	19
2.1.1 Redes de Comunicación	19
2.1.2 Sensores y Actuadores	22
2.1.3 Robotic Operating System (ROS)	23
2.2 Circuito de Control	25
2.2.1 Nivel de entrada	26
2.2.2 Nivel de procesamiento	29
2.2.3 Nivel de salida	31
RESULTADOS DE DESARROLLO	33
3.1 Sistema de estados interno	33
3.2 Sistema de control de velocidad	38
CONCLUSIONES Y TRABAJOS FUTUROS	42
BIBLIOGRAFÍA	43
ANEXOS	46
A.1 Impacto Socio-económico	46
A.2 Impacto ético y profesional	46
A.3 Presupuesto	47
A.4 Fichas de datos	48

Capítulo 1

INTRODUCCIÓN

Este Trabajo de Fin de Grado ha sido realizado en colaboración con el Centro de Automática y Robótica (CAR) del Consejo Superior de Investigaciones Científicas (CSIC), sobre el proyecto SWalker [1][2]. Este proyecto presenta una plataforma de rehabilitación de tipo andador inteligente, especialmente diseñado para pacientes con fractura de cadera de edad avanzada. Como desarrollo de este trabajo, se detallan los procesos de diseño, desarrollo e implementación de la arquitectura de control y procesado de información sobre la arquitectura ROS.

En este capítulo se realiza una aproximación al problema clínico que supone la fractura de cadera, sus proyecciones futuras, el estado del arte de los procesos de rehabilitación y la motivación para el trabajo.

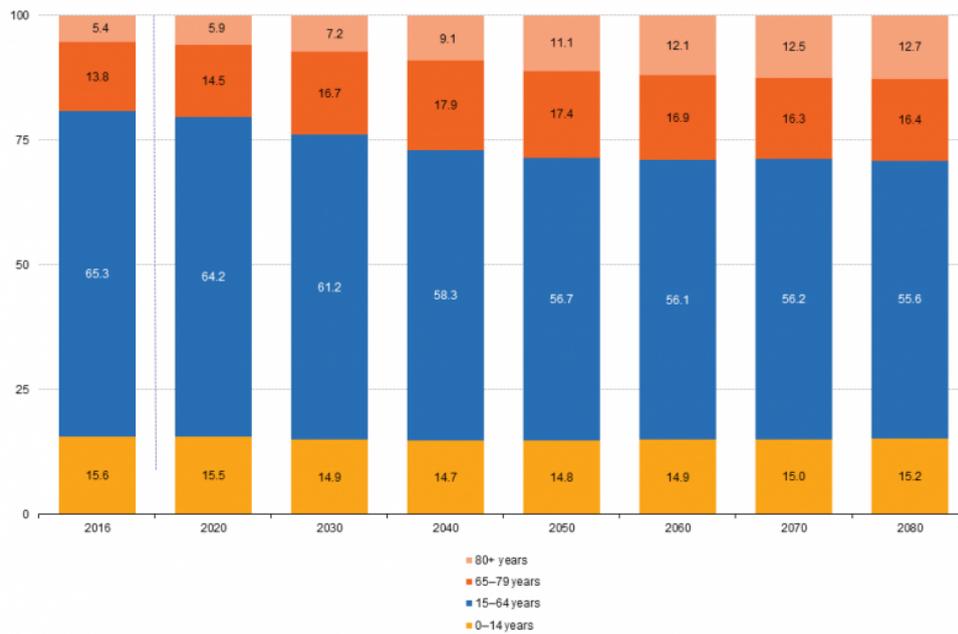
1.1 Problema clínico: Fractura de cadera

A lo largo de las últimas décadas, las tendencias demográficas globales han demostrado un creciente cambio en la estructura por edades de la población, debido principalmente al descenso de la natalidad y el aumento de la longevidad. Este proceso se conoce como “envejecimiento poblacional” y es consecuencia directa de las transiciones demográficas asociadas a las evoluciones sociales y tecnológicas propias del desarrollo de las sociedades [3].

En la Unión Europea, se prevé que el porcentaje de población que supera los 80 años se duplique en el año 2050 con respecto a las métricas del año 2016 (de un 5.4% a un 11.4%) [4], elevando el porcentaje de población de más de 65 años por encima del 25% (ver Figura 1).

Este aumento de población y esperanza de vida lleva asociado un crecimiento en la incidencia de patologías asociadas a la edad avanzada, especialmente las ligadas a fragilidad ósea y reducción de la actividad física. En concreto, la fractura de cadera es un problema clínico comúnmente derivado de accidentes domésticos.

Las fracturas de cadera se asocian fuertemente con el debilitamiento de los huesos debido a patologías como la osteoporosis. Este efecto, combinado con el riesgo de sufrir accidentes como caídas, debido a la reducción de visión, coordinación y fuerza muscular asociada a la edad avanzada, constituye una de las principales causas de la aparición de fracturas. Debido a esto, diversos estudios han determinado que el riesgo residual ajustado por mortalidad de sufrir una fractura una vez superados los 60 años, asciende al 44% en mujeres y 25% en hombres, con una tendencia ascendente [5][6].



Note: 2016: estimate, provisional. 2020-80: projections (EUROPOP2015).
Source: Eurostat (online data codes: demo_pjangroup and proj_15ndbims)

Figura 1: Estimación de reparto poblacional. Fuente: [7]

1.1.1 Efectos derivados

El sufrir una fractura de cadera tiene importantes efectos en la vida del paciente y sus círculos, tanto de forma inmediata como a largo plazo. En la mayoría de los casos, las fracturas de cadera requieren de cirugía y estancia hospitalaria prolongada, y comúnmente no se alcanza una recuperación completa.

Desde el punto de vista económico, el tratamiento de una fractura de esta clase dentro de los países de la Unión Europea oscila entre 2.000 y 25.000 euros por paciente dependiendo del país, únicamente para el primer año [8], con una media de 13.816 euros en el año 2012. Adicionalmente, para los años posteriores, se estima que existe un gasto medio de 2.400 euros adicionales debido a los efectos a largo plazo que tiene la fractura en el paciente [9]. Estos efectos, abarcan modificaciones en el estilo de vida, como pueden ser el internamiento en centros geriátricos, costes médicos asociados a terapias y revisiones, o incluso pérdida de productividad para los casos en población activa.

En lo relativo a la realización de vida diaria posterior al tratamiento inicial, una fractura de cadera implica un alto riesgo de desarrollar discapacidades y/o dependencias crónicas en un periodo de entre uno y dos años. Como resultado de un compendio de estudios en los efectos de las fracturas de cadera [10], se destacan los siguientes:

- Reducción de movilidad: tras dos años, los pacientes exhiben importantes limitaciones motoras, especialmente en la parte inferior del cuerpo, con especial dificultad para la marcha.
- Pérdida de independencia en las actividades diarias (Activities of daily living - ADLs): un 24% de los pacientes exhiben pérdidas en la independencia en cuanto a realización de actividades diarias, pasado un año de la fractura. Adicionalmente, estos pacientes son significativamente más propensos a la dependencia funcional y a continuar con dificultades en las actividades diarias pasados dos años de la fractura.
- Reducción en la participación en vida social y comunitaria: más de la mitad (53%) de los pacientes reportaron necesitar de asistencia externa para realizar actividades que requieren desplazarse más allá de unos pocos metros.
- Deterioro en el estado cognitivo y de salud general: pasados dos años de la fractura, la mitad de los pacientes reportaron un descenso en su estado cognitivo y de salud, especialmente en el caso de las mujeres.
- Traslado a instituciones geriátricas: Entre 6 y 12 meses tras la fractura, entre un 10 y 20% de los pacientes previamente independientes son institucionalizados como consecuencia de la misma. Esta proporción demostró mantenerse estable durante los siguientes 10 años.

Como resultado de los estudios anteriores, se observó que la mayoría de los pacientes que experimentaron una recuperación hasta niveles similares a antes de la fractura, manifestaban un rápido progreso en los primeros meses. Específicamente en el apartado de movilidad y marcha, entre un 40 y un 60% de los pacientes que alcanzaron niveles de movilidad pre-fractura, realizaron esta recuperación en un periodo de 6 meses post-fractura.

Los ya mencionados efectos adversos, especialmente la pérdida de movilidad y fuerza en el tren inferior, traen como consecuencia que muchos pacientes no consigan una recuperación completa y desarrollen una dependencia funcional, o incluso se vean obligados a abandonar definitivamente sus residencias y su independencia. Adicionalmente, esta debilidad, junto con los efectos físicos, cognitivos y psicológicos derivados, suponen un riesgo adicional debido al aumento de la probabilidad de sufrir una caída o accidente similar.

1.1.2 Tratamientos rehabilitadores. Rehabilitación de Marcha

Diversos estudios han probado la eficacia de los ejercicios de rehabilitación fisioterapéutica con objetivo de mejorar la fuerza y el equilibrio en pacientes de edad avanzada, llegando incluso a convertirse en recomendaciones estandarizadas para personas adultas por encima de 65 años [11][12][13]. En concreto, los ejercicios concentrados en el entrenamiento de marcha o “gait training”, proporcionan mejoras sustanciales en la fuerza del tren inferior, equilibrio estático y dinámico, y coordinación y calidad de la marcha [14]. Esta clase de ejercicios presentan una práctica completa, dinámica y con efectos positivos en el estado emocional y psicológico del paciente, quien puede observar en tiempo real su progreso y recobrar un sentido de que puede realizar una actividad tan fundamental como lo es la marcha.

En la actualidad, sin embargo, no existe un consenso estandarizado para la realización de esta clase de ejercicios rehabilitadores, ni un conjunto de métodos de evaluación consistentes a lo largo de

los estudios realizados. A pesar de esto, un factor consistente en cuanto a la aplicación de estas terapias es la efectividad de ejercicios repetitivos con alta frecuencia, intensidad y número de repeticiones, de forma continuada y frecuente en el tiempo [15][16]. El principal problema que surge al aplicar esta clase de tratamiento es la dificultad organizativa y la limitación de recursos en lo que a profesionales sanitarios se refiere. Esta clase de ejercicios, especialmente los relacionados con la marcha, requieren de atención completa por parte del fisioterapeuta o profesional encargado de la sesión, y suponen un esfuerzo físico importante para estos profesionales. Esto se debe principalmente a la falta de ergonomía y asistencia externa disponible, delegando no solo la supervisión, sino también la manipulación física del paciente sobre el profesional sanitario. Son estas complicaciones las que han limitado considerablemente la cantidad y calidad de los estudios en la literatura, ya que no es posible el tratamiento constante e intensivo del paciente una vez este abandona el centro hospitalario.

Probada la eficacia de estos tratamientos, se propone una aproximación nueva a estas terapias, con el fin de reducir la carga física de los profesionales, mejorar la recolección de datos durante las sesiones y proveer de una infraestructura asistencial tanto para paciente como para el profesional sanitarios. Esta aproximación consiste en el uso de equipos robotizados, en las denominadas terapias asistidas por robots o “robot-based therapy”. Durante los últimos años, el uso de este tipo de plataformas para la rehabilitación motora ha aumentado considerablemente, ofreciendo una opción eficiente en cuanto a recursos utilizados, con buenos resultados y aceptación por parte de los pacientes [17].

Dentro de los dispositivos disponibles en lo que a rehabilitación de marcha se refiere, destacan los exoesqueletos y andadores inteligentes. El objetivo de estas plataformas es asistir al paciente mientras este realiza una marcha controlada, proporcionando funciones de monitorización, soporte y control de dirección, así como ofreciendo protección ante una posible caída. Estos dispositivos están compuestos generalmente por una serie de subsistemas que permiten ofrecer estas funcionalidades:

- Soporte parcial/total de peso: estos dispositivos cuentan con mecanismos basados en arneses u otros soportes para reducir la cantidad de peso que soporta el paciente sobre sus piernas, con el fin de facilitar la realización de los movimientos.
- Órtesis mecanizada: conseguida mediante el uso de módulos electromecánicos. Estos dispositivos reproducen los movimientos asociados a la marcha, ya sea de forma activa ejerciendo fuerza sobre el paciente; pasiva, acompañando el movimiento como guía y arnés de seguridad; o en algunos casos, contraria, ofreciendo una resistencia al movimiento.
- Control de dirección/movimiento: algunos dispositivos cuentan con sistemas que permiten regular el desplazamiento del paciente, ya sea a través de la limitación movimiento, con la inclusión de cintas transportadoras o separando al paciente por completo del suelo, o dirigiendo su movimiento sobre el espacio, controlando su dirección a través de los mecanismos de órtesis o trenes de soporte adicionales.
- Monitorización de trayectoria y movimiento: mediante la incorporación de una variedad de sensores dentro de los componentes de órtesis, es posible recoger información sobre los movimientos que realiza el paciente con mayor exactitud. El disponer de esta información reduce la necesidad de una monitorización constante por parte del profesional sanitario y permite un análisis y seguimiento posterior sin pérdidas de precisión.

Actualmente existen diversos dispositivos de este tipo comercialmente disponibles. A continuación se detalla una revisión de algunos de ellos.

Lokomat [18]: Desarrollado por la casa Hocoma, Lokomat ofrece una plataforma de rehabilitación de marcha asistida diseñada para adultos y niños con patologías neurológicas. Contiene un sistema de electromecánico que moviliza el tren inferior para seguir los movimientos de paso durante la marcha. Dispone también de una cinta transportadora que permite que el paciente no se desplace en relación a la sala, manteniendo un contacto con el suelo durante la marcha. Así mismo, dispone de interfaces de realimentación tanto para el paciente como para el profesional sanitario, a través de pantallas integradas (ver Figura 2).



Figura 2: Lokomat Pro. Fuente [18]

Gait Trainer GT II [19]: Desarrollado por Reha-Stim, la plataforma Gait Trainer GT II es una plataforma de rehabilitación de marcha para pacientes adultos con patologías ortopédicas o neurológicas avanzadas. Consiste en un módulo de soporte de peso basado en un arnés, y módulos electromecánicos sujetos en los pies del paciente que simulan el movimiento de marcha de forma determinista. El paciente está rodeado de una estructura de soporte y el profesional cuenta con una pantalla que le informa del estado de la terapia (ver Figura 3).



Figura 3: GT II. Fuente [19]

EksoNR [20]: Desarrollado por Ekso Bionics, la plataforma EksoNR está diseñada para asistir a pacientes con patologías neurológicas en los estados finales de rehabilitación de la marcha. Consta de componentes para el ajuste de movimientos del tren inferior y posturales, presentando al profesional sanitario una pequeña pantalla situada en la espalda del paciente para la visualización de estadísticas y el control del sistema. Adicionalmente cuenta con soportes adicionales como muletas o andadores, que proveen soporte adicional ya que el paciente se desplaza directamente sobre el suelo. El sistema soporta su propio peso, pero no el del paciente (ver Figura 4).



Figura 4: EksoNR. Fuente [20]

Estos dispositivos han probado su efectividad y viabilidad comercial en el campo de la rehabilitación asistida por robots. Sin embargo, han sido diseñados principalmente para su uso en terapias de neurorehabilitación con entrenamientos de marcha, por lo que presentan características que los hacen incompatibles con el uso por pacientes de edad avanzada, especialmente aquellos con fractura de cadera reciente. Entre otros aspectos, algunas limitaciones importantes son la falta de soporte para el equipamiento o el propio paciente, o la dificultad de acceso para un paciente con movilidad y fuerza reducida en el tren inferior.

En vista de la falta de literatura y proyectos enfocados hacia este problema concreto, el grupo de Ingeniería Neural y Cognitiva (gNEC) del CAR propuso la plataforma SWalker, un andador inteligente específicamente diseñado para la rehabilitación de pacientes con fractura de cadera.

1.2 Plataforma Robótica SWalker

El proyecto SWalker tiene como objetivo principal proporcionar una plataforma segura para mejorar la rehabilitación de fractura de cadera en pacientes de edad avanzada. Es por esto que el sistema ha sido diseñado en torno a 3 aspectos principales:

- Seguridad y usabilidad.
- Promover la movilización y el soporte de peso autónomo por el usuario.
- Monitorización para realizar un seguimiento de las sesiones de rehabilitación.

Los pacientes que harán uso del sistema están realizando un proceso de rehabilitación, por lo que presentarán limitaciones en su rango de movimiento y fuerza en sus piernas. Esto se traduce en una reducida capacidad para seguir patrones de marcha estándar, mantener el equilibrio o coordinación motora [21]. Como parte del proceso de recuperación de una fractura, el soporte parcial de peso corporal permite aliviar y mejorar el proceso de curación de los huesos, por lo que es un aspecto a tener en cuenta.

En el apartado de motivación y experiencia de usuario, nos encontraremos con pacientes con variados grados de percepción sobre sus capacidades de recuperación y mejora, generalmente dependiente de sus capacidades cognitivas.



Figura 5: Plataforma Robótica SWalker. Fuente: [1]

El diseño del sistema ha sido realizado de acuerdo a estos principios, y siguiendo las tres directivas principales ya comentadas. La plataforma consiste en un sistema de soporte de peso corporal y tracción motorizada. El paciente se fija a este sistema mediante un arnés, y se le provee de soporte adicional a través de dos barras paralelas ajustables a los lados. En conjunto con el arnés, el paciente se fija a un sistema de módulos electromecánicos que siguen el movimiento de sus piernas, y

que nos proporciona información sobre el rango de movimiento de la cadera. El sistema incorpora una unidad de control que captura la información de los sensores y genera las respuestas de velocidad y elevación necesarias para el movimiento. Esta unidad de control cuenta con mecanismos de comunicación para reenviar la información que recoge y permitir su interacción a través de una interfaz gráfica externa. A través de la interfaz gráfica, el profesional sanitario podrá realizar un seguimiento de la terapia en tiempo real, visualizar la información del paciente y ajustar parámetros relativos a la sesión de rehabilitación.

Analizando más en profundidad el sistema, podemos dividirlo en dos partes principales:

1.2.1 Estructura mecánica

La estructura física sobre la que opera el sistema fue diseñada para garantizar funcionalidad completa para individuos con una altura máxima de 180 cm y 90 kg de peso. Está compuesta por un marco en forma de T sobre 4 ruedas, dos de ellas libres. Para mejorar la estabilidad y percepción de seguridad del paciente, el sistema integra dos barras paralelas ajustables, similares a las utilizadas en terapias de rehabilitación convencional. Estas barras pueden ser retiradas para facilitar el traslado del paciente desde una silla de ruedas al sistema, permitiendo un acceso sencillo al arnés de sujeción para su ajuste. Para asegurar la robustez del sistema, la estructura ha sido fabricada en aluminio con refuerzos de acero para las partes sometidas a mayor estrés mecánico.

Adicionalmente al marco en sí, la estructura mecánica cuenta con los tres siguientes módulos:

Sistema de tracción

Las ruedas traseras están acopladas a motores y motorreductores Kelvin K80 63.105, equipados con encoders que proveen información sobre la velocidad del sistema (ver Anexos A.4 1-2). Este sistema se encarga del movimiento del sistema completo y es capaz de movilizar tanto el peso del equipo como del paciente en caso de suspensión total. Su velocidad máxima está limitada a 0.6 m/s y puede ser ajustada por el profesional sanitario.

Este sistema recibe los valores de velocidad instantáneos desde el sistema de control según el modo de terapia que se está llevando a cabo en cada momento.

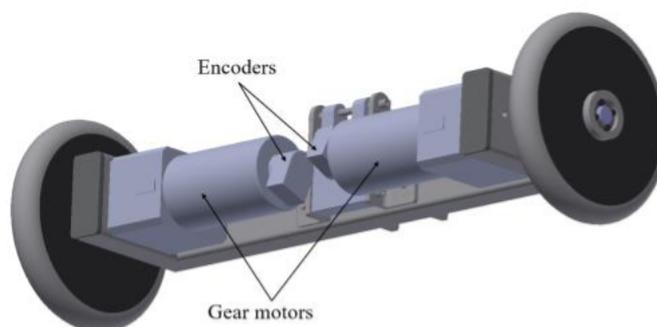


Figura 6: Sistema de Tracción. Fuente: [18]

Sistema de soporte de peso (elevación)

Este sistema está compuesto por dos actuadores cilindro-pistón E21BX300-U-001, Bansbach easylift, posicionados verticalmente, activados por una bomba hidráulica controlada por un motor eléctrico (ver Anexo A.4 3). Este mecanismo cuenta con un sistema de raíl que eleva una estructura rígida, a la que el paciente está firmemente acoplado. El diseño de este sistema permite un soporte de peso con un rango desde el 0% (el paciente está completamente suspendido en el aire y no soporta peso) hasta el 100% (el paciente está completamente en el suelo y soporta todo su peso).

Adicionalmente, el sistema incorpora una galga (tipo FT1 *code CFT1350KC25, REP transducers*) que permiten medir la cantidad de peso soportado por el sistema (ver Anexo A.4 4). Esta información es enviada a la unidad de control a través de un bus CAN.

Sistema de soporte de cadera

Este sistema consiste en una estructura de acero de altura y anchura ajustable. Esta estructura tiene acoplado un arnés ortopédico que permite la estabilización del torso del paciente y el soporte de su peso de forma segura. Adicionalmente, el paciente recibe soporte dinámico en la cadera mediante dos barras de aluminio aseguradas a los muslos. Estas barras tienen un grado de libertad en el plano sagital y siguen el movimiento flexo/extensivo durante la marcha. Este movimiento es transmitido a través de un sistema de poleas a un potenciómetro, lo que permite realizar medidas del rango de movimiento dinámico durante la marcha del paciente.

Mediante procesos de calibración muy simples, este sistema produce medidas de gran precisión, que mediante tarjetas de adquisición de datos (DAQ) independientes para cada lado, son enviadas a través de conexiones CAN a la unidad de control.

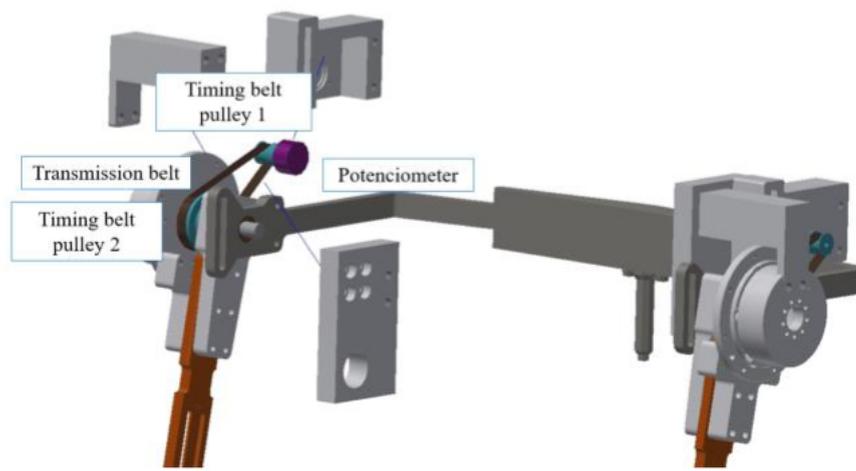


Figura 7: Estructura de fijación y soporte. Fuente: [18]

1.2.2 Arquitectura de control

La arquitectura del control del SWalker está compuesta por dos Raspberry Pi 4 Modelo B conectadas a través de una red Ethernet interna. Estas raspberries poseen denominaciones y un reparto de funciones distinto:

- Pi Master: esta raspberry ejecuta los procesos de control principales del sistema. Incorpora un accesorio CAN-HAT que permite la comunicación a través de buses I2C y CAN (ver Anexo A.4 5).
- Pi Worker: esta raspberry ejecuta los procesos de comunicación y procesamiento adicionales necesarios para el funcionamiento del sistema. Dispone de una serie de puertos GPIO disponibles.

La red de raspberries ejecuta un conjunto de nodos ROS (ver Sección 2.1.3), encargados del funcionamiento completo del sistema. Esta red de nodos obtendrá datos de los distintos sensores del SWalker, realizará procesos de filtrado y preprocesado e introducirá esta información tratada en el sistema ROS. Una vez obtenida esta información se almacenará localmente para su posterior análisis por parte del equipo médico y se utilizará para controlar la marcha de acuerdo a los parámetros de control establecidos. Por último, se generará una salida de velocidades y extensiones para la elevación, para continuar con el movimiento del paciente hasta el fin de la terapia.

1.3 Motivación y objetivos

Como ya se ha comentado en puntos anteriores, la fractura de cadera tiene importantes consecuencias a corto y largo plazo para el paciente, y suponen un descenso considerable en su calidad de vida. Así mismo, representa un problema con afluencia creciente, debido a las tendencias poblacionales comentadas en la Sección 1.1.1. Se ha demostrado que el uso de terapias de rehabilitación basadas en la marcha, con un seguimiento y constancia suficientes, suponen una oportunidad prometedora para lograr una mejora considerable en las capacidades de los pacientes. Desafortunadamente, las limitaciones organizativas y de recursos, hacen que el tratamiento continuado de estos pacientes no sea viable, limitándose a sesiones distanciadas en el tiempo y de corta duración, principalmente por la dificultad de praxis de este tipo de ejercicios por parte de los fisioterapeutas.

El proyecto SWalker presenta la integración de las terapias asistidas por robots al campo de la rehabilitación de marcha para pacientes de fractura de cadera, siguiendo los positivos resultados obtenidos para el uso de estos sistemas en otros campos de la medicina y la rehabilitación. El aprovechamiento de la tecnología y sus capacidades, de forma conjunta y coordinada con los esfuerzos de los profesionales del sector, aplicado a un problema como lo es la fractura de cadera representan el objetivo final del proyecto SWalker.

Dicho esto, en concordancia con este objetivo, presento este trabajo de Fin de Grado, cuyo objetivo es el de continuar el desarrollo y evolución de este proyecto, a través del diseño e implementación del sistema de control principal, haciendo uso de la arquitectura ROS (Robot

Operating System) y el lenguaje de programación Python. Para lograr este objetivo, será necesario la realización de los siguientes apartados:

- Diseño e implementación del circuito de control completo, dentro de la arquitectura de control.
- Diseño, desarrollo e implementación funcional de un sistema de comunicación interno entre nodos.
- Diseño, desarrollo e implementación de un sistema de generación de patrones de velocidad de acuerdo a las características de los modos de funcionamiento.

Capítulo 2

SISTEMA DE CONTROL

En este capítulo, realizaremos una exploración de la arquitectura de control del SWalker. Para ello, realizaremos una revisión de los sistemas de obtención de información de los que disponemos, como se relacionan estos con las redes de comunicación presentes en el SWalker, y los mecanismos de actuación final de los que disponemos.

2.1 Infraestructura de control

El sistema de control del Swalker tiene como objetivo principal, gestionar la entrada, procesado y salida de datos para asegurar un correcto desarrollo de las terapias de marcha. Para realizar estas funciones, hace uso de tres partes principales que conforman la infraestructura de control. Estas partes son las redes de comunicación responsables de la transmisión de datos entre componentes del sistema, el conjunto de sensores y actuadores que captan la información y ejecutan las órdenes, y las arquitectura de programación y comunicación ROS, que sirve como base para el funcionamiento del circuito de control a nivel de software.

2.1.1 Redes de Comunicación

El software de control del SWalker opera sobre 2 Raspberries Pi 4 Modelo B diferenciadas, como ya mencionamos en la Sección 1.2.2. Estas raspberries, denominadas “Pi Worker” y “Pi Master”, contienen las conexiones a los sensores de entrada y a los actuadores de salida, y en ellas se ejecutan todos los componentes de software involucrados en el control del SWalker.

Para poder interactuar con el sistema de forma externa, se añade a este conjunto de componentes un PC externo. Este ordenador permitirá el acceso al conjunto de raspberries para realizar un seguimiento del sistema como desarrolladores, así como manipular parámetros relacionados con la terapia y monitorizar el progreso de la misma, de cara a su uso en pruebas clínicas.

Para que todos los componentes del sistema puedan establecer una comunicación, es necesario que exista una serie de redes adaptadas a las necesidades de transmisión del sistema. Para cubrir esta necesidad, la plataforma SWalker cuenta con dos redes diferenciadas:

- **Red Interna:** En esta red se conectan ambas raspberries y el PC externo a través de enlaces Ethernet y comunicación inalámbrica WiFi. En esta red, las raspberries son accesibles a través de *ssh* gracias a sus conexiones inalámbricas, donde serán identificables a través de IPs fijas. Adicionalmente, las raspberries disponen de una comunicación directa entre ellas gracias a su conexión Ethernet. Todos los parámetros de conexión necesarios para establecer las conexiones estarán contenidos localmente en las raspberries. Si se necesitara modificar los ajustes de conexión,

debido a cambios en la red inalámbrica, la raspberries pueden ser operadas manualmente para introducir dichos cambios y asegurar la conectividad (ver Figura 9).

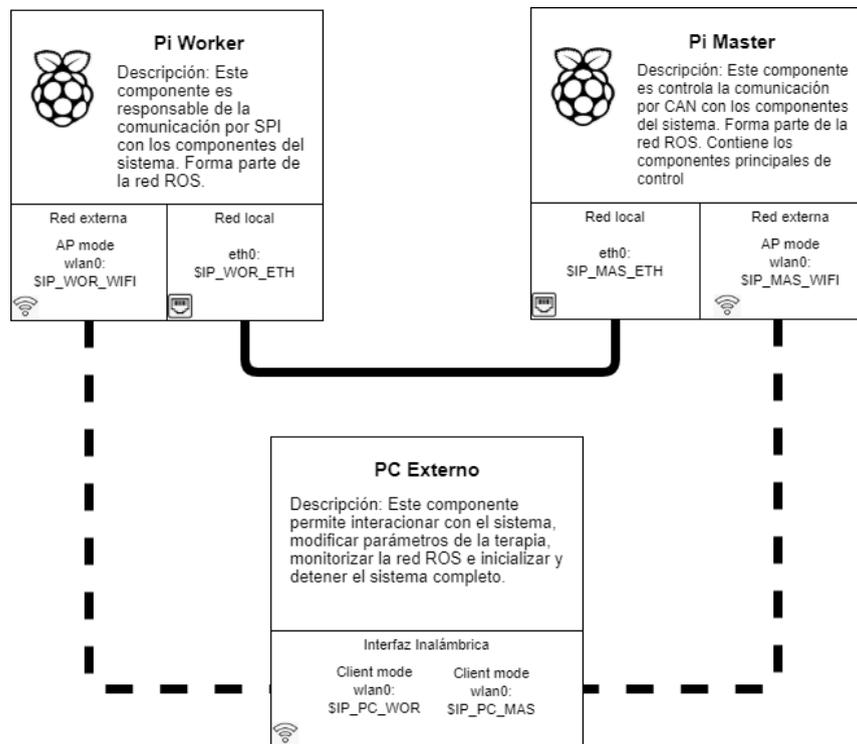


Figura 9: Red Interna SWalker.

- **Red Periférica:** Esta red integra la conexión de los componentes hardware, sensores y actuadores, con la red de raspberries. A través de esta red se realizará la obtención de datos a partir de los diferentes sensores del SWalker y se transmitirán las respuestas a los actuadores, necesarias para el funcionamiento de los componentes (ver Figura 10). Esta red está compuesta por tres circuitos diferenciados:

Circuito SPI: la raspberry Pi Worker cuenta con un conjunto de pines GPIO abiertos. Utilizando estos pines, se realizarán una serie de conexiones utilizando el protocolo Serial Peripheral Interface (SPI), para adquirir los datos obtenidos por los encoders del sistema de tracción [22].

El protocolo SPI permite una comunicación síncrona y de sencilla implementación para componentes electrónicos situados a corta distancia y sigue la estructura esclavo-maestro. Un bus SPI cuenta con 4 canales diferenciados que intervienen en la comunicación, siendo estos SS o *Select*, SLKC, MOSI y MISO.

Para efectuar la comunicación, el maestro selecciona un componente esclavo haciendo uso del canal *Select*. Efectuada la selección, maestro y esclavo intercambian información a través de los canales MISO (“Master Input Slave Output”) y MOSI (“Master Output Slave Input”). La transmisión de información es controlada a través de los pulsos de reloj generados por el componente maestro, transmitidos por el canal SLCK (“Clock”).

Circuito CAN: Gracias al accesorio CAN-HAT acoplado al raspberry Pi Master, es posible integrar el uso del protocolo de comunicación Controller Area Network (CAN) en el sistema. Este circuito será utilizado para realizar la adquisición de datos de los sensores integrados en los sistema de elevación y soporte.

En el protocolo CAN los componentes de la red comparten un único bus de transmisión compuesto por dos cables trenzados, denominados CAN_H (high) y CAN_L (low). Este bus es capaz de transmitir información gracias a la tensión variable de los dos canales [23]. La transmisión de información en forma de bits se basa en la capacidad de estos dos canales diferenciados de situarse en una tensión común, entre -2 y 7 voltios (estado recesivo), o en un estado de diferencia de tensión de al menos 1.5 voltios (estado dominante). En estado dominante los valores de tensión generalmente utilizados para CAN_L y CAN_H son de 1.5 y 3.5 voltios respectivamente. A través de la oscilación entre estos estados, el bus transmite bits 0, situándose en estado recesivo durante un pulso de reloj, o 1 si se encuentra en estado dominante.

El protocolo CAN sigue el modelo productor/consumidor, donde un componente de la red actúa como proveedor de información, generando mensajes a través del sistema. Estos mensajes llevan asignados un identificador y se transmiten a través del bus CAN mediante el uso de tramas. A través de este identificador, los componentes de la red que están actuando como consumidores pueden identificarlo y aceptar o rechazar la recepción del mensaje.

El uso de este tipo de protocolo permite una comunicación de mayor distancia en comparación con SPI, y reduce la cantidad de cableado utilizada. La principal ventaja que ofrece CAN es que a través del uso de este bus común se pueden enlazar varios componentes, permitiendo que todos tengan acceso a la información que se transmite por el bus, a través de un único puerto.

Circuito I2C: El accesorio CAN-HAT acoplado a la raspberry Pi Master, cuenta adicionalmente con componentes que permiten el uso del protocolo I2C. Denominado Circuito Inter-Integrado (Inter-Integrated Circuit, I2C), este protocolo permite la transmisión de información de forma síncrona a través de dos canales únicos, denominados SCL (reloj) y SDA (datos) [24]. En nuestro sistema el uso principal de este circuito es el de sustentar el envío de respuestas para los actuadores del sistema, tanto para el caso de los motores de tracción como para el motor del sistema de elevación.

En este tipo de comunicación, los dispositivos disponen de una dirección identificativa, y se sigue el sistema de multimaestro, que permite que existan varios dispositivos que pueden actuar como maestro, aunque solo uno de ellos puede ejercer este rol durante la comunicación.

Para realizar la transmisión de datos, el maestro envía una señal denominada “start condition”, donde se avisa a los componentes esclavos que se va a iniciar una transmisión. Seguidamente se envía la dirección del dispositivo con el que se va a realizar la comunicación y el tipo de la misma, que puede ser lectura (recepción de datos desde el esclavo) o escritura (envío de datos al esclavo). El esclavo seleccionado responde con un ACK y se inicia la transmisión, respondiendo el esclavo con otro ACK a cada bit recibido. Por último el maestro concluye la transmisión con una señal “stop condition” para liberar el bus.

La principal ventaja que se presenta con el uso de este protocolo es la reducción de pines necesarios para realizar la comunicación, de cara a su integración en tarjetas electrónicas. Este sistema se utiliza frecuentemente para la comunicación con dispositivos con baja tasa de muestreo [25].

2.1.2 Sensores y Actuadores

De acuerdo con lo comentado en la Sección 1.2.1, la estructura mecánica del SWalker está compuesta por tres módulos principales, que están conectados al sistema de control. Estos módulos integran sensores capaces de recoger información del paciente y el desarrollo de la terapia, y actuadores que realizan acciones sobre el sistema físico, en base a una señal digital generada por el sistema. Estos componentes reciben y envían información a través de las redes de comunicación expuestas en la sección 2.1.1. (ver figura 10), y son los siguientes :

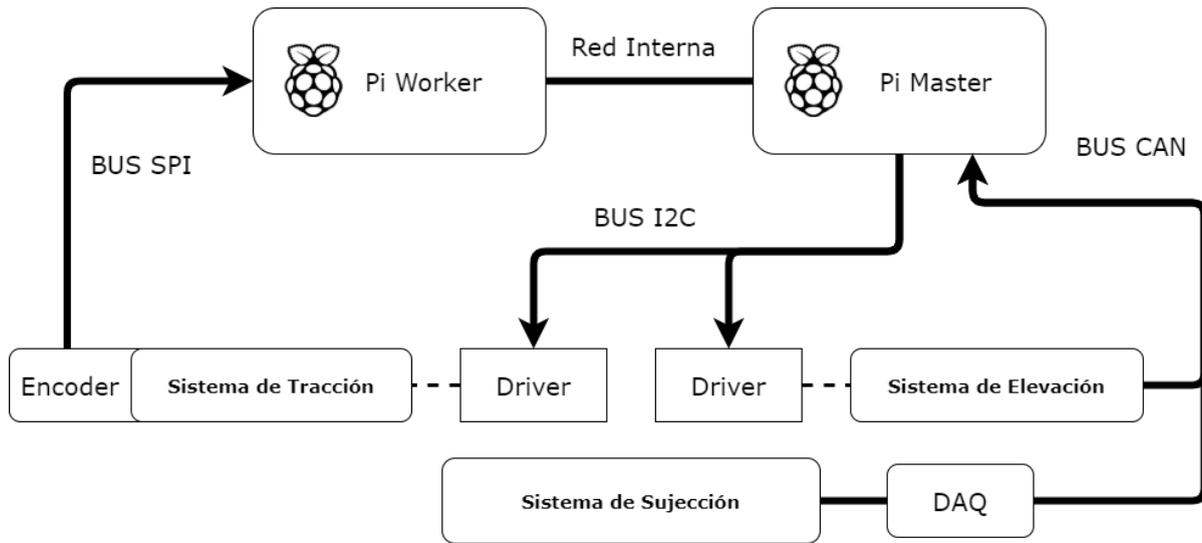


Figura 10. Esquema de conexiones de módulos mecánicos en el SWalker.

Sistema de tracción: su componente principal son la pareja de motores Kelvin K80 responsables del desplazamiento del sistema completo.

Para conocer la velocidad de la plataforma, el sistema de tracción cuenta con una serie de sensores integrados. Estos sensores son los encoders asociados a los motores, comentados en la Sección 1.2.1. Estos encoders, pueden medir las revoluciones que realizan los motores de tracción, y transmitirlas a través de sus conexiones al bus SPI (ver Sección 2.2.1).

Para poder realizar el control de velocidad del sistema, los dos motores del sistema de tracción reciben las respuestas de velocidad de forma individual para cada uno de ellos. El control de los motores se realiza mediante los drivers conectados a los mismos, que reciben las respuestas de velocidad a través de su conexión al bus I2C (ver Sección 2.2.3).

Sistema de soporte de peso (elevación): su componente principal es el pistón hidráulico que permite el soporte de peso del paciente.

Para obtener una medida de la fuerza que soporta el sistema, a fin de monitorizar el porcentaje de peso restante que soporta el paciente sobre su cuerpo, el sistema de elevación cuenta con una galga extensiométrica. Esta galga está conectada al bus CAN del SWalker, gracias a la acción de una tarjeta de adquisición de datos (DAQ). Este dispositivo está programado para devolver, a través de una trama

CAN, el valor de medida de esta galga como respuesta a una petición general o *broadcast* a través del bus CAN (ver Sección 2.2.1).

Para controlar la elevación del pistón, se utiliza una bomba hidráulica controlada por un motor eléctrico. Este motor, a su vez, se controla mediante el envío de respuestas a través del bus I2C a un driver de control, de forma similar a los motores del sistema de tracción

Sistema de soporte de cadera: su componente principal está formado por la estructura de soporte dinámico asociado a los muslos del paciente, la cual sigue el movimiento de las piernas durante la marcha.

Este componente dispone de una serie de sensores para medir los parámetros de este movimiento. Estos sensores se componen principalmente de un potenciómetro y una galga extensiométrica. El potenciómetro genera una salida dependiendo del valor de rotación que experimenta la articulación de la cadera en el plano sagital, dentro de un intervalo limitado. La galga genera una medida de fuerza que corresponde con el torque que genera el paciente sobre el sistema.

Estas medidas son recogidas por una tarjeta de adquisición de datos (DAQ) para cada pierna, similar a la utilizada en la galga del sistema de elevación. Este componente combina las medidas de ambos sensores en una sola trama y la transmite a través del bus CAN como respuesta a una petición *broadcast*. Estas tramas llevan asociado un identificador que permite distinguir de qué extremidad (cadera izquierda o derecha) proviene la medida.

Este módulo mecánico no contiene ningún componente activo, por lo que no contiene conexiones destinadas al envío de respuestas hacia actuadores.

2.1.3 Robotic Operating System (ROS)

ROS (Robotic Operating System - Sistema Operativo Robótico) es un conjunto de herramientas, librerías y estándares que conforman una infraestructura flexible para el desarrollo de sistemas robotizados [26]. ROS ofrece una colección de características especialmente curadas y desarrolladas para facilitar el desarrollo de software de robots de propósito general y representa el esfuerzo conjunto de una comunidad en constante evolución y mejora gracias a el estatus *open-source* o de fuente abierta del proyecto.

La arquitectura ROS se creó con la prioridad de ofrecer la capacidad de diseñar sistemas lo más modulares posibles, permitiendo que esté en manos de los desarrolladores tanto el diseño de las estructuras de software, como el reparto de tareas, así como la posibilidad de realizar una integración completa con otros sistema diferentes. Adicionalmente, el proceso de diseño distribuido de un sistema basado en ROS potencia la colaboración e integración de componentes distintos, que serían incompatibles de otra manera.

Un ejemplo de estas características es visible al comprender cómo se construye una red de nodos ROS (ver Figura 11). Un sistema basado en ROS estará conformado por una red de procesos o “nodos”, independientes unos de otros, y conectados gracias a la infraestructura de comunicación que provee ROS. Esto permite que estos nodos operen por separado, repartan funciones y tengas

desarrollos disjuntos, incluyendo diferencias tan notables como el propio lenguaje de programación de alto nivel que utiliza cada uno.

Los nodos ROS pertenecientes a la red son capaces de comunicarse de forma directa unos con otros, pero existe un nodo maestro o “Master” que actúa como un registro global, coordinando la comunicación. Los nodos independientes acceden al nodo maestro para obtener la información necesaria para establecer la comunicación con otros nodos, que se hace a través de un protocolo común, normalmente TCPROS, que utiliza puertos TCP/IP.

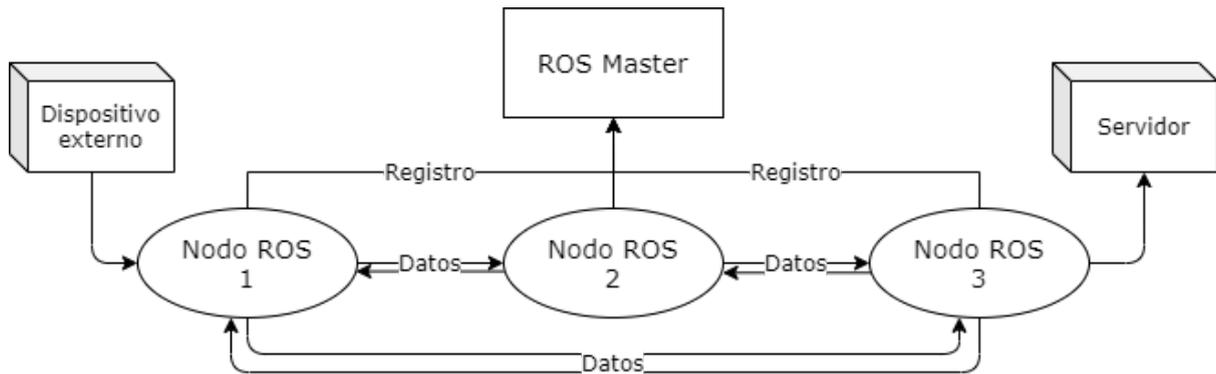


Figura 11. Ejemplo de Grafo Computacional de una Red ROS.

En el caso del SWalker, la versión de ROS utilizada es Melodic Morenia [26] y ofrece la capacidad de programar nodos ROS tanto en Python como en C++ gracias al uso de sus librerías (rospy y roscpp respectivamente). Los nodos ROS conforman lo que se denomina el Grafo Computacional, que es una representación de cómo estos nodos se agrupan y comunican entre sí, haciendo uso de las herramientas de comunicación que provee la infraestructura [28] (ver Figura 12). Estas herramientas son:

Tópicos (topics): los tópicos ROS constituyen la vía de comunicación más común entre nodos y siguen la estructura de publicador/suscriptor. Cada tópico es identificable a través de un nombre único. En estos tópicos, el tipo de información que se publica, entendido como el tipo de datos que se transmiten (booleans, números, strings, etc...), se define cuando se inicializa el primer publicador. Uno o varios nodos pueden actuar como publicadores, transmitiendo información a través de dicho tópico. Una vez creado el tópico al existir al menos un publicador, otros nodos pueden suscribirse a este y recibir los datos cuando son publicados. Un mismo nodo puede actuar al mismo tiempo como publicador y suscriptor. Los tópicos actúan como un canal “instantáneo” de comunicación, lo que implica que no almacenan esta información, simplemente la transmiten.

Servicios (services): los servicios ROS surgen como una alternativa a los tópicos especialmente diseñada para gestionar interacciones de tipo pregunta/respuesta (request/reply) necesarias para sistemas distribuidos o redes que requieran sincronización. Los servicios también se identifican con un nombre específico, pero van acompañados de una definición de la estructura de los mensajes de petición y respuesta, definida por el desarrollador. Para su uso, un nodo específico debe proveer el servicio (actuando como servidor), y los nodos clientes que lo deseen pueden hacer uso de ese servicio (actuando como clientes). El nodo servidor recoge las peticiones y puede gestionarlas a

través de funciones integradas para generar respuestas variadas, con la única limitación de que los mensajes deben cumplir las especificaciones estipuladas en la configuración.

Servidores de parámetros (parameter server): El servidor de parámetros actúa como el almacenamiento compartido y centralizado para todos los nodos, permitiendo acceso a la información a través de claves o “keys”.

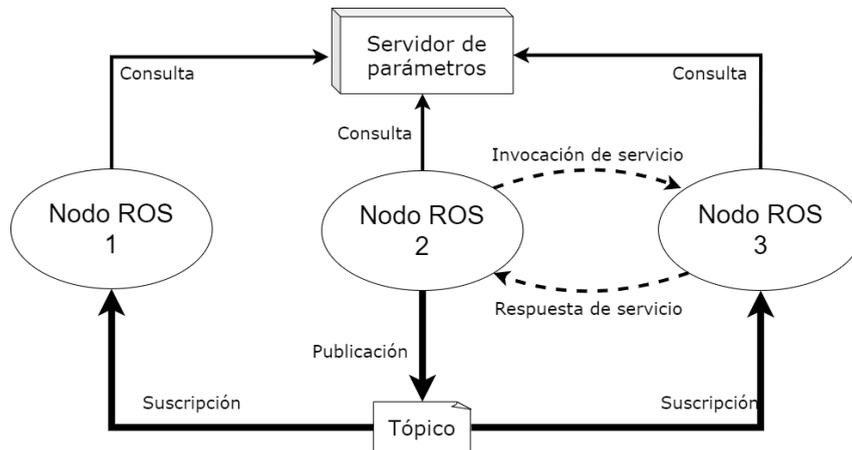


Figura 12: Diagrama de enlace de nodos ROS.

2.2 Circuito de Control

Tal como hemos comentado, el objetivo del sistema de control es capturar la información producida por los sensores del SWalker, procesar y gestionar esta información, y generar las respuestas adecuadas para los actuadores del sistema. Para poder cumplir este objetivo, a la hora de diseñar el circuito de control software en base a la arquitectura ROS, se han diferenciado tres capas principales de nodos, de acuerdo a la función que cumplen (ver Figura 13):

Nivel de entrada (INPUT): En este conjunto de nodos se realiza la conexión y captura de información desde los sensores hardware, a través de las interfaces CAN y SPI de las raspberries.

Nivel de procesamiento (PROCESSING): En este conjunto de nodos se agrupan las funciones principales del sistema una vez obtenida la información de los sensores.

Nivel de salida (OUTPUT): En este conjunto de nodos se realiza la transformación de las respuestas generadas a órdenes transmisibles a través de la interfaz I2C y compatibles con el funcionamiento de los actuadores.

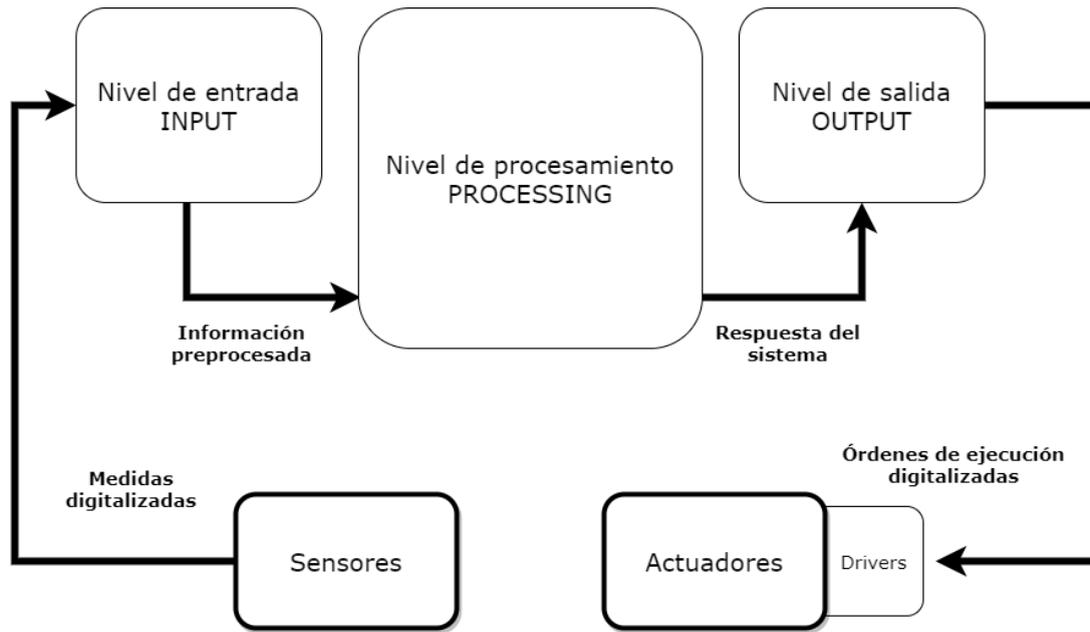


Figura 13: Niveles de funcionamiento. Circuito de control SWalker.

2.2.1 Nivel de entrada

En el nivel de entrada encontramos una serie de nodos cuyo objetivo es capturar los paquetes con las medidas obtenidas de los diferentes sensores del SWalker a través de las redes de comunicación de las raspberries, como se comentó en la Sección 2.1.2. Dependiendo de subsistema mecánico del que reciban información distinguiremos:

Conexión con los DAO del sistema de soporte: Realizada por CAN a través de la interfaz de Pi Master. Con este conjunto de nodos se realizará la captura de las tramas CAN que contienen las información de los sensores asociados al movimiento de la cadera del paciente.

Los DAQ integran en una misma trama la información de los sensores disponibles para cada pierna (potenciómetro y galga). Estos DAQ devuelve una trama CAN con la medida actual de estos cuatro sensores codificada como 4 valores hexadecimales para cada sensor, ocupando los 4 bytes más significativos del total de 8 que se transmite en cada mensaje CAN. Este mensaje es enviado con un identificador que permite distinguir la pierna derecha (RIGHT_HIP - ID: 75), de la pierna izquierda (LEFT_HIP - ID: 70). Estos mensajes son enviados como respuesta a un mensaje CAN de *broadcast* (ID: 68) generado desde el sistema de control, a modo de estabilizar el muestreo. La interfaz CAN permite un flujo máximo de 1Mb/s.

Estas tramas hexadecimales son deconstruidas y procesadas por separado para su conversión en medidas en el sistema internacional. Para eliminar el ruido presente en la señal previo a realizar las conversiones, se utilizan dos tipos de filtros, dependiendo del origen de la señal a tratar.

En el caso de las medidas de la galga, la señal se trata utilizando un filtro de segundo orden de Respuesta Finita al Impulso (Finite Impulse Response, FIR), donde el valor de salida generado se produce como combinación del valor de entrada actual y un número finito de valores de entrada

anteriores. Estos valores de entrada se tratan utilizando los coeficientes asociados a este filtro, actuando como pesos para el valor de salida [29].

Para las medidas del potenciómetro, se utiliza un filtro Bicuadrático (Biquadratic, biquad). Este tipo de filtros de segundo orden, incorporan tanto una respuesta asociada a un filtro de respuesta finita como uno de respuesta infinita al impulso. Estos filtros representan estructuras estables y robustas para el procesamiento de señales, especialmente en el caso de eliminación de ruido. Una vez realizado el filtrado las medidas del potenciómetro son convertidas en ángulos siguiendo una tabla de conversión incluida en sistema. Mediante esta conversión es posible determinar el ángulo de la pierna del paciente en la articulación de la cadera con respecto a la vertical.

Este conjunto de funciones es realizado por dos nodos ROS independientes denominados “sensor_aquisition_node” y “sensor_processing_node” (ver Figura 15). El nodo de adquisición se encarga de la captura de las tramas CAN y envía los datos hexadecimales decodificados a través de un conjunto de tópicos, que incluyen un tópico por cada sensor y por cada pierna, resultando en un total de 4 tópicos. El nodo de procesado se suscribe a estos tópicos y realiza el filtrado y conversión de los valores hexadecimales a medidas en el sistema internacional. La información procesada se publica a través de 4 nuevos tópicos, que comprenden fuerza y ángulo actual en cada una de las extremidades.

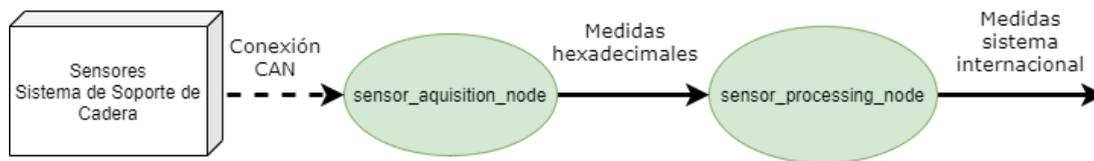


Figura 15. Circuito de nodos ROS de adquisición para el sistema de Soporte de Cadera

Conexión con la galga del sistema de elevación: Realizada a través de la interfaz CAN de Pi Master. Comparte las características de captura y envío de tramas CAN con la conexión del sistema de soporte, respondiendo a mensajes de *broadcast* con tramas conteniendo la información de los sensores, junto con un identificador específico (ELEVATION - ID: 80). En el caso de elevación únicamente se transmiten 4 valores hexadecimales conteniendo la información de la galga, manteniendo la trama total en 64 bits, ocupando únicamente los 2 primeros bytes más significativos.

A la hora de realizar el procesado de los datos obtenidos de la galga, se reduce su ruido mediante el uso de un filtro FIR, similar al utilizado en la galga del sistema de soporte. Realizado esto, se transforma la medida digitalizada del sensor a un valor de peso soportado por el sistema en kilogramos.

Este conjunto de funciones se realiza en el sistema a través de tres nodos diferentes. “ELEVATION_aquisition” es responsable de la captura de las tramas CAN y la publicación de los datos asociados a la galga en hexadecimal a través de un tópico ROS. “ELEVATION_offset” permite introducir modificaciones al estado inicial del sistema si es necesario ajustar la elevación de forma manual. “ELEVATION_odom” procesa la información generada por “ELEVATION_aquisition” y la combina con las modificaciones introducidas por “ELEVATION_offset”. “ELEVATION_odom” publicará a través de un nuevo tópico la información procesada final del estado del sistema de elevación (ver Figura 16).

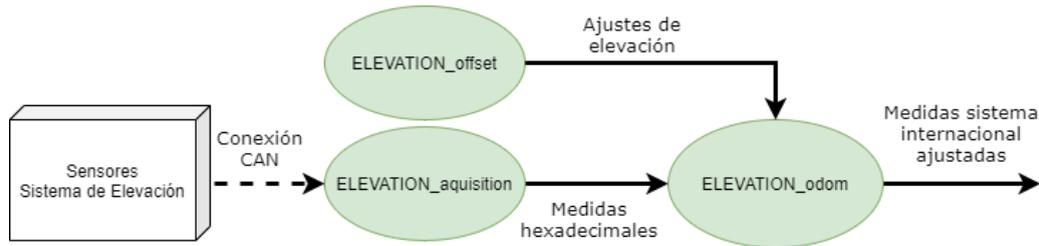


Figura 16. Circuito de nodos ROS de adquisición para el sistema de Elevación.

Conexión con los encoders del sistema de tracción: Realizada a través del bus SPI de Pi Worker. Este conjunto de nodos realiza lecturas periódicas de los encoders acoplados a cada motor del sistema de tracción. Estas lecturas se pueden realizar directamente a través de la librería de gestión SPI “Python-spidev”. Debido a que esta medida se obtiene directamente del encoder asociado al motor de forma digital, no es necesario que el sistema realice un proceso de filtrado.

Las consultas realizadas a los encoders devuelven las revoluciones que ha realizado el motor. Conociendo la diferencia de tiempo entre consultas, es posible convertir estas medidas en rotaciones por unidad de tiempo (rotaciones por minuto, rpm). Obtenidas las rotaciones por minuto, se utilizan las unidades conocidas de diámetro de las ruedas para obtener las velocidades asociadas a cada una. Estas velocidades son posteriormente utilizadas para generar medidas de velocidad angular y lineal del sistema completo, así como para realizar un seguimiento de la variación de su posición.

Este conjunto de funciones se realiza a través de tres nodos ROS independientes. Estos nodos son “ODOM_encoder_aquisition”, “ODOM_motor_velocity” y “ODOM_Walker_odom”. “ODOM_encoder_aquisition” realiza la consulta con el encoder para cada motor por separado y publica esta información en dos tópicos. Esta información es recogida por “ODOM_motor_velocity”, quien realiza la conversión de cuenta de revoluciones a velocidad de cada rueda, publicándose en un nuevo tópico. Por último, esta información se combina en “ODOM_Walker_odom” para generar una salida de odometría combinada para todo el sistema, que se publica en un único tópico (ver Figura 17).

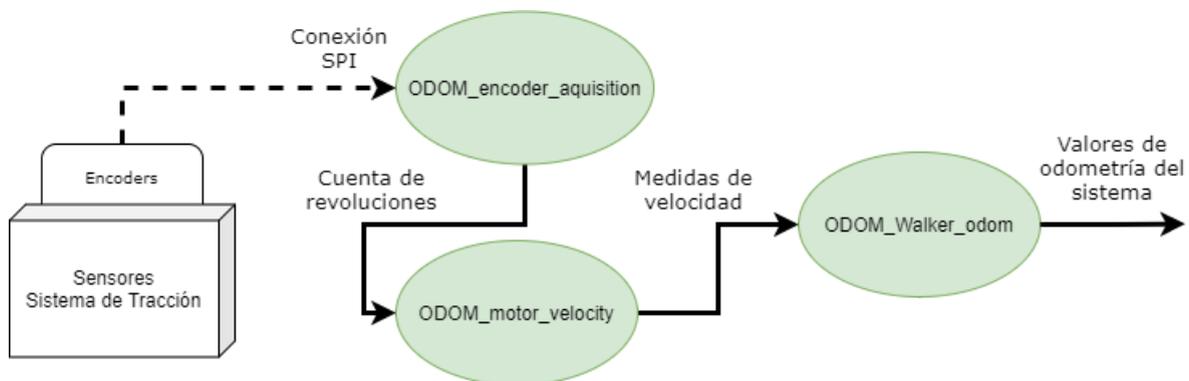


Figura 17. Circuito de nodos ROS de adquisición para el sistema de Tracción

2.2.2 Nivel de procesamiento

En el nivel de procesamiento encontramos los nodos responsables de integrar la información obtenida y tratada por el nivel de entrada, y realizar las transformaciones necesarias para obtener una salida útil. En este nivel se generan las respuestas necesarias para que el sistema completo realice la terapia junto con el paciente.

Para poder realizar un seguimiento de la terapia, el sistema cuenta con un perfil de marcha objetivo que el paciente debe seguir para realizar la sesión correctamente. Este perfil se define a través de una serie de medidas objetivo para los ángulos de las articulaciones de la cadera y rodilla a lo largo del ciclo de marcha. Estos perfiles representan el movimiento asociado a un ciclo completo en una marcha natural, que incluye dos pasos, uno con cada pierna. Compuestos por 200 muestras por perfil, las medidas numéricas representan valores angulares asociados al sistema de referencia indicado en la Figura 18.

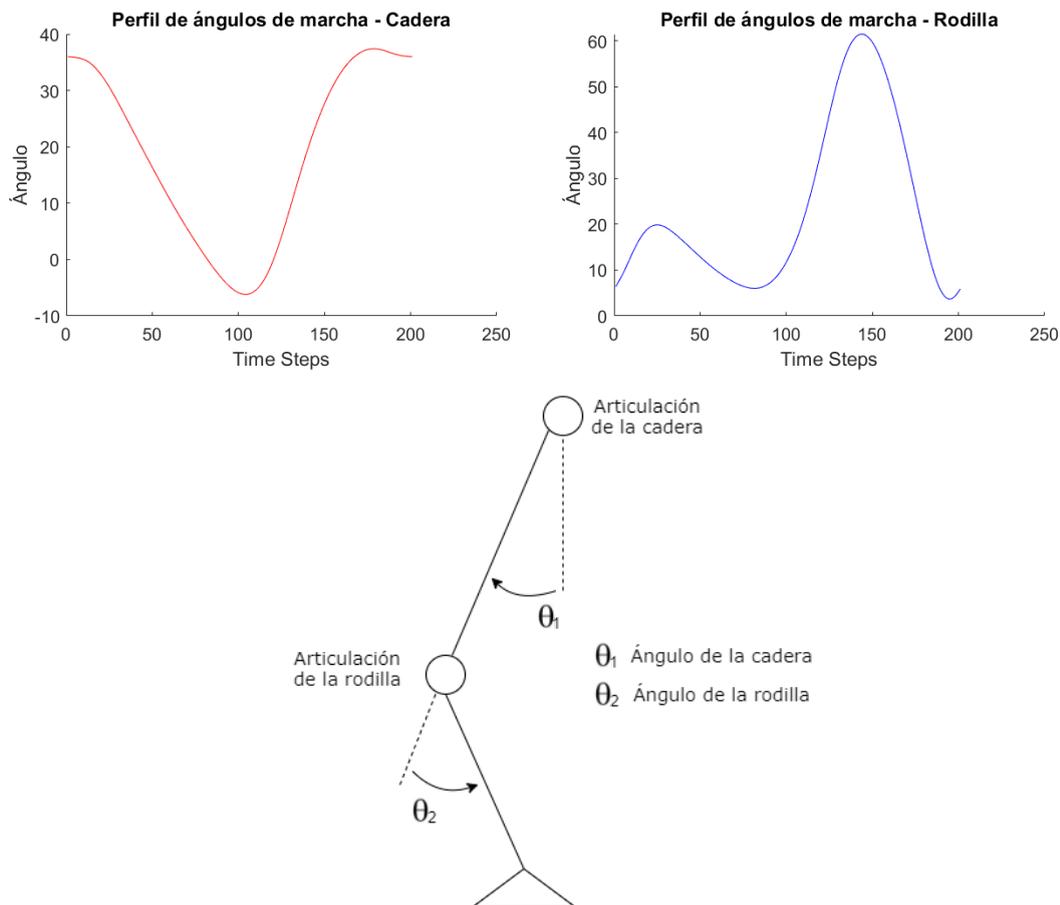


Figura 18. Perfiles de ángulos de marcha y modelo biomecánico para la pierna.

La existencia de estos perfiles nos permite estimar cómo se desplazan las piernas del paciente con respecto a su cadera a lo largo de la marcha. El modelo biomecánico de referencia se debe ajustar a los parámetros del paciente para realizar un seguimiento adecuado. Esto conlleva la modificación de las medidas de longitud para los segmentos de la pierna, siendo estos el muslo y la pierna propiamente

dicha (desde la rodilla hasta el tobillo). Combinando ambos perfiles de acuerdo al modelo biomecánico ajustado, es posible generar un perfil de movimiento bidimensional del pie del paciente, con respecto a la cadera del mismo (ver Figura 19).

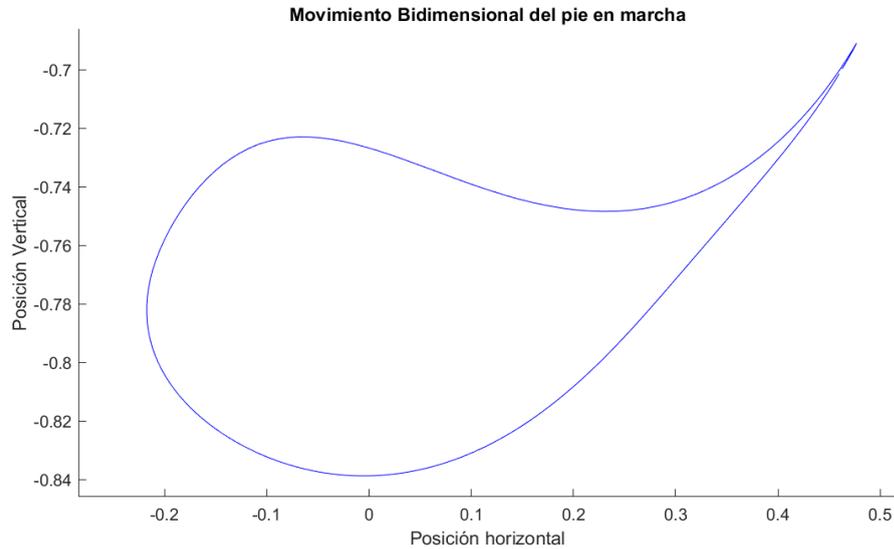


Figura 19. Movimiento bidimensional del pie durante la marcha.

Este movimiento representa una marcha suave, donde el paciente realiza los pasos de forma continua, existiendo siempre un punto de contacto con el suelo. Gracias a estas características, es posible desacoplar el movimiento del pie del marco de referencia de la cadera. Esto se debe, como se representa en la Figura 20, a que cuando el pie experimenta una velocidad negativa con respecto a la cadera, lo cual se corresponde con el pie estando apoyado en el suelo, este está realizando una fuerza sobre el cuerpo, provocando el avance durante el paso y manteniendo su posición fija en el espacio. Esta situación permite que esa pareja de velocidades verticales y horizontales se transmitan al cuerpo del paciente, pudiendo ser utilizadas por el sistema como la variación de elevación de la cadera y la velocidad lineal del sistema.

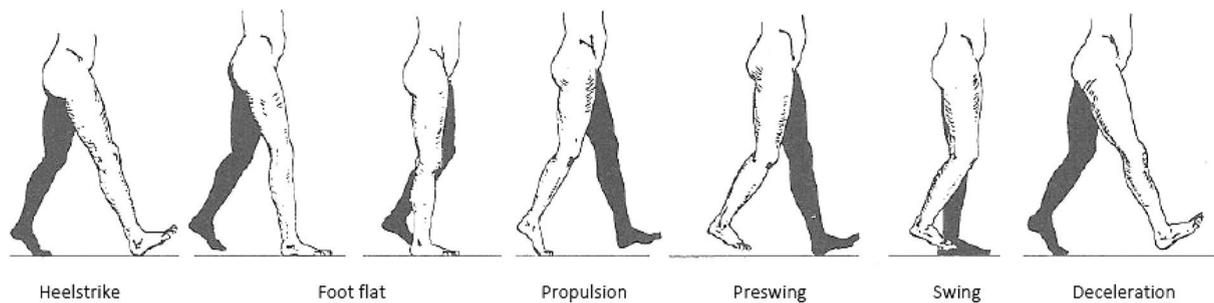


Figura 20. Fases del ciclo de marcha. Fuente: [30]

Sin embargo, estos perfiles pueden ser diferentes de la secuencia específica de pasos que esté realizando el paciente, así que es necesario que se realicen ajustes a este modelo de marcha objetivo de acuerdo a las características concretas del paciente y de la sesión de terapia (ver Sección 3.2).

2.2.3 Nivel de salida

Una vez procesada la información necesaria para poder continuar la terapia, es necesario que esta información se transforme en respuestas para los actuadores del sistema, capaces de ser enviadas a través de la interfaces disponibles.

Para poder generar estas respuestas de control, se utiliza un controlador PID (Proporcional Integral Derivativa) [32]. Un controlador de este tipo utiliza una medida *setpoint* o referencia, que representa el estado objetivo del sistema, y una medida del estado actual del mismo. Utilizando estos dos estados, se calcula la diferencia o error de ambas señales, se realiza un tratamiento de este a través de los tres componentes que integran el controlador y se adapta para producir una secuencia interpretable por el actuador. Estas transformaciones asociadas al error pretenden maximizar la estabilidad futura del sistema, reduciendo variaciones y posibles efectos oscilatorios. Para la generación de los controladores PID en el sistema, se hace uso de la librería “simple_pid” de Python [31]. Los tres componentes del procesado realizado por un controlador PID son:

- **Proporcional:** Compuesto por un producto de la señal de error y una constante proporcional que actúa como parámetro de ajuste. Esta parte del controlador aporta la respuesta instantánea a la diferencia entre estados. Carece de memoria o capacidades predictivas, produciendo efectos de sobreoscilación al alcanzar su valor objetivo debido a la inercia del sistema.
- **Integral:** Compuesto por la integral de la señal de error desde el punto inicial hasta el actual, multiplicada por la ganancia integral, como parámetro de ajuste. Este componente nos permite tener en cuenta los errores pasados, actuando como la memoria del sistema. Permite reducir el efecto de artefactos ocurridos durante el estado estacionario, que provocan oscilaciones si se gestionan únicamente a través del modo proporcional.
- **Derivativo:** Compuesto por la derivada de la señal de error en el instante actual, multiplicado por la ganancia derivativa, como parámetro de ajuste. Permite realizar una predicción del estado futuro del sistema en base a las tendencias actuales, y ajustar acordeamente la respuesta.

Gracias al uso de este tipo de controladores, la salida de control generada a partir de la referencia o estado objetivo se regula de tal forma que evitamos cambios excesivamente bruscos o efectos oscilatorios indeseados. En el sistema de control SWalker, estos controladores se ejecutan para el control de la velocidad de tracción, por separado para cada rueda, y uno adicional para el control del motor responsable del sistema de elevación.

Una vez obtenidos los parámetros objetivos, siendo estos la velocidad objetivo, lineal y angular del sistema, y la extensión objetivo del sistema de elevación, estos parámetros se introducen, junto con los valores de medida actual, al controlador PID para generar la respuesta que se enviará a través de las interfaces. El envío de estas señales de control se realiza a través del bus I2C conectado a Pi Master.

Para llevar a cabo estas funciones se utiliza únicamente un nodo ROS, “MOTORS_motor_control”. El utilizar un único nodo permite evitar la competencia al utilizar el bus I2C, gracias a la centralización de los procesos de lectura y escritura. Este nodo obtiene toda la

información que necesita de los tópicos de parámetros objetivo generados por el nivel de procesamiento, y los de medidas actuales generadas por el nivel de entrada (ver figura 21).

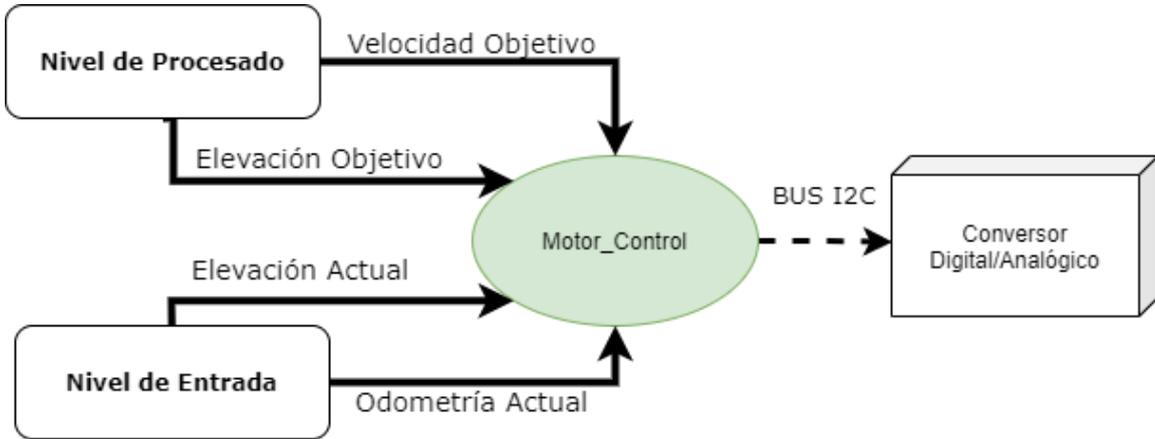


Figura 21. Circuito de nodos ROS para el nivel de salida

Capítulo 3

RESULTADOS DE DESARROLLO

En este capítulo se realizará una exploración de los conjuntos de nodos ROS y herramientas modulares añadidos como parte del desarrollo de la plataforma SWalker. Estas herramientas tienen como objetivo facilitar la realización de las tareas necesarias para el correcto funcionamiento del sistema, entre las que se incluyen la comunicación y monitorización interna del sistema, y la generación de patrones de marcha para las sesiones de terapia.

3.1 Sistema de estados interno

Para poder conseguir un funcionamiento continuado, robusto y seguro de la plataforma robótica, es necesario que existan mecanismos capaces de realizar un seguimiento de la evolución de las distintas partes de la red ROS.

Tal como se expuso en la sección 2.2, el sistema de control del SWalker funciona gracias al flujo continuado de información, lo que presenta riesgos si no se dispone de medidas para reaccionar ante posibles fallos en partes concretas del sistema.

Debido a ello, es necesario que exista una forma de que cada nodo, dada su independencia a nivel de procesamiento, sea capaz de conocer el estado de aquellos otros nodos de los que dependa para su funcionamiento. Para poder obtener esta información, primero es necesario definir un conjunto de códigos capaz de contener la información sobre el ciclo de vida de un nodo, e interpretable por los mismos. Este sistema de códigos, debe ser capaz de reflejar cuales son las características actuales de funcionamiento del nodo concreto, y su posible evolución.

Para lograr esto, se ha diseñado un sistema de ciclo de vida basado en “estados”. Estos estados definen las características de funcionamiento actuales del nodo, y son almacenados localmente en cada nodo, constituyendo una lista personalizada para cada uno (ver Figura 22).

Una vez establecido el sistema de estados, es necesario que esta información sea accesible para aquellos nodos que lo necesiten. Gracias al acceso a esta información, cada nodo podrá efectuar cambios en su funcionamiento, y por extensión, en su estado, de acuerdo a los estados que presenten sus nodos vecinos. Para lograr esto, es necesario que el sistema disponga de dos funcionalidades principales.

Capacidad de consulta: Cada nodo debe ser capaz de realizar consultas a otros nodos a fin de comprobar su estado. Esto requiere que exista un canal de comunicación directo nodo-nodo, que permita el intercambio de mensajes.

Capacidad de cambio: Cada nodo debe ser capaz de modificar su propio estado cuando sea necesario, mediante el uso de funciones y condiciones personalizadas. Las características que pueden propiciar un cambio de estado pueden ser únicamente propias (evolución natural del ciclo de funcionamiento del nodo), o propiciadas por cambios en otros nodos.

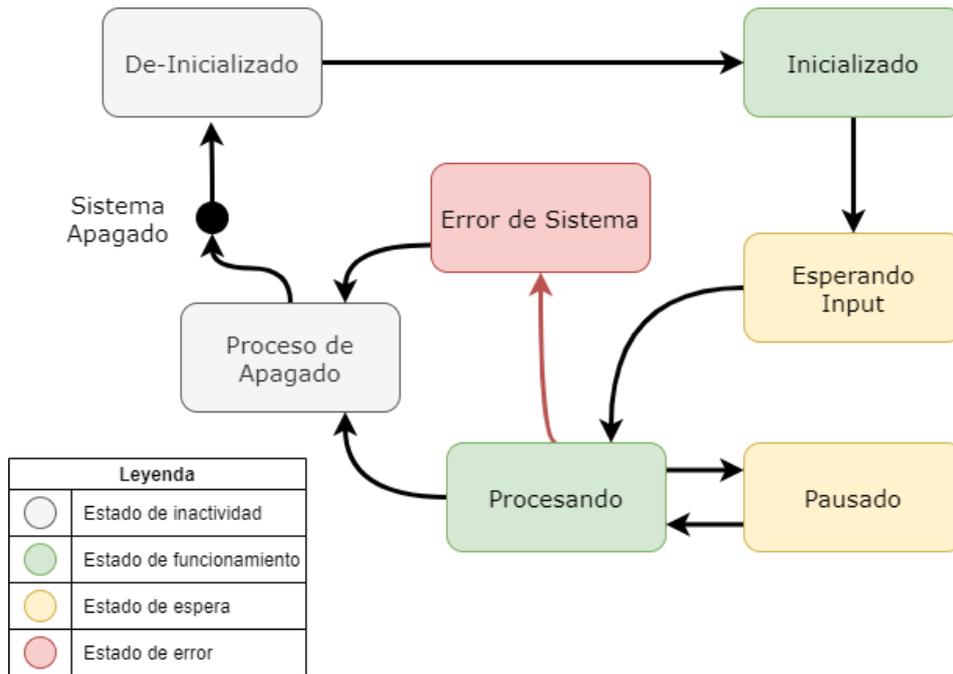


Figura 22. Ejemplo ciclo de vida por estados de un nodo ROS SWalker.

Por último, las características ya mencionadas de este sistema facilitan la extensión del mismo hacia un sistema de monitorización y control distribuido. Esto se reduce a incluir la capacidad de que un nodo reaccione a una consulta concreta modificando su estado, o realizando una función específica.

Estas características constituyen un sistema de monitorización interno, que permiten realizar un seguimiento automático y localizado de la salud de la red, y reaccionar ante posibles cambios en esta. Así mismo, provee a cada nodo los mecanismos necesarios para obtener esta información, alterarse a sí mismo en consecuencia, y realizar cambios en otros nodos si así fuera necesario.

Teniendo en cuenta todas estas características, es necesario realizar una implementación haciendo uso de las herramientas que provee ROS, así como el lenguaje de programación de alto nivel Python. El resultado de este desarrollo es la librería “State Service Util” o S_S. Esta librería está diseñada para ser importada dentro de cada nodo ROS de la red, y provee las instrucciones necesarias para gestionar el estado interno del nodo, así como realizar las consultas y órdenes pertinentes a los nodos que lo requieran.

La librería “State Service Util” hace uso de los servicios ROS para gestionar la comunicación directa nodo-nodo. Cada nodo ejecutará sus funciones de forma secuencial, progresando por los estados asociados a su ciclo de forma normal. Si en algún punto de su ciclo, el nodo requiere interactuar con otro de alguna manera, éste puede consultar el estado de dicho nodo gracias al servicio ROS disponible (ver Sección 2.1.3).

El funcionamiento de S_S se basa en dos roles principales que puede ejercer cada nodo, con respecto a la comunicación:

- **Nodo servidor / Monitorizado:** Un nodo que actúe como servidor deberá llevar un registro de su propio estado, y recibirá consultas y peticiones desde otros nodos. Para realizar estas funciones debe disponer de herramientas que le permitan:
 - Almacenar, consultar y modificar su estado.
 - Recibir y responder a consultas y peticiones.
 - Distinguir entre consultas distintas y realizar operaciones en base a ellas.

- **Nodo cliente / Monitor:** Un nodo que actúe como cliente deberá ser capaz de realizar conexiones con otro nodo o conjunto de nodos, enviar consultas y peticiones, y almacenar las respuestas. Adicionalmente, debe ser capaz de reaccionar a las respuestas que recibe, produciendo cambios en sí mismo. Para realizar estas funciones, se debe disponer de herramientas que permitan:
 - Almacenar la lista de nodos objetivo de la comunicación.
 - Establecer comunicación con otros nodos.
 - Modificar y comprobar la integridad de las conexiones con los nodos.
 - Almacenar las respuestas a las consultas enviadas.
 - Reaccionar ante respuestas específicas.

Para ejemplificar este proceso, se propone su implementación entre dos de los nodos responsables de la comunicación con los sensores del SWalker, contenidos en el nivel de entrada. Como se ilustra en la Figura 23, el nodo responsable de la captura de información actuará como servidor, y el nodo que trata esta información decodificada, como cliente.

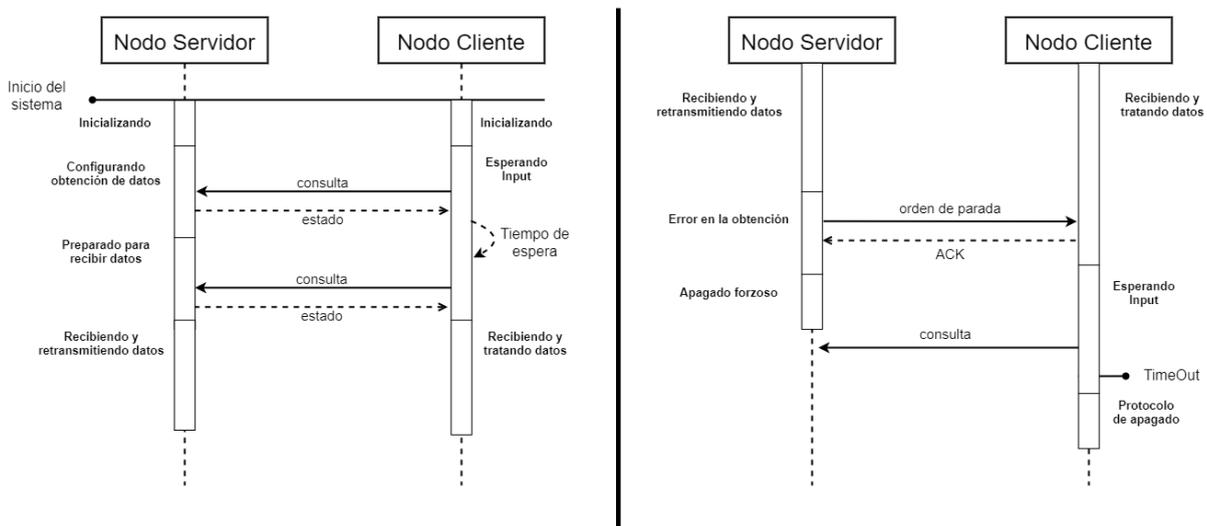


Figura 23. Diagrama de secuencia de dos casos de funcionamiento coordinado entre dos nodos ROS SWalker. Izquierda: Inicialización. Derecha: Error de comunicación.

En la parte izquierda de la figura, podemos observar cómo ambos nodos son inicializados cuando se realiza el arranque del sistema. El nodo servidor continúa con sus procesos de configuración necesarios para realizar la comunicación con los sensores. El nodo cliente ha terminado su configuración, y está esperando recibir la información de los sensores de parte del nodo de

adquisición, que actúa como servidor. Debido a esto, realiza una consulta al nodo servidor para comprobar si está listo para comenzar el funcionamiento. Debido a que el estado del nodo no es el correcto, el nodo cliente comienza una espera para realizar otra consulta. Una vez el nodo servidor alcanza el final de su configuración, espera a recibir una consulta con el estado concreto del nodo cliente, encargado de gestionar esta información. Cuando se produce la consulta, ambos nodos son conscientes de que el otro está preparado para la comunicación, y comienza el intercambio de información, de acuerdo a lo comentado en la Sección 2.2.1.

En caso de eventos inesperados, como puede ser un error de comunicación, los nodos pueden reaccionar ante ello utilizando las herramientas del “State Service Util”. En el caso reflejado en la parte derecha de la Figura 23, el nodo de adquisición experimenta un error fatal en la lectura de datos, que provoca un apagado forzoso. Antes de terminar el proceso, el nodo reacciona enviando una señal de pausa al nodo cliente, para que detenga el procesado. Debido al apagado forzoso, el sistema no puede transmitir una señal de apagado al resto del sistema. Para evitar un bloqueo del resto de nodos, el nodo cliente vuelve a solicitar el estado del nodo servidor. Debido al apagado, este no será capaz de responder, y ocurrirá un *timeout* del mensaje, indicando al nodo cliente que el servidor ha dejado de estar operativo. Con esta información, el nodo cliente puede efectuar los pasos que necesite para continuar con las funciones del sistema, siendo en este caso el apagado del nodo.

De cara a su implementación, cada nodo puede actuar como servidor y como cliente al mismo tiempo. Para asegurar la correcta monitorización de todo el sistema, es necesario que todos los nodos actúen como servidor, para poder consultar su estado en todo momento, si así fuera necesario.

Configuración del servidor.

Para iniciar la configuración como servidor, los nodos dispondrán de una función concreta, “Service_setup”. Esta función recibirá como parámetro el nombre que recibe el nodo dentro de la red ROS, como un “string”. Seguidamente, el sistema procederá a levantar el servicio ROS de forma que sea accesible con el nombre “node_name_state_service”, donde “node_name” se sustituirá por el nombre del nodo. Esta instrucción requiere que se le suministre el archivo de configuración del servicio, tal como se indica en la Sección 2.1.3.

Una vez inicializado el servicio, éste dispondrá de una configuración por defecto, conteniendo un estado inicial con valor “De-Initialized”, y una serie de respuestas comunes a consultas determinadas, como lo son “ping”, que devuelve un respuesta simple para indicar que la conexión está activa, o “state”, que devuelve el estado interno del nodo.

Para poder modificar y consultar el estado en el que se encuentra, ya que este se almacena como una variable en la librería “State Service”, se pone a disposición del nodo funciones para recuperar y modificar este valor, denominada “get_state” y “set_state” respectivamente.

Debido a que cada nodo necesita realizar operaciones concretas como respuesta a consultas distintas, la función manipuladora de mensajes o *handler* contiene un apartado diseñado para ser sobrescrito una vez se importa en el nodo ROS.

Configuración del cliente.

Para iniciar la configuración como cliente, cada nodo dispondrá de una función que levantará las conexiones a cada nodo servidor, gracias al uso de los servicios ROS. Para poder crear estas

conexiones, será necesario proveer al sistema ROS del nombre del servicio objetivo y el archivo de configuración necesario. Esta conexión se almacenará en un diccionario, utilizando como clave asociada el nombre del nodo objetivo, para facilitar su posterior acceso. Estas conexiones son creadas por defecto sin realizar comprobaciones sobre el estado del servicio de cara al servidor, por lo que es posible realizar una consulta cuando el servidor no está disponible.

Completado el establecimiento de servicios, el nodo cliente dispondrá de una función específica “Command_send” que le permitirá realizar consultas a un nodo objetivo. Esta función tomará como parámetros el nombre del nodo objetivo, a fin de identificar su conexión dentro del diccionario previamente mencionado, así como el contenido de la consulta a enviar. Esta función tiene la limitación de que tanto el mensaje de consulta como el de respuesta, deben adecuarse a los parámetros estipulados en el archivo de configuración de servicio.

Este sistema presenta una integración modular y de mínima invasión en los nodos que lo integran. Su desarrollo como una librería de Python permite que sus cambios se realicen de forma centralizada, lo que reduce el riesgo de desactualización de código en cada nodo. Adicionalmente, su adaptabilidad a las necesidades específicas de cada nodo, así como su potencial para la expansión fuera de las funcionalidades por defecto aportan flexibilidad a la hora de realizar su función de comunicación. Gracias al uso de este sistema, la secuencialización de procesos y la monitorización de progreso de conjuntos de nodos se puede realizar de una forma ordenada y controlada.

3.2 Sistema de control de velocidad

En el momento de realización de este Trabajo de Fin de Grado, el SWalker realiza la asistencia a la marcha a través de un único modo de funcionamiento principal. En este modo, denominado “Marcha Determinada” o “Set Walk”, el paciente realiza la marcha de acuerdo a un patrón de velocidades objetivo definido al comienzo de la sesión.

Este patrón de velocidad se genera a partir de la aplicación de los perfiles de movimiento objetivo de la cadera y rodilla, comentados en la Sección 2.2.2. Debido a las posibles limitaciones en movilidad que pueda padecer el paciente, será necesario realizar diversos ajustes en estos perfiles para asegurar un correcto desarrollo de la terapia. Estos ajustes producirán un nuevo conjunto de perfiles de movimiento objetivo ajustados al paciente y a las características de la terapia impuestas por el profesional sanitario.

Ajustes de amplitud: Debido a la condición de los pacientes que utilizarán el sistema, es común que estos pacientes presenten una reducción del rango de movimiento de sus piernas. Esto requiere que los perfiles de movimiento utilicen un intervalo de valores reducido, lo que conlleva un suavizado de la marcha.

Para poder realizar este ajuste, será necesario que el sistema disponga de un intervalo de valores que representen el rango de movimiento que el paciente es capaz de ejecutar. Este intervalo puede ser manualmente introducido por el profesional, o medido utilizando el SWalker, previo a la marcha. Debido a que el SWalker no dispone de sensores para monitorizar el movimiento de la rodilla, si no se introduce un rango de movimiento específico, se generará uno utilizando las proporciones de reducción aplicadas en el perfil de la cadera sobre el rango de movimiento del perfil original de la rodilla.

En base a estos intervalos, el sistema realizará un escalado en amplitud de los perfiles de movimiento objetivo, en base a la Fórmula 3.1:

$$g(t) = \left(\frac{f(t) - f_{min}}{f_{max} - f_{min}} * (g_{max} - g_{min}) \right) + g_{min} \quad (3.1)$$

Donde $g(t)$ representa el nuevo perfil de movimiento generado, $f(t)$ el perfil de movimiento original, f_{max} y f_{min} el máximo y mínimo del perfil de movimiento original, y g_{max} y g_{min} los extremos del intervalo tomado como rango de movimiento del paciente.

El resultado de este ajuste es un conjunto de perfiles de movimiento que conservan la forma de señal de los perfiles originales, ajustando sus valores máximos y mínimos a una versión compatible con la movilidad del paciente. Estos nuevos perfiles se ajustarán de una forma más fiel al movimiento que realizará el paciente (ver Figura 24).

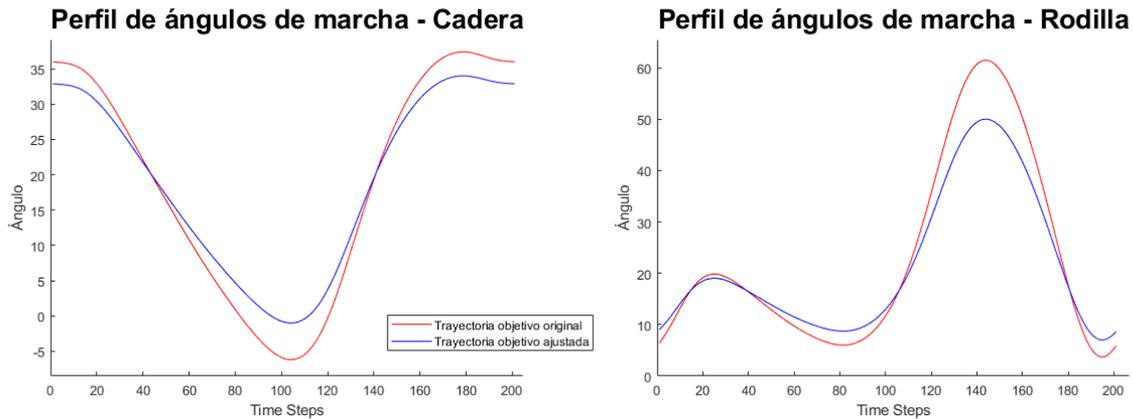


Figura 24. Patrones de movimiento objetivos ajustados a rangos dinámicos reducidos.

Ajustes temporales: Obtenidos los perfiles de movimiento ajustados, para modificar el patrón de velocidad generado será necesario modificar la duración de cada ciclo de marcha.

De acuerdo a lo expuesto en la Sección 2.2.2, el patrón de velocidades generado depende directamente de la duración de estos ciclos, por lo que un aumento en la duración de cada ciclo implica menos pasos por unidad de tiempo, y por tanto menos velocidad de movimiento. El profesional sanitario podrá seleccionar previo a la terapia la duración de cada ciclo, y obtendrá una previsualización del patrón de velocidades generado.

Los perfiles de movimiento originales disponen de 200 muestras, recogidas durante un ciclo de 4 segundos de duración, con una tasa de muestreo de 100 Hz.

Para realizar este ajuste, se espaciarán las muestras del perfil original, de forma que abarquen uniformemente el intervalo de tiempo especificado. Esto tendrá como resultado la extensión de la señal a lo largo del nuevo intervalo de tiempo, aumentando el tiempo entre muestras como consecuencia. Para combatir este efecto y conservar la tasa de 100 Hz de los perfiles originales, se realizará una interpolación de *splines* del perfil con las muestras espaciadas.

El resultado de este ajuste será un conjunto de perfiles de movimiento con una duración mayor pero conservando la densidad de muestras por unidad de tiempo. Los nuevos perfiles mantendrán las características de señal de los perfiles originales, asegurando que no se altere el movimiento de marcha.

En la Figura 25 podemos observar el ajuste temporal de los perfiles a un intervalo de 6 segundos, que equivale a 300 muestras a 100 Hz. Para ilustrar el espaciado, se ha resaltado una serie de valores en el perfil original, así como su correspondencia en el perfil ajustado.

En base a estos nuevos perfiles, se puede generar el patrón de velocidad a seguir durante la marcha. En la Figura 26, se presenta una comparativa de los patrones generados a partir de los perfiles de movimiento sin el ajuste temporal (en rojo), y tras realizar este ajuste (azul).

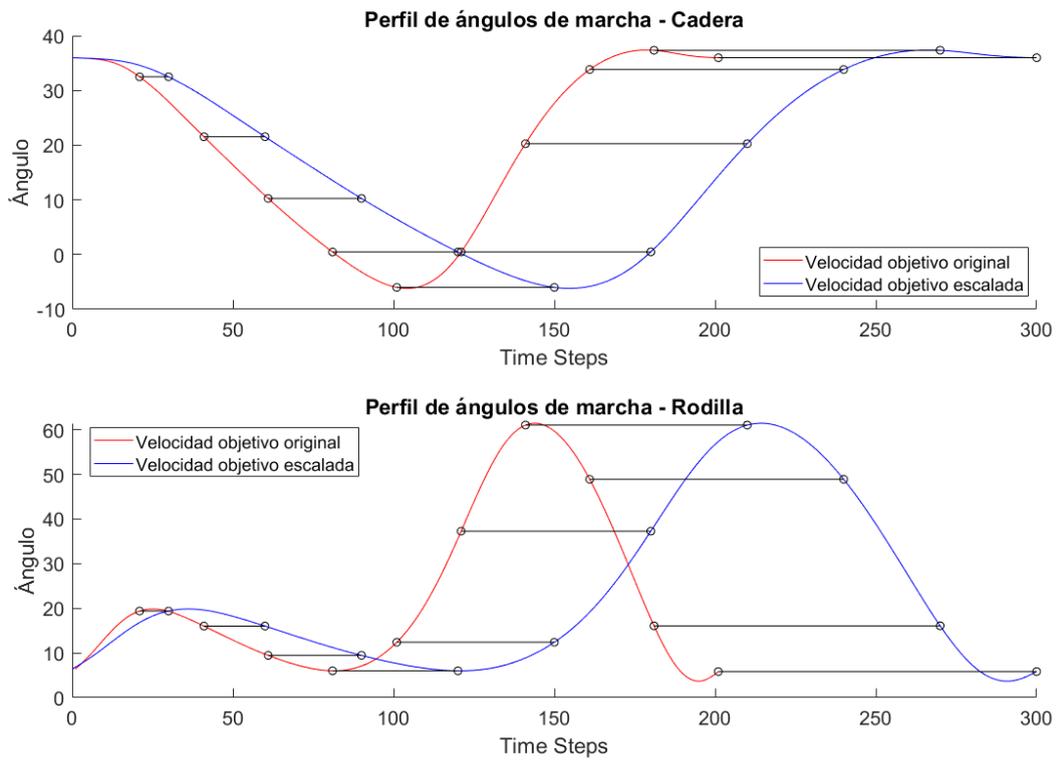


Figura 25. Patrones de movimiento articular ajustados temporalmente.

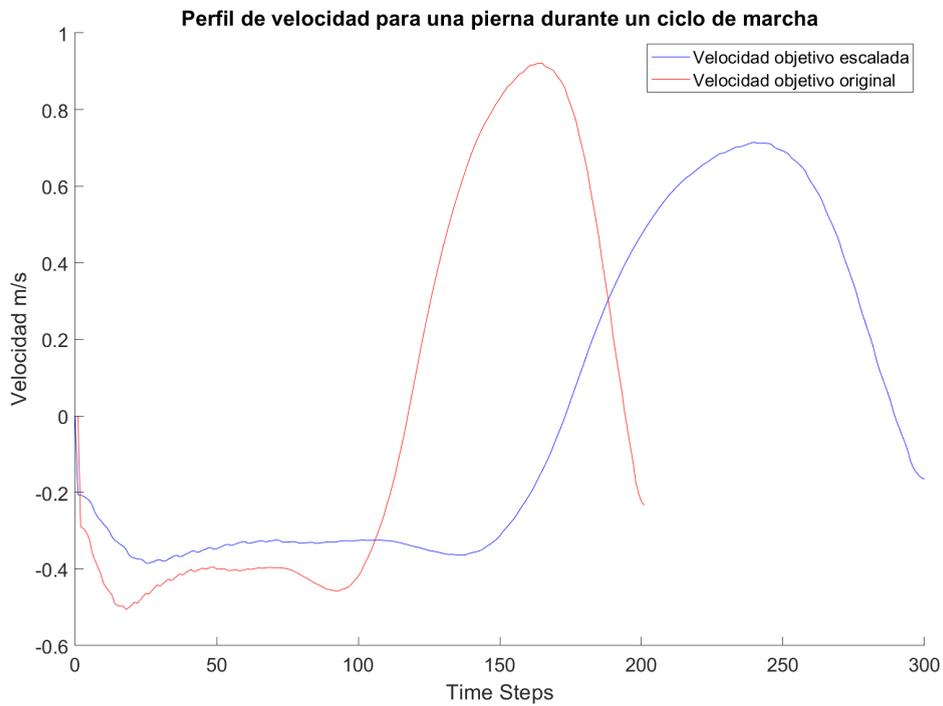


Figura 26. Perfil de velocidad generado en base a los patrones temporalmente ajustados.

Encadenamiento: Como parte de la terapia de marcha, el paciente realizará un número determinado de pasos para completar una sesión. El ajuste final necesario para generar el patrón de velocidades que guiará la sesión consistirá en encadenar un número determinado de ciclos de los perfiles de movimiento, acorde a la cantidad de pasos objetivo.

El número de pasos a realizar será introducido por el profesional sanitario previo a la terapia. Este ajuste, requiere únicamente que se concatenen varias copias de los perfiles de movimiento modificados utilizando los ajustes anteriores, debido a que los perfiles originales han sido tratados para generar una señal continua y compatible con este proceso.

Realizados estos ajustes, los perfiles de movimiento han sido ajustados a las características tanto del paciente como de la sesión a realizar. Si se genera un perfil de movimiento bidimensional del pie del paciente, tal como se expone en la Sección 2.2.2, obtenemos un movimiento más suave y con una amplitud menor, correspondiente con el movimiento previsto para un paciente con movilidad reducida (ver figura 27).

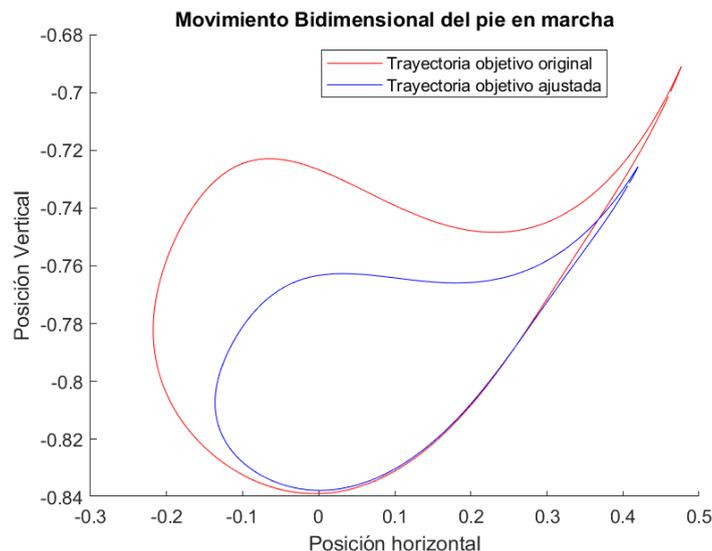


Figura 27. Comparación de perfiles de movimiento originales y ajustado para pacientes con movilidad reducida.

Gracias a esta nueva simulación de movimiento, podemos aplicar los principios de marcha expuestos en la Sección 2.2.2 para extraer la velocidad del sistema. Para ello, será necesario realizar un cruce de dos patrones desfasados, para tener en cuenta el efecto de las dos piernas. El resultado final será un patrón de velocidad combinado para el sistema completo. Este patrón puede ser transmitido al nivel de salida para generar las respuestas necesarias para mantener la velocidad determinada por el patrón a medida que progresa la marcha.

Capítulo 4

CONCLUSIONES Y TRABAJOS FUTUROS

El objetivo de este Trabajo de Fin de Grado es el de proveer a la plataforma SWalker de un sistema de control basado en la arquitectura ROS. Este sistema, basado en los tres niveles de procesamiento indicados en la Sección 2.2, provee de una sólida base para la comunicación y el tratamiento de información necesaria para realizar las terapias de marcha.

La incorporación al sistema de las herramientas de control de estados internos, y el sistema de control de velocidad aportan las características necesarias para poder realizar una primera implementación completa del sistema. El desarrollo de las mismas, ha respetado los principios de modularidad y accesibilidad que el sistema requería, permitiendo una implementación sencilla y abierta a posibles nuevas modificaciones, tanto en lo relativo al propio sistema de control, como a la evolución de la plataforma física.

La incorporación de estos nuevos sistemas proveen a la plataforma de las capacidades necesarias para gestionar su propia información de cara a tener un funcionamiento ordenado, lógico y controlable, tanto manual como automáticamente. Así mismo, no reduce las posibilidades de expansión y modificación de la plataforma, sino que provee una base sobre la que construir.

En base a esto, la plataforma SWalker dispone de un amplio abanico de posibilidades de expansión y mejora, entre las que destacamos una serie de posibles líneas de trabajo futuras:

- Integración completa del sistema de comunicación interna en todos los nodos del sistema.
- Diseño de ciclos de vida para todos los nodos, teniendo en cuenta las dependencias de funcionamiento cruzadas entre distintos nodos de la red.
- Desarrollo de un nodo de monitorización y control general de todo el sistema.
- Expansión del sistema de control de velocidad para la generación dinámica de perfiles de marcha en tiempo real.
- Desarrollo de un sistema de preanálisis de marcha, capaz de evaluar el movimiento del paciente de acuerdo al patrón generado.

La plataforma SWalker presenta una importante oportunidad de desarrollo en el ámbito de la rehabilitación robotizada. Su naturaleza modular y su enfoque accesible, permiten que sea posible la ramificación exponencial en lo que a potenciales funcionalidades se refiere. Estas líneas de desarrollo constituyen las proyecciones más inmediatas para posibles características futuras del proyecto.

BIBLIOGRAFÍA

1. Costa, V., Ramírez, O., Lora-Millan, J.S., Urendes, E., Rocon, E., Perea, L. and Raya, R., Design of a robotic platform for hip fracture rehabilitation in elderly people. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)* (pp. 599-604). IEEE.
2. Costa, V., Ramírez, Ó., Perea, L., Velásquez, A., Otero, A., Rocon, E. and Raya, R., 2021. SWalker: a robotic platform for hip fracture rehabilitation.
3. Ham Chande, Roberto. 'Envejecimiento demográfico. Sin soluciones sobre la sobrevivencia decorosa de la población mayor'. Revista: Demos. Año de publicación: 2003. Número 16. Páginas: 42-43. ISSN: 0187-7550. En Español.
4. Ec.europa.eu. 2021. [online] Available at: <<https://ec.europa.eu/futurium/en/system/files/ged/eprs-briefing-633160-demographic-trends-eu-regions-final.pdf>> [Accessed 12 June 2021].
5. Nguyen, N.D., Ahlborg, H.G., Center, J.R., Eisman, J.A. and Nguyen, T.V. (2007), Residual Lifetime Risk of Fractures in Women and Men. *J Bone Miner Res*, 22: 781-788. <https://doi.org/10.1359/jbmr.070315>
6. Icks, A., Haastert, B., Wildner, M. et al. Trend of hip fracture incidence in Germany 1995–2004: a population-based study. *Osteoporos Int* 19, 1139–1145 (2008). <https://doi.org/10.1007/s00198-007-0534-6>
7. Ec.europa.eu. 2021. *File:Population structure by major age groups, EU-28, 2016-80 (% of total population).png - Statistics Explained*. [online] Available at: <[https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Population_structure_by_major_age_groups,_EU-28,_2016-80_\(%_of_total_population\).png](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Population_structure_by_major_age_groups,_EU-28,_2016-80_(%_of_total_population).png)> [Accessed 18 June 2021].
8. Hernlund, E., Svedbom, A., Ivergård, M., Compston, J., Cooper, C., Stenmark, J., McCloskey, E., Jönsson, B. and Kanis, J., 2013. Osteoporosis in the European Union: medical management, epidemiology and economic burden. *Archives of Osteoporosis*, 8(1-2).
9. Oskar, S., Borgström, F., Zethraeus, N., Johnell, O., Lidgren, L., Ponzer, S., Svensson, O., Abdon, P., Ornstein, E., Ceder, L., Thorngren, K., Sernbo, I. and Jönsson, B., 2008. Long-term cost and effect on quality of life of osteoporosis-related fractures in Sweden. *Acta Orthopaedica*, 79(2), pp.269-280.
10. Dyer, S., Crotty, M., Fairhall, N., Magaziner, J., Beaupre, L., Cameron, I. and Sherrington, C., 2016. A critical review of the long-term disability outcomes following hip fracture. *BMC Geriatrics*, 16(1).
11. Sherrington, C., Tiedemann, A. and Cameron, I., 2011. Physical exercise after hip fracture: an evidence overview. *Eur J Phys Rehabil Med*, 47(2), pp.297-307.
12. Liu, C.J. and Latham, N.K., 2009. Progressive resistance strength training for improving physical function in older adults. *Cochrane database of systematic reviews*, (3).

13. Howe, T.E., Rochester, L., Neil, F., Skelton, D.A. and Ballinger, C., 2011. Exercise for improving balance in older people. *Cochrane database of systematic reviews*, (11).
14. Verfaillie, D.F., Nichols, J.F., Turkel, E. and Hovell, M.F., 1997. Effects of resistance, balance, and gait training on reduction of risk factors leading to falls in elders. *Journal of Aging and Physical Activity*, 5(3), pp.213-228.
15. Maranesi, E., Riccardi, G.R., Lattanzio, F., Di Rosa, M., Luzi, R., Casoni, E., Rinaldi, N., Baldoni, R., Di Donna, V. and Bevilacqua, R., 2020. Randomised controlled trial assessing the effect of a technology-assisted gait and balance training on mobility in older people after hip fracture: study protocol. *BMJ open*, 10(6), p.e035508.
16. Taveggia, G., Borboni, A., Mulé, C., Villafañe, J.H. and Negrini, S., 2016. Conflicting results of robot-assisted versus usual gait training during postacute rehabilitation of stroke patients: a randomized clinical trial. *International journal of rehabilitation research. Internationale Zeitschrift fur Rehabilitationsforschung. Revue internationale de recherches de readaptation*, 39(1), p.29.
17. Carpino, G., Pezzola, A., Urbano, M. and Guglielmelli, E., 2018. Assessing effectiveness and costs in robot-mediated lower Limbs rehabilitation: a meta-analysis and state of the art. *Journal of healthcare engineering*, 2018.
18. Hocoma. 2021. *Lokomat® - Hocoma*. [online] Available at: <<https://www.hocoma.com/us/solutions/lokomat/>> [Accessed 14 June 2021].
19. Reha-Stim. 2021. *Gait Trainer GT II - Reha-Stim*. [online] Available at: <<https://reha-stim.com/gt-ii/>> [Accessed 14 June 2021].
20. Carlan, A., 2021. *EksoNR - The Next Step in NeuroRehabilitation - Ekso Bionics*. [online] Ekso Bionics. Available at: <<https://eksobionics.com/eksonr/>> [Accessed 13 June 2021].
21. Rehr T. et al. "The Robot ALIAS as a Database for Health Monitoring for Elderly People". In: Wichert R., Klausning H. (eds) *Ambient Assisted Living. Advanced Technologies and Societal Change*. Springer, Berlin, Heidelberg, 2014 [23]D. C. Kerrigan, L. W. Lee, J. J. Collins, P. O. Riley, and L. A. Lipsitz, "Reduced hip extension during walking: Healthy elderly and fallers versus young adults," *Arch. Phys. Med. Rehabil.*, vol. 82, no. 1, pp. 26–30, 2001.
22. Es.wikipedia.org. 2021. *Serial Peripheral Interface - Wikipedia, la enciclopedia libre*. [online] Available at: <https://es.wikipedia.org/wiki/Serial_Peripheral_Interface> [Accessed 16 June 2021].
23. Es.wikipedia.org. 2021. *Bus CAN - Wikipedia, la enciclopedia libre*. [online] Available at: <https://es.wikipedia.org/wiki/Bus_CAN> [Accessed 16 June 2021].
24. Aprendiendo Arduino. 2021. *I2C*. [online] Available at: <<https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/>> [Accessed 16 June 2021].
25. Es.wikipedia.org. 2021. *I²C - Wikipedia, la enciclopedia libre*. [online] Available at: <<https://es.wikipedia.org/wiki/I%C2%B2C#Uso>> [Accessed 16 June 2021].

-
26. Ros.org. 2021. *ROS.org | About ROS*. [online] Available at: <<https://www.ros.org/about-ros/>> [Accessed 14 June 2021].
 27. Wiki.ros.org. 2021. *melodic - ROS Wiki*. [online] Available at: <<http://wiki.ros.org/melodic>> [Accessed 14 June 2021].
 28. Wiki.ros.org. 2021. *ROS/Concepts - ROS Wiki*. [online] Available at: <<http://wiki.ros.org/ROS/Concepts>> [Accessed 14 June 2021].
 29. Es.wikipedia.org. 2021. *FIR (Finite Impulse Response) - Wikipedia, la enciclopedia libre*. [online] Available at: <[https://es.wikipedia.org/wiki/FIR_\(Finite_Impulse_Response\)](https://es.wikipedia.org/wiki/FIR_(Finite_Impulse_Response))> [Accessed 16 June 2021].
 30. Gommans, L.N., Smid, A.T., Scheltinga, M.R., Cancrinus, E., Brooijmans, F.A., Meijer, K. and Teijink, J.A., 2017. Prolonged stance phase during walking in intermittent claudication. *Journal of vascular surgery*, 66(2), pp.515-522.
 31. The Profesor García. (2019, July 16). *Entendí el Control PID, si lo aprendes vas a poder...* [Video File]. YouTube: <https://www.youtube.com/watch?v=gtsZ2hswKJk>
 32. PyPI. 2021. *simple-pid*. [online] Available at: <<https://pypi.org/project/simple-pid/>> [Accessed 16 June 2021].

ANEXOS

A.1 Impacto Socio-económico

La fractura de cadera es una patología fuertemente asociada con el envejecimiento de la población. Su alta incidencia en paciente con una edad superior a 60 años, así como las crecientes tendencias demográficas globales, convierten a la fractura de cadera en una realidad cada vez más común entre la población.

Tal como se expresó en la Sección 1.1.1, el sufrir una fractura de cadera tiene importantes efectos económicos asociados. Estos costes aplican un estrés creciente tanto en la economía personal del propio paciente, como en el sistema sanitario del mismo. Adicionalmente, un paciente que sufra esta clase de fractura tiene un bajo pronóstico de mejora debido a las limitaciones en los tratamientos. Esto produce una alta tasa de dependencia funcional derivada para los pacientes, y tiene graves efectos en su calidad de vida.

Este Trabajo de Fin de Grado ha constituido un impulso al desarrollo de una plataforma para la rehabilitación de pacientes de cadera, especialmente en fases tempranas de su recuperación. La realización de este tipo de terapias tiene asociados efectos positivos en cuanto a los pronósticos de mejora, lo que ayuda a mitigar los efectos anteriormente comentados. Adicionalmente, la facilitación de la realización de este tipo de terapias gracias a la asistencia ofrecida por la plataforma de cara a las sesiones de terapia, permite acelerar y reducir la carga de trabajo asociada a cada sesión, como medida para reducir costes económicos y de tiempo asociados al personal sanitario.

A.2 Impacto ético y profesional

El progreso continuado de campos como la robótica, la inteligencia artificial y la ciencia de datos a la largo de la última década ha generado nuevas preguntas y cuestiones en cuanto a la ética y las capacidades de estas tecnologías en diversos campos. Un área especialmente sensible es la de la tecnificación de los procesos de salud, especialmente en el área de cuidado y rehabilitación de pacientes.

Entre estas cuestiones, las más inmediatas surgen en relación al uso directo de tecnologías como lo son las plataformas robotizadas de rehabilitación. Estas cuestiones pueden surgir a un potencial usuario del sistema cuando debe decidir si quiere utilizarlo, e incluyen cuestiones como la fiabilidad, tolerancia a fallos y posibles accidentes. La literatura nos muestra que existe una creciente cantidad de estudios que prueban la seguridad y potenciales beneficios asociados al uso de plataformas robotizadas, pero el total de estudios puede ser considerado aún escaso. El desarrollo de este tipo de plataformas, sigue no solo los estándares y buenas prácticas asociadas a los profesionales de campos técnicos, como puede ser las ingenierías o las ciencias de la información, sino aquellos impuestos para su aplicación en campos de salud. La meticulosidad y rigurosidad no sólo durante el diseño, sino durante el desarrollo y evaluación, culminando con una validación técnica y clínica, presentan una serie de evidencias que cubren todos los estándares necesarios para asegurar un correcto funcionamiento del sistema, manteniendo la seguridad del paciente como primera prioridad.

Adicionalmente, es posible que se despierten inquietudes acerca de la moralidad de la inserción de este tipo de tecnologías en el campo. Estas cuestiones, surgen de un potencial riesgo a una monitorización excesiva, debido a la captura de información necesaria para el funcionamiento de los sistemas. Sobre esto, el argumento de la deshumanización o la reducción de personal puede generar un rechazo hacia el uso de esta clase de sistemas, aún con los beneficios comentados. Esta clase de inquietudes se producen debido principalmente a la falta de conocimiento en el sector, un sano escepticismo y la influencia de la cultura popular de los últimos años.

Para resolver estas cuestiones es necesario conocer que los sistemas desarrollados tienen el objetivo de trabajar en conjunto con el personal sanitario, a fin de reducir su carga de trabajo y permitir un aprovechamiento más pleno de sus facultades en tareas que de verdad lo requieran. Un fisioterapeuta trabajará en conjunto con el SWalker, supervisando y dirigiendo la terapia, permitiendo no solo reducir su fatiga, sino conseguir un máximo beneficio para el paciente, quien recibe una atención completa, supervisada y continua. En lo relativo a la protección de datos, el SWalker así como otras plataformas similares, trabajan no solo bajo normativas europeas de protección de datos, sino que limitan la recolección de esta información a los parámetros médicos necesarios, siendo su único uso objetivo el seguimiento médico del paciente.

A.3 Presupuesto

Como parte de la realización de este Trabajo de Fin de Grado, se incluye un desglose del coste asociado a los recursos empleados para la realización de el mismo. Estos recursos han sido provistos por el “Consejo Superior de Investigaciones Científica”, la “Universidad Politécnica de Madrid” y la “Escuela Superior de Ingenieros de Telecomunicación”.

Para la realización del desarrollo expuesto, ha sido necesario disponer de ambas Raspberry Pi 4B, así como un total de 3 tarjetas SD, con una capacidad de 32 GB de almacenamiento, que han actuado como la memoria de dichas raspberries. Con respecto a las tarjetas, debido a que tanto la distribución de ROS, la imagen utilizada para el sistema operativo (Raspbian) y todos los paquetes adicionales de desarrollo constituyen proyecto gratuitos de código abierto, no han tenido un impacto en el presupuesto.

Adicionalmente, el accesorio CAN-HAT acoplado a Pi Master, se ha adquirido por separado. Para realizar la configuración de todos los componentes, así como la quema de las imágenes en las tarjetas, ha sido necesario el uso de un ordenador externo, y periféricos tales como un monitor y teclado (en este caso con un ratón integrado) para el control de las raspberries. Debido a sus características, el accesorio CAN-HAT ha requerido recibir alimentación de una fuente externa.

Para realizar las pruebas necesarias para implementar los algoritmos, así como para desarrollar parte de los sistemas, ha sido necesaria la utilización del programa MATLAB. Para este presupuesto, se incluye la adquisición de la licencia de un año, que se ajusta más adecuadamente al periodo de realización de este trabajo de entre las disponibles.

Junto con los costes de material necesarios, se incluyen los recursos humanos asociados, en la forma de los honorarios abonados a un estudiante de ingeniería en prácticas durante la duración del periodo de desarrollo del proceso.

El total de costes asociados asciende a 10736 € para el desarrollo de los sistemas incluidos en este Trabajo de Fin de Grado.

	Unidades	Coste por Unidad (€/U)	Total (€)
Licencia MATLAB	1	800	800
Raspberry Pi 4B	2	65	130
Accesorio CAN-HAT	1	25	25
Ordenador Externo	1	600	600
Fuente de Alimentación	1	60	60
Monitor externo	1	60	60
Teclado combinado	1	25	25
Tarjeta SD	3	12	36
	Horas totales	Coste por hora (€/hr)	Total (€)
Honorarios estudiante de ingeniería	500	18	9000
TOTAL			10736

A.4 Fichas de datos

1. Motor reductor Kelvin K80: Elmeq.es. 2021. [online] Available at: <<https://www.elmeq.es/ftp/productsFiles/368/KELVIN-K80-K90-ES.pdf>> [Accessed 18 June 2021].
2. Encoder: Infineon.com. 2021. [online] Available at: <https://www.infineon.com/dgdl/Infineon-Encoder_HEDS-5540-A14-AP-v01_00-EN.pdf?fileId=5546d46147a9c2e40147d3d593970357> [Accessed 18 June 2021].
3. Pistón de elevación: Bansbach, “Bansbach Linear actuator system.”
4. Galga SB-FT1: Pcm-uk.com. 2021. [online] Available at: <<https://pcm-uk.com/wp-content/uploads/2020/08/SB-FT1-Data-Sheet.pdf>> [Accessed 18 June 2021].
5. CAN-HAT: Seedstudio.com. 2021. *2-Channel CAN-BUS(FD) Shield for Raspberry Pi*. [online] Available at: <<https://www.seedstudio.com/2-Channel-CAN-BUS-FD-Shield-for-Raspberry-Pi-p-4072.html>> [Accessed 18 June 2021].