UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

Escuela Politécnica Superior de Elche

Departamento de Ciencia de Materiales, Óptica y Tecnología Electrónica

TÉCNICAS AVANZADAS DE INTERACCIÓN MULTIMODAL PARA ENTORNOS DE ROBÓTICA DE REHABILITACIÓN

TESIS DOCTORAL

por

Luis Daniel Lledó Pérez

Ingeniero Técnico de Telecomunicaciones

Abril 2017

UNIVERSITAS
Miguel
Hernández

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

Escuela Politécnica Superior de Elche

Departamento de Ingeniería de Sistemas y Automática

TÉCNICAS AVANZADAS DE INTERACCIÓN MULTIMODAL PARA ENTORNOS DE ROBÓTICA DE REHABILITACIÓN

Autor:

Luis Daniel Lledó Pérez
Ingeniero Técnico de Telecomunicaciones

Directores:

Nicolás M. García Aracil
Doctor Ingeniero Industrial

José María Sabater Navarro
Doctor Ingeniero Industrial

Elche, Abril 2017

UNIVERSITAS
Miguel
Hernández

AUTORIZACIÓN DE PRESENTACIÓN DE TESIS DOCTORAL

Directores: Dr. Nicolás M. García Aracil y Dr. José María Sabater Navarro

Título de la tesis: Técnicas avanzadas de interacción multimodal para entornos de robótica de rehabilitación.

Autor: Luis Daniel Lledó Pérez

Departamento: Ingeniería de Sistemas y Automática
Universidad Miguel Hernández

Los directores de la tesis reseñada certificamos que ha sido realizada bajo nuestra dirección por D. Luis Daniel Lledó Pérez en el Departamento de Ingeniería de Sistemas y Automática de la Universidad Miguel Hernández y autorizamos su presentación de acuerdo con lo dispuesto en el R. D. 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales.

Elche, a viernes, 26 de Mayo de 2017

Firmado: D. Nicolás M. García Aracil

Firmado: D. José María Sabater Navarro

UNIVERSITAS
Miguel
Hernández

DEPARTAMENTO DE CIENCIA DE MATERIALES, ÓPTICA Y TECNOLOGÍA ELECTRÓNICA

Dña. Piedad de Aza Moya, Catedrático de Universidad y Directora del Departamento de Ciencia de Materiales, Óptica y Tecnología Electrónica,

Certifica

Que el trabajo realizado por D. Luis Daniel Lledó Pérez titulado "**Técnicas avanzadas de interacción multimodal para entornos de robótica de rehabilitación.**", ha sido dirigido por el Dr. D. Nicolás M. García Aracil y Dr. D. José María Sabater Navarro y realizado en el Departamento de Ingeniería de Sistemas y Automática, se encuentra en condiciones de ser leído y defendido como Tesis Doctoral ante el correspondiente tribunal en la Universidad Miguel Hernández.

Lo que firmo para los efectos oportunos en Elche a 26 de Mayo de 2017.

Dña. Piedad de Aza Moya
Director Departamento de Ciencia de Materiales, Óptica y Tecnología Electrónica

DEPARTAMENTO DE CIENCIA DE MATERIALES, ÓPTICA Y TECNOLOGÍA ELECTRÓNICA
Campus de Elche. Avda. de la Universidad, s/n – 03202 ELCHE
Tel.: 96 665 84 85

UNIVERSITAS
Miguel
Hernández

Índice general

Índice de figuras

Índice de tablas

UNIVERSITAS
Miguel
Hernández

Agradecimientos

Toda tesis doctoral necesita un gran esfuerzo y dedicación por parte del autor y de los directores, sin embargo, existe un cierto número de personas que acompañan, aconsejan y apoyan a lo largo de muchos años para contribuir en la evolución de todos los aspectos que constituyen a una persona. A todos ellos me gustaría dedicar la finalización de esta tesis.

En primer lugar agradecer a mis directores de tesis por la dedicación y el apoyo que me han brindado durante todas las etapas de esta tesis doctoral. De manera que sin sus conocimientos y experiencia profesional no hubiera podido completar todos los objetivos marcados. También agradecerles el haberme dado la oportunidad de tener una visión más amplia del mundo de la investigación y por la confianza depositada desde el principio para entrar en su grupo de investigación.

A todos los compañeros del grupo de investigación con los que he compartido tantas horas de trabajo y han prestado atención a mis consultas sugiriendo sus valiosas ideas. También, agradecer a las entidades que han financiado la temática de esta tesis doctoral.

En especial a toda mi familia por el cariño, su apoyo incondicional en todas las decisiones que he tomado, entre otras muchas cosas, y en particular a mis padres por su eterna entrega y brindarme tanto el sustento moral como económico para poder estudiar y culminar el objetivo de alcanzar un futuro mejor, porque han sido una indudable referencia y guía durante toda la vida.

A todos mis amigos que han estado siempre en todo momento, cuando se les necesitaba y cuando no también, compartiendo su vitalidad y sentido del humor. Gracias a todos y cada uno de ellos por su cooperación desinteresada.

Pero, sobre todo, gracias a mi pareja por su paciencia, compresión y confianza durante todos estos años que hemos estado juntos, los cuales han sido los mejores.

UNIVERSITAS
Miguel
Hernández

Resumen

La presente tesis presenta la creación de nuevos métodos multimodales de interacción y control de terapias virtuales dentro del campo de la robótica aplicada a la neuro-rehabilitación motora para recuperar la funcionalidad del miembro superior en pacientes con Daño Cerebral Adquirido. Los resultados obtenidos del trabajo realizado durante el desarrollo de esta Tesis Doctoral se recopilan en tres publicaciones en revistas incluidas en el *Journal Citation Reports(JCR)*.

En la primera parte de la tesis, se presenta un sistema de generación de terapias virtuales con cualquier tipo de entorno basado en actividades de la vida diaria o juegos terapéuticos utilizando la Realidad Virtual. Este sistema permite crear una gran variedad de terapias virtuales de manera rápida, económica y sencilla. Además el sistema propuesto en el marco de esta tesis, permite mejorar la motivación y la adherencia al tratamiento por parte del usuario, así como evitar el sentimiento de frustración al adaptar el nivel de dificultad en función de su estado emocional. Una de las características principales del sistema es la capacidad de interacción de las tareas generadas con múltiples dispositivos robóticos basados en una configuración de efector final para controlar el avatar dentro de los entornos virtuales.

En la segunda parte de la tesis, se desarrolla una interfaz multimodal de neuro-rehabilitación asistida por robots que integra la estimación del estado psico-fisiológico del paciente dentro del lazo de control del sistema. Mediante dicho sistema se analizan señales fisiológicas del usuario como son el ritmo cardíaco, la frecuencia respiratoria, la temperatura y la respuesta galvánica de la piel para adaptar automáticamente la realimentación proporcionada por el entorno de Realidad Virtual modificando el nivel de dificultad de los ejercicios generados por el motor de tareas. La clasificación de las señales se ha llevado a cabo con un nuevo algoritmo matemático basado en redes neuronales y la teoría de conjuntos difusos.

Para finalizar se realiza un estudio acerca de cómo influye la visualización de entornos virtuales en dos dimensiones o en tres dimensiones en el rendimiento sensorimotor del miembro superior en pacientes con Daño Cerebral Adquirido durante el desarrollo de terapias de neuro-rehabilitación asistidas por un dispositivo robótico, a partir del análisis estadístico de los movimientos cinématicos registrados por dicho dispositivo. Este estudio se fundamenta en la presentación de dos tareas virtuales con la misma finalidad pero con diferente visualización de gráficos en pantalla, mientras los pacientes dirigen el efector final del dispositivo robótico para cumplir los objetivos de la terapia.

Abstract

This thesis presents the creation of new multimodal methods of interaction and control of virtual therapies in the field of robotic applied to neuro-rehabilitation for recovery the funcionality of the upperlimb of patients with Acquired Brain Injury. The outcomes achieved during the development of this Thesis are collected in three publications included in the *Journal Citation Reports(JCR)*.

A system of virtual therapies generation, with any type of environments based in daily life activities or therapeutic games, is presented in the first part of the thesis. The aim of this system is to create a large variety virtual therapies in a quickly, economic and simple way. In addition, the proposed system improves the patient's motivation and adherence to the treatment. Thus, the frustation feeling is avoided through the adaptation of the difficulty level depending on patient's emotional state. One of the main features of this system is the interaction capacity of the generated tasks with various robotic devices based on an end-effector configuration to control the avatar inside the virtual environments.

In the second part of the thesis, a multimodal interface for neuro-rehabilitation therapies assisted by robotic devices is developed. This interface integrates the estimation of patient's psychophysiological state in the control loop. Through this system, user's physiological signals such as pulse, respiration rate, skin temperature and galvanic skin response are analysed to adapt automatically the feedback provided by the virtual environments. Then, the difficulty level of the tasks generated by the motor task can be modified. The classification of physiological states are carried out by a new mathematical algorithm based in neural networks and the set fuzzy theory.

Finally, a study about how the sensorimotor performance of the upper limb in patients with Acquired Brain Injury is influenced by the virtual environments visualization in two or three dimensions. This study is perfor-

med during neuro-rehabilitation therapies assisted by robotic devices. This robotic device is in charge of recording the kinematic movements to complete a statistical analysis. The presentation of two virtual tasks with the same finality, but with different graphic visualization on a screen while the patient manages the end-effector of the robotic device to achieve the therapy targets, is the methodology of this study.

Parte I

Tesis

UNIVERSITAS
Miguel
Hernández

Capítulo 1

Introducción

UNIVERSITAS
Miguel
Hernández

1.1. Introducción

El denominador común de los pacientes con Daño Cerebral Adquirido (DCA) es el padecimiento de una lesión cerebral súbita cuya afectación principal repercute en el Sistema Nervioso Central (SNC) con diversas secuelas de carácter psíquico, físico y sensorial. Estos efectos producen anomalías en el funcionamiento cognitivo, la comunicación, problemas en el nivel de alerta, alteraciones sensoriales y emocionales o hasta incluso cambios de comportamiento, que condicionan de manera decisiva el desarrollo vital de la persona (Fernández et al., 2009), siempre dependiendo de las áreas específicas del cerebro que se vean afectadas por la lesión. Con respecto a los problemas físicos que puede ocasionar destacan la espasticidad, restricciones de movimientos, déficit en la coordinación y el debilitamiento muscular (Soyuer y Öztürk, 2007). Las lesiones en las regiones frontales y parietales del cerebro provocan parálisis (hemiplejia) o pérdida de fuerza junto con la destreza de media parte del cuerpo con lesiones (hemiparesia). Las causas etiológicas más frecuentes del DCA son los ictus o Accidente Cerebro Vascular (ACV), los Traumatismo Cráneo Encefálico (TCE), tumores cerebrales, infecciones cerebrales y anoxias o hipoxias. Estas patologías producen una serie de inconvenientes, que en la mayoría de los casos reducen la autonomía y la independencia de los sujetos para realizar tareas cotidianas en entornos cercanos.

El DCA es una discapacidad con gran incidencia a nivel nacional e internacional. A nivel mundial, la incidencia estimada en 2007 fue de 101/100.000 (Shiroma et al., 2012), mientras que en Europa la incidencia está estimada en 235 casos por cada 100.000 habitantes (Tagliaferri et al., 2006) durante el periodo de 1995-2006. En España viven 420.064 personas con DCA, donde el 78 % de los casos tuvieron origen por un ACV y el 22 % restante por TCE y otras causas. Sin embargo, en el periodo de 2010 y 2012 se han llegado a dar 104.071 nuevos casos, siendo 99.284 por ACV, 4.937 por TCE y 481 por anoxia. En la Figura 1.1 se muestra un resumen sobre la incidencia y prevalencia del DCA en España. Estos datos se han obtenido del informe *"Las personas con Daño Cerebral Adquirido en España"* publicado en 2015 (FEDACE, 2015) por FEDACE (Ambrona, 1999) con la colaboración del Real Patronato sobre Discapacidad.

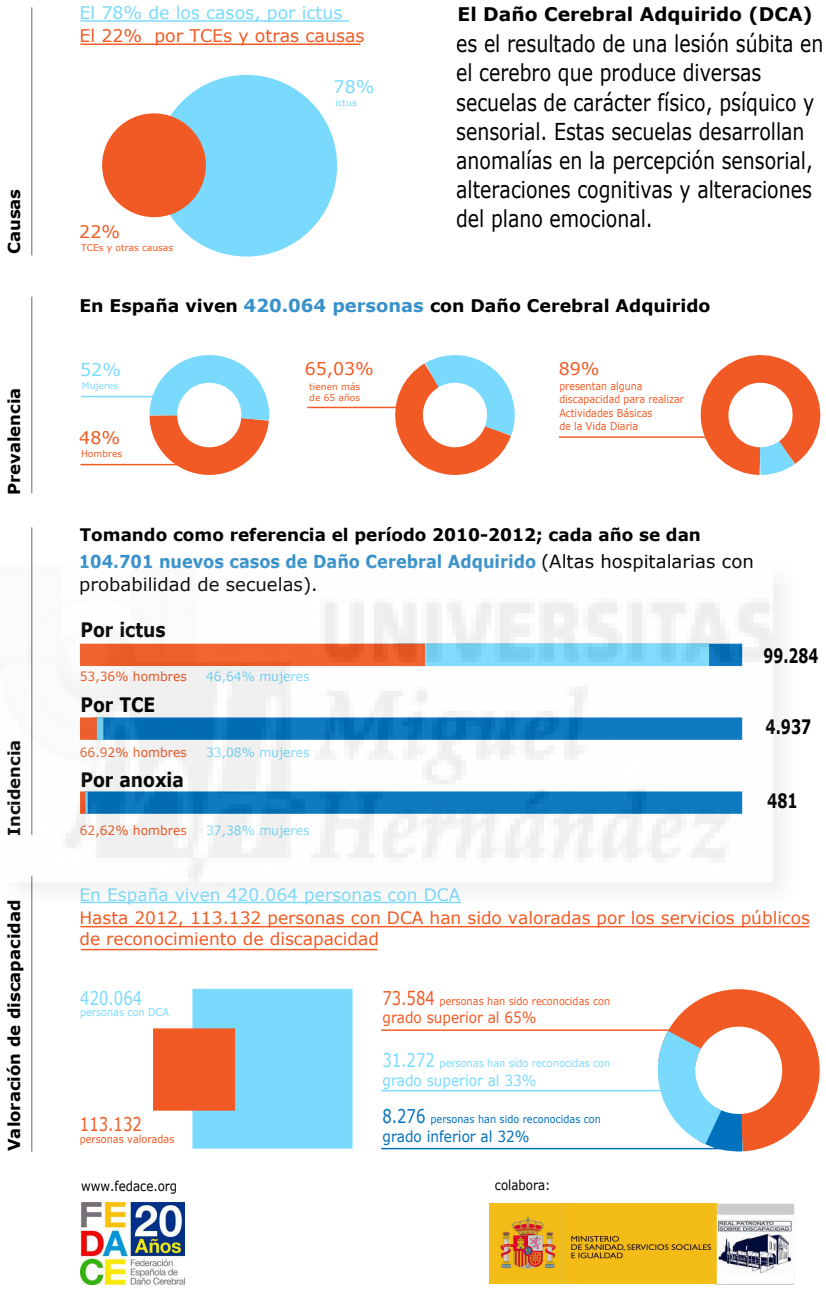


Figura 1.1: Resumen general de la incidencia y prevalencia del Daño Cerebral Adquirido en España (extraída de la página web de la institución FEDACE).

Los datos indican que el ACV es la causa principal de incapacidad permanente y deterioro de la funcionalidad física con perdida de fuerza muscular, espasticidad y falta de coordinación, sobre todo déficits en la función motora de la extremidad superior (Jørgensen et al., 1999). Por estas razones, es necesario la implantación de programas de rehabilitación multidisciplinar para reducir las secuelas provocadas por el ACV. Sin embargo, sólo aproximadamente del 5 % al 20 % recuperan totalmente la funcionalidad perdida (Nakayama et al., 1994), lo que quiere decir que gran parte de los pacientes permanece con una funcionalidad motora limitada después de la rehabilitación. Por lo tanto, aún existe la exigencia de desarrollar nuevas estrategias de entrenamiento que mejoren la rehabilitación de estos pacientes.

En los últimos años se han estado utilizando los dispositivos robóticos como una nueva estrategia de rehabilitación que permite la recuperación de la funcionalidad motora a través de la repetición de ejercicios físicos basados en movimientos voluntarios, los cuales son capaces de producir mejoras después de una lesión traumática, así como promover la plasticidad neuronal. Varios estudios sugieren que la tecnología robótica se puede utilizar para mejorar la calidad y la evaluación en la rehabilitación neurológica (Garcia et al., 2011), mejorando la productividad y reduciendo costes en este campo. Recientemente, en (Norouzi-Gheidari et al., 2012) se ha demostrado que las sesiones de rehabilitación realizadas con dispositivos robóticos obtienen mejores resultados que las terapias convencionales durante la rehabilitación del miembro superior de pacientes con ACV. Por estas razones, la rehabilitación con dispositivos robóticos puede proporcionar una mejoría en la calidad de vida de los pacientes, dándoles más independencia en las actividades de la vida diaria (Pollock et al., 2014).

Por otro lado, la Realidad Virtual (RV) puede proporcionar nuevas características a este tipo de rehabilitación complementando a los sistemas robóticos. La RV es una plataforma tecnológica que permite desarrollar entornos generados por ordenador donde los sujetos pueden explotar e interactuar con cualquier tipo de objetos o eventos para realizar tareas motoras. RV ofrece una forma precisa para controlar todos los elementos de una escena y sus objetivos, ajustando cada tarea a un usuario específico. La principal característica que proporciona la RV es la posibilidad de repetir la misma tarea en cualquier momento modificando factores como el nivel de complejidad, el tiempo y la intensidad de la práctica. De esta manera, la terapia virtual se puede utilizar para promover el aprendizaje y la rehabilitación motora debido a que la RV se puede ajustar para generar un entorno, esce-

nario o actividad que permita al usuario practicar sus habilidades motoras y de esta forma promover la plasticidad neuronal (Doyon y Benali, 2005). La posibilidad de modificar factores como la repetición, la intensidad, el tiempo y la especificidad de las actividades de las terapias virtuales es beneficioso para este tipo de recuperación neuronal (Kleim y Jones, 2008).

Algunos ensayos científicos y clínicos han demostrado la efectividad de la RV como una herramienta de intervención para la rehabilitación de diferentes lesiones con condiciones neurológicas especificas (Burdea, 2002; Crosbie et al., 2007). Sin embargo, se necesita un tipo concreto de dispositivo de control para interacturar con las actividades virtuales en función de la extremidad afectada por la enfermedad. Existe un amplio panorama sobre sistemas de rehabilitación para el miembro superior que utilizan la tecnología robótica incluyendo una visualización de RV (Maciejasz et al., 2014). En algunos estudios, el movimiento repetitivo guiado por dispositivos robóticos y dirigido por RV mejora el control motor en los pacientes con lesiones del miembro superior (Merians et al., 2006). Aparte de esto, existen algunos estudios clínicos sobre el desarrollo de sistemas RV para ofrecer terapias de rehabilitación para la recuperación motora de la función de la mano (Jack et al., 2001) o para mejorar el rendimiento de las actividades de la vida diaria en pacientes posteriores al ictus (Laver et al., 2012; Turolla et al., 2013). Por otra parte, también se ha utilizado la RV para implementar un entorno de navegación en Tres Dimensiones (3D) y explorar la influencia del envejecimiento en la memoria episódica (Jebara et al., 2014). En (Fluet y Deutsch, 2013) se presentó una visión general de múltiples estudios basados en la utilización de RV para la rehabilitación sensoriomotor posterior al ACV donde se ha evaluado una comparación en la eficacia entre la RV y el nivel de atención.

Todos estos estudios apoyan el uso de los dispositivos róboticos vinculados con la RV para la rehabilitación de la funcionalidad sensoriomotriz como una herramienta en el tratamiento de pacientes con ACV, siempre que el enfoque terapéutico esté bien definido para promover la reorganización cortical (Staines et al., 2001). Con este tipo de sistemas, un robot se encarga de guiar al paciente para llevar a cabo un movimiento determinado mientras visualiza un entorno virtual que le motiva a realizarlo, y al mismo tiempo el terapeuta recibe datos que proporcionan información importante referente a la evolución de los pacientes. De esta manera se involucra un poco más a los pacientes para que completen su terapia de rehabilitación intentando reducir los tiempos de recuperación.

1.2. Motivación del trabajo

Cada año aumentan los casos de pacientes hospitalizados por ACV con pérdida súbita de funciones neurológicas y, como previsión, existe un ritmo creciente debido al envejecimiento de la población, ya que según la Organización Mundial de la Salud el número de personas mayores de 65 años aumentará en un 207 % en todo el mundo en los próximos años. De la misma manera que aumenta el numero de casos, también mejora la capacidad de comprensión del funcionamiento del SNC y cómo afecta el DCA, así como el desarrollo de la tecnología, permitiéndonos implementar nuevas estrategias de rehabilitación. Actualmente existen diversas estrategias basadas en dispositivos robóticos vinculados con sistemas RV para realizar ejercicios repetitivos a través de la asistencia funcional de la extremidad superior en términos de amplitud de movimiento, fuerza y coordinación. De esta manera se intenta recuperar la independencia perdida a causa de esta lesión cerebral sobre aspectos tan cotidianos como las actividades de la vida diaria.

Como se ha comentado en el apartado anterior, existen muchos dispositivos robóticos utilizados en la rehabilitación de este tipo de lesiones, lo cuales son beneficiosos desde el punto de vista clínico y motor. En especial los sistemas que se basan tanto en la recuperación de la función motora como en el aprendizaje de tareas que ayudan al paciente a recordar la autonomía que poseía antes del ACV (Mehrholz et al., 2008). Esto se realiza con sistemas de RV donde el paciente es capaz de observar el movimiento de la extremidad para interactuar con elementos de escenarios virtuales que simulan escenas o Actividades de la Vida Diaria (AVD).

Por lo tanto, la motivación fundamental de este trabajo es el desarrollo de un nuevo enfoque de terapia auto-adaptable que favorezca la mejor recuperación de las personas que han sufrido un ACV, esencialmente con la ayuda de robots y vinculando un sistema de visualización con el objetivo de incrementar la motivación y la atención a la hora de realizar un movimiento como indica (Kleim y Jones, 2008). Por otro lado, se intenta reducir el nivel de frustración o aburrimiento, el cual es una de las causas más importantes que dan lugar al abandono de la terapia por parte del paciente, a partir de la adaptación de los niveles de dificultad de las tareas para mantener siempre la misma motivación del paciente dentro de unos límites que eviten la relajación o el estrés durante el desarrollo de la terapia. Además, en esta tesis se busca determinar qué tipo de visualización del entorno virtual es más beneficioso para los pacientes: gráficos en Dos Dimensiones (2D) o 3D.

1.3. Estructura de la tesis

El presente documento recoge todos los aspectos generales para alcanzar el objetivo principal que es desarrollar un sistema de generación de tareas virtuales que permita la implementación de una variedad de ejercicios, con un elevado grado de visualización e interactuación con el entono, en terapias de neuro-rehabilitación asistidas por dispositivos robóticos y aplicando un control bio-cooperativo para obtener una interfaz auto-adaptativa en función del estado psico-fisiológico del usuario. De este trabajo se han presentado tres artículos en revistas incluidas en el *Journal Citation Reports* (JCR). La tesis está dividida en seis capítulos, tres de ellos se centran en los fundamentos de los artículos publicados. También se ha añadido un capítulo de introducción para explicar el estado del arte y un capítulo final donde se comentan las conclusiones alcanzadas. El documento queda estructurado de la siguiente manera:

- **CAPÍTULO 1**: El capítulo inicial enuncia los ámbitos de actuación de la tesis, aportando una introducción al problema que se quiere abordar junto con su posible solución, la motivación y un resumen de todas las contribuciones en revistas donde se han presentado las aportaciones y resultados obtenidos en este trabajo.
- **CAPÍTULO 2**: El segundo capítulo presenta un breve análisis del estado del arte en lo que respecta a las herramientas actuales para el diseño e implementación de sistemas computacionales de visualización basados en RV. Después se realiza una revisión de los sistemas encargados de utilizar RV durante del proceso de rehabilitación motora de personas que han sufrido un ACV, así como los sistemas robóticos para miembro superior más destacados que interactúan con los pacientes utilizando la RV como método de atención y motivación. Para finalizar el capítulo del estado del arte, se presentan algunos sistemas bio-cooperativos que utilizan las señales fisiológicas para estimar el estado psico-fisiológico del paciente y actualizar el nivel de dificultad de los juegos terapéuticos, considerando al paciente dentro del lazo de control del sistema.
- **CAPÍTULO 3**: En este capítulo se describe detalladamente todo el proceso de ingeniería de software sobre el análisis, diseño, implementación y pruebas de un sistema de generación de juegos terapéuticos

formados por tareas que incluyen entornos virtuales de manera realista, para terapias virtuales asistidas por dispositivos robóticos. Este sistema permite la obtención de tareas virtuales de manera rápida y optimizada con módulos software de libre acceso. El principal aporte consiste en la automatización del sistema para organizar y gestionar su contenido así como modificar el nivel de dificultad, y por consiguiente la mayor parte del tiempo de producción se destina al diseño visual de la tarea sin perder tiempo en la parte de programación de los elementos y de los objetivos que se deben cumplir para completar su finalidad. Al final del capítulo se presenta una serie de tareas implementadas con este motor de tareas que se han utilizado en terapias seguidas por pacientes post-ictus en un hospital de larga estancia, siendo asistidos por dos tipos de dispositivos robóticos. También se definen los sistemas de interacción que son compatibles con este software.

- **CAPÍTULO 4**: Dentro de este capítulo se define un sistema de control bio-cooperativo mostrando el proceso de adquisición y procesamiento de las señales fisiológicas para realizar una estimación del estado psico-fisiológico de los paciente utilizando un nuevo método de clasificación que combina conceptos de la lógica difusa y las redes neuronales para aprovecharse de las ventajas de ambas técnicas. Con la clasificación de estas señales se generan comandos que permiten la actualización de la complejidad de una tarea implementada con el sistema explicado a lo largo del capítulo 3. De esta manera la terapia se adapta al paciente, intentando maximizar su participación y su motivación dentro de las terapias virtuales.

- **CAPÍTULO 5**: En este capítulo se detalla un estudio acerca de la influencia de la visualización de tareas virtuales, mostradas en dos y tres dimensiones, en la función sensoriomotriz de pacientes con ACV durante la realización de una terapia virtual de neuro-rehabilitación. Con este estudio se pretende definir si existen diferencias en el movimiento cinemático del miembro superior utilizando un dispositivo robótico como herramienta de evaluación mientras se visualiza el mismo entorno virtual con diferentes gráficos visuales.

- **CAPÍTULO 6**: En el último capítulo se exponen las conclusiones derivadas de la realización de la tesis y una serie de trabajos futuros que permitirían ampliar los puntos de la investigación presentada.

1.4. Publicaciones

El desarrollo de esta tesis doctoral ha dado lugar a las siguientes publicaciones en revistas impactadas:

1. Luis D. Lledó, Francisco J. Badesa, Miguel Almomacid, José M. Cano-Izquierdo, José M. Sabater-Navarro, Eduardo Fernández, Nicolás García-Aracil, "Supervised and Dynamic Neuro-Fuzzy Systems to Classify Physiological Responses in Robot-Assisted Neurorehabilitation", PLoS ONE, vol.10, no.5, pp.1-16, 2015. ISSN: 1932-6203.
   Factor de Impacto (JCR) = 3.057, Grupo: A, primer quintil (11/63) en la categoría MULTIDISCIPLINARY SCIENCES

2. Luis D. Lledó, Jorge A. Díez, Arturo Bertomeu-Motos, Santiago Ezquerro, Francisco J. Badesa, José M. Sabater-Navarro, Nicolás García-Aracil, "A Comparative Analysis of 2D and 3D Tasks for Virtual Reality Therapies Based on Robotic-Assisted Neurorehabilitation for Post-Stroke Patients", Frontiers in Aging Neuroscience, vol.8, pp.205, 2016. ISSN: 1663-4365.
   Factor de Impacto (JCR) = 4.348, Grupo: A, primer quintil (6/49) en la categoría GERIATRICS & GERONTOLOGY

3. Luis D. Lledó, Arturo Bertomeu-Motos, José M. Catalán, Jorge A. Díez, Francisco J. Badesa, José M. Sabater-Navarro, Nicolás García-Aracil, "A software platform for virtual tasks implementation with adaptable difficulty level in robot-assisted neurorehabilitation therapies", Software - Practice and Experience. In review. ISSN: 0038-0644.
   Factor de Impacto (JCR) = 0.652, Grupo: C, tercer quintil (76/106) en la categoría COMPUTER SCIENCE, SOFTWARE ENGINEERING

Capítulo 2

Estado del arte

En este capítulo se expone una revisión del estado del arte de las técnicas y herramientas utilizadas en la simulación 3D para entornos de robótica de rehabilitación basados en realidad virtual. Es importante adquirir una visión general de las principales herramientas de simulación para conocer mejor el funcionamiento de los entornos aplicados en terapias de rehabilitación. Posteriormente se analizan brevemente los sistemas de rehabilitación virtual más utilizados en la actualidad para la recuperación funcional de miembros superiores e inferiores a través de la interacción hombre-sistema. Después, se describen algunos de los sistemas robóticos utilizados en la rehabilitación del miembro superior más destacados. Por último, se presentan los nuevos sistemas de rehabilitación biocooperativos capaces de actualizar la intensidad del tratamiento a partir del estado psico-fisiológico de la persona determinado por sus constantes fisiológicas.

UNIVERSITAS
Miguel
Hernández

2.1. Diseño de entornos virtuales

El concepto más importante en el diseño de los entornos de RV es la generación de gráficos por ordenador. Los gráficos se encargan de replicar toda la información que el ser humano puede recibir por la vista durante pequeños periodos de tiempo. La gestión de estos elementos se lleva a cabo por un ordenador, el cual los produce a partir de modelos matemáticos y los plasma en una proyección con cierto nivel de detalle con la posibilidad de simular todo tipo de entornos para cualquier aplicación.

Las principales características técnicas (Alexander et al., 2005) de esta tecnología son: la fidelidad audiovisual y funcional, que permite a los pacientes tener una mejor motivación de participación (Witmer y Singer, 1998) sobre el escenario virtual debido al grado de similitud gráfica-sonora y al comportamiento realista de los elementos; la accesibilidad para usuarios con todo tipo de experiencia técnica; la heterogeneidad ante la posibilidad de distribución de software y hardware a múltiples plataformas; la conectividad ante todo tipo de soportes para captar el mayor número posible de usuarios; y la componibilidad para generar entornos virtuales con el mínimo gasto de desarrollo posible.

Todos estos criterios sobre la RV han desencadenado una gran cantidad de inversión e investigaciones en los últimos años. Existe un elevado número de empresas dedicadas a los diferentes sectores de la realidad virtual que proporcionan aplicaciones, generación de contenidos, herramientas de desarrollo, plataformas de distribución, hardware de captura de movimientos, de visualización y de entrada para el usuario. La elección de las herramientas software es una medida importante para completar los objetivos de diseño e implementación de una manera eficaz en términos de tiempo, inversión y flexibilidad. Conocer las funcionalidades de estas herramientas es fundamental para la elaboración de aplicaciones RV confiables y robustas. En la Figura 2.1 se muestra un esquema con las herramientas más utilizadas.

Por lo tanto, existe gran cantidad de herramientas para desarrollar entornos virtuales con fines de enseñanza (Fiolhais y Trindade, 1999), entretenimiento, práctica, rehabilitación, telecomunicaciones, preparación de intervenciones quirúrgicas, visualización científica (Tamura et al., 2001), comercialización, fabricación, etc. Para facilitar el desarrollo de aplicaciones

Figura 2.1: Herramientas para el desarrollo de entornos virtuales.

interactivas de simulación virtual, el elemento más importante es el motor de juego ya que engloba todos los demás componentes modulares constituidos por el framework de renderizado, el framework de simulación física, las librerías gráficas, librerías de audio e incluso incorpora herramientas de modelados de gráficos visuales. Sin embargo, existe la posibilidad de seleccionar los módulos de manera independiente para desarrollar versiones más reducidas, en términos de recursos software y costes, de este tipo de aplicaciones de entornos virtuales. Los motores de renderizado añaden la funcionalidad de visualización en tres dimensiones de los modelos diseñados con los programas de creación de contenido, mientras que los motores de simulación física calculan comportamientos dinámicos en tiempo real a través de algoritmos de detección de colisiones. Estos dos elementos son la base de este tipo de aplicaciones, los cuales pueden ser complementados con interfaces gráficas de usuario y mediante la incorporación de sonidos.

2.2. Sistemas de rehabilitación virtual

La rehabilitación virtual mezcla el concepto de terapia tradicional con la tecnología interactiva que sustituye la percepción de la realidad por estímulos generados por ordenador con el objetivo de recuperar las funciones motoras perdidas. Esta combinación permite obtener un tratamiento basado en ejercicios de simulación con cualquier tipo de entorno para el entrenamiento motor o cognitivo. Un aspecto destacable de la RV en la rehabilitación virtual es que proporciona un control total de todos los aspectos de los ejercicios como pueden ser el número de repeticiones, la intensidad, o el aspecto de los objetos del entorno. Por otro lado, la utilización de dispositivos externos contribuye a conseguir un tratamiento más controlado, ya que pueden recoger datos objetivos en tiempo real. De esta manera, se pueden implementar nuevos sistemas de asistencia que son capaces de medir información sobre el proceso de recuperación del paciente para ofrecer al terapeuta una mejor gestión del tratamiento y optimizar los ejercicios. Por consiguiente, la utilización de los sistemas de rehabilitación virtual permiten llevar a cabo terapias que:

- Mejoran la adaptación de la tarea y el nivel de dificultad para cada paciente.
- Mejoran la participación física y psicológica de los pacientes.
- Incrementan el cumplimiento y el compromiso del paciente a corto y largo plazo.
- Aumentan la la motivación debido a que este factor es clave en el proceso de reorganización neuronal necesario para re-adquirir las capacidades motoras perdidas a raíz del accidente.

En (Kim et al., 2005), (Kleim y Jones, 2008) y (Saposnik et al., 2011) corroboran que la atención y la motivación son esenciales, indicando que los principios para una rehabilitación motora se basan en que los entornos virtuales simulen el mundo real con comportamientos realistas y funcionales. En definitiva, la RV permite la simulación de entornos con aspectos de la vida cotidiana orientado a realizar tareas diarias (Koenig et al., 2009).

2.2.1. Sistemas pasivos de rehabilitación virtual

Los sistemas de RV empleados en la neuro-rehabilitación se pueden clasificar en dos grupos dependiendo del tipo de interacción sistema-usuario y la búsqueda de diferentes objetivos en el tratamiento. El primer grupo se basa en el registro de gestos realizados por el usuario donde el sistema de control se encarga de analizar y procesar los movimientos para evaluar la evolución del paciente durante las sesiones de rehabilitación. Este grupo está formado por sistemas de captura de movimiento utilizando dispositivos de bajo coste (Figura 2.2), como el control remoto WiiMote de la consola Wii de Nintendo que registra cambios en la aceleración y la orientación para trasladar los movimientos del jugador a la pantalla o el sistema Wii Balance Board que es una tabla capaz de obtener la presión ejercida sobre ella, el periférico EyeToy y el mando Move de Sony PlayStation, o el sistema Kinect de la consola Xbox de Microsoft para capturar los movimientos del usuario a través de una cámara de vídeo con sensor de profundidad. También existen dispositivos como Leap Motion el cual detecta el movimiento natural de las manos.

Figura 2.2: Dispositivos de bajo coste para la captura de movimientos.

En la Tabla 2.1 se recogen algunos de los sistemas de rehabilitación virtual para diferentes partes del cuerpo más utilizados dentro de este grupo de interacción, añadiendo una pequeña descripción del sistema.

La mayor parte de los sistema mencionados en la tabla se basan en tecnología de control por gestos a partir de una cámara de profundidad para

Empresa	País	Producto	Descripción
GestureTek Health www.gesturetekhealth.com	Canadá	IREX Multi-Sensory Corner	Rehabilitación funcional de todo el cuerpo. Realiza un control con las manos o con otras partes del cuerpo.
Accelerated Care Plus www.acplus.com	USA	OmniVR	Diseñado para pacientes geriátricos.
LiteGait www.litegait.com	USA	ViTiUp	Intervenciones terapéuticas para la parte superior del cuerpo.
Brontes Processing www.virtual-reality-rehabilitation.com	Polonia	SeeMe	Permite la práctica de ejercicios de coordinación, equilibrio, tiempos de reacción y memoria.
Mira www.mirarehab.com	Reino Unido	MIRA	Sistema destinado a niños.
VirtualWare www.virtualwaregroup.com	España	VirtualRehab Body VirtualRehab Hands	Ejercicios destinados a rehabilitar trastornos de movimiento, postura y déficit motor. Diseñado para realizar movimientos de pulgar, dedos y mano.
BTS www.btsbioengineering.com	Italia	BTS NIRVANA	Incentiva el alcance de objetivos.
LabHuman www.labhuman.com	España	Umbrella BioTrak	Rehabilitación virtual para el brazo. Proporciona entornos de juego con escalas clínicas.

Tabla 2.1: Sistemas comerciales de rehabilitación virtual con captura de movimiento.

integrar la imagen de los pacientes dentro de un entorno virtual, donde se pueden visualizar interactuando en tiempo real con objetos o controlar ciertos avatares que reaccionan al movimiento del paciente. De esta manera se incrementa la sensación de presencia dentro de la terapia utilizando una realimentación visual y auditiva sin limitación de movimiento debido a que no se necesita utilizar otros dispositivos de control, a excepción de elementos móviles de registro de orientaciones y aceleraciones en algunos casos. Estos sistemas incorporan ejercicios para el entrenamiento y la rehabilitación de determinadas funciones. Permiten la ejecución de diferentes tipos de ejercicios virtuales a través de la monitorización en pantalla y la posibilidad de gestionar el progreso de rehabilitación de los pacientes modificando las repeticiones de los ejercicios. Este tipo de sistemas necesitan la asistencia de un terapeuta para evaluar las habilidades del paciente.

De los sistemas anteriores destaca IREX. Este sistema se ha utilizado en varios ensayos clínicos para comprobar la rehabilitación de pacientes con este tipo de técnicas de interacción, dando resultados favorables (Jang et al., 2005; Kim et al., 2011; Yang et al., 2014). Permite diseñar programas de ejercicios interactivos para articulaciones individuales, movimientos com-

binados o funcionales de todo el cuerpo, dando al terapeuta una medida del rango de movimiento, la frecuencia de los ejercicios y las repeticiones completadas. Los beneficios proporcionados a los pacientes son la mejora del funcionamiento ejecutivo, mejora de la reorganización cortical, mejora del equilibrio y el control de tronco, mejora de la movilidad, aumento de la sensación de control, el rendimiento y la independencia, mejora de la socialización y el aumento del deseo de participar. En la Figura 2.3 se puede observar un usuario utilizando el sistema.

Figura 2.3: Sistema IREX (extraída de la página web de la empresa GestureTek Health).

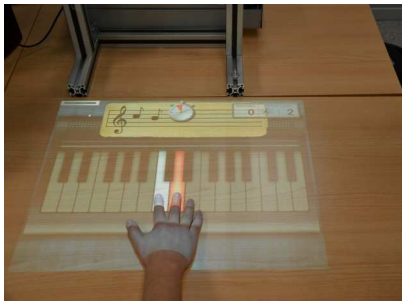
En el ámbito español cabe destacar el producto BioTrak, comercializado por Bienetec e ideado por LabHuman, por su variedad en los tipos de rehabilitación que puede llevar a cabo como el equilibrio en bipedestación y en sedentación, la memoria y atención, la anosognosia. Además permite realizar tareas de la vida diaria, e incluso la rehabilitación musculo-esquelética del hombro. Es un sistema formado por una cámara de profundidad, una plataforma de presión, una tablet multitáctil y una mesa multitáctil para el trabajo colaborativo en clínica. Otro factor a tener en cuenta, es el módulo de gestión orientado a la tele-rehabilitación el cual permite estudiar

la evolución de los pacientes, asignarles tareas o informar de hitos desde zonas distantes a través de Internet. En (Lloréns et al., 2011) se presenta un estudio clínico con dicho sistema con resultado satisfactorio. En la Figura 2.4 se muestra una sesión de rehabilitación con el sistema en la casa de un paciente.

Figura 2.4: Sistema BioTrak (extraída de la página web de Biotraksuite).

A parte de los sistemas anteriores, en la Figura 2.5 se recogen imágenes de gran parte de los sistema de rehabilitación virtual expuestos en la Tabla 2.1. La mayoría de estos sistemas comerciales de rehabilitación virtual se basan en el registro de movimientos mediante cámaras de visión y profundidad (Weiss et al., 2009), y el usuario no necesita mecanismos a parte para la captura de datos cinenámitos. Sin embargo, el terapeuta debe presenciar constantemente la ejecución de los movimientos del paciente y apoyarlo para realizar las trayectorias de referencia predeterminadas. En algunas ocasiones, no es posible para el paciente llevar a cabo la terapia virtual debido a sus capacidades físicas residuales.

En la literatura científica existe una amplia variedad de estudios sobre la utilización de la RV en la rehabilitación motora después de un ACV (Saposnik et al., 2011; Sveistrup, 2004; Laver et al., 2012; Holden, 2005).



Umbrella

VirtualRehab Body

OmniVR

Multi-Sensory Corner

SeeMe

BTS Nirvana

Figura 2.5: Sistemas de rehabilitación virtual expuestos (extraídos de MedicalEXPO).

En estos estudios se han podido comprobar resultados significativos en la recuperación motora, así como en la reorganización cortical (Jang et al., 2005).

Aparte de la utilización de cámaras para el registro del movimiento, existen algunos sistemas que necesitan la incorporación de elementos conectados al usuario. Por ejemplo, en (Cameirão et al., 2010) se ha desarrollado un sistema RV llamado *Rehabilitation Gaming System* (RGS), el

cual está orientado a la rehabilitación del déficit motor de las extremidades superiores en pacientes con ACV mediante la utilización de guantes. Los entornos desplegados generan un entrenamiento individualizado a través de la combinación en la ejecución de movimientos con la observación de una acción conectada con unas extremidades virtuales mostradas con perspectiva en primera persona. En la Figura 2.6 se muestra un paciente utilizando el sistema. Además se ha investigado el impacto clínico del sistema RGS sobre los tiempos de recuperación de ACVs agudos, obteniendo resultados positivos (da Silva Cameirão et al., 2011).

Figura 2.6: Sistema RGS (Cameirão et al., 2010).

Con respecto a la rehabilitación de la marcha, también existen dispositivos destinados a este fin. En (Deutsch et al., 2001) se ha desarrollado un sistema llamado *Rutgers Ankle Rehabilitation System* (RARS) que permite la rehabilitación de la extremidad inferior navegando por un entorno virtual mostrado en una pantalla. Consiste en unas plataformas cuya estructura se acopla a pies y tobillos registrando la fuerzas generadas por el movimiento para interactuar con el ejercicio virtual (Figura 2.7). También, se han realizado diversos estudios para validar este sistema (Boian et al., 2003; Mirelman et al., 2010).

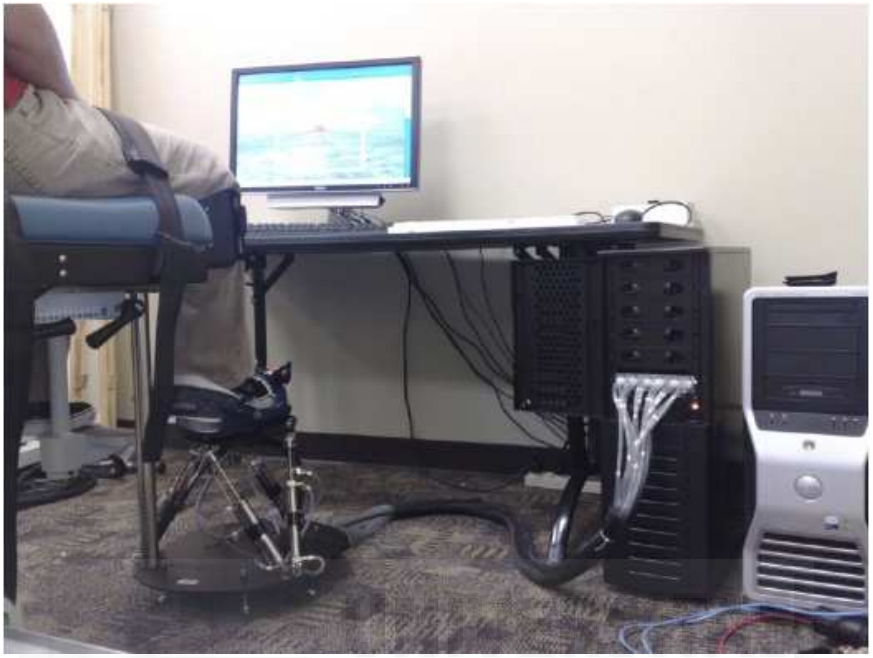


Figura 2.7: Sistema RARS (extraída de la página web del proyecto Rutgers).

2.2.2. Dispositivos robóticos para la rehabilitación del miembro superior

Dentro del proceso de recuperación de los pacientes con DCA, uno de los principales factores es la repetición de los ejercicios y trayectorias de manera correcta. Para dirigir el tratamiento del paciente se han diseñado dispositivos robóticos como herramienta de rehabilitación para asistir al movimiento durante el tratamiento con el objetivo de realizar trayectorias más precisas y consistentes donde se involucra un gran número de movimientos repetitivos. Con este tipo de dispositivos es más fácil controlar el nivel de progreso en la recuperación proporcionando al terapeuta medios para optimizar los tratamientos. Muchos de estos dispositivos robóticos ya se están utilizando en la práctica clínica, así como en la evaluación clínica (Dukelow, 2011; Maciejasz et al., 2014), debido a que muchas investigaciones han demostrado el potencial de los robots para mejorar la terapia de neuro-rehabilitación (Lum et al., 2002; Fasoli et al., 2003).

Desde los años 90 hasta la actualidad se han ido diseñando este tipo de sistemas de rehabilitación motora focalizados en las extremidades superio-

res, extremidades inferiores, en control del equilibrio y la postura e incluso para la deambulación. Sin embargo, en esta sección se van a describir solamente los dispositivos destinados al miembro superior, que es el punto donde se centra esta tesis. Todos estos sistemas se pueden diferenciar por las características técnicas, tipo de asistencia, la forma de dirigir las trayectorias, la manera de movilizar la extremidad y otras muchas particularidades. En lo que respecta a la manera de movilizar las extremidades se pueden clasificar en dos arquitecturas bien definidas (Dario et al., 2003):

- Arquitectura de tipo efector final: En este tipo de configuración sólo existe un punto de contacto entre el usuario y el dispositivo, donde el efector final del robot es equivalente al efector natural de la extremidad humana.

- Arquitectura de tipo exoesqueleto: Este tipo corresponde a una estructura mecánica fijada al miembro del usuario donde las articulaciones y los segmentos del robot coinciden con los del cuerpo humano.

En la Tabla 2.2 se muestra un resumen de los múltiples sistemas robóticos que se han implementado, mediante los cuales se han realizado estudios clínicos con pacientes. Entre todos ellos cabe destacar los sistemas implementados por el Instituto Tecnológico de Massachusets que han proporcionado el diseño de algunos dispositivos comerciales (Figura 2.8) siguiendo conceptos de modularidad y reconfigurabilidad como son: Inmotion ARM (basado en el sistema MITManus (Krebs et al., 2004)) e InMotion WRIST (Krebs et al., 2007) para la pronación, supinación, flexión, extensión, aducción y abducción del antebrazo, junto con el módulo Inmotion Hand para el entrenamiento de movimientos de pinza y liberación de la mano.

Otros sistemas a destacar debido los buenos resultados clínicos son:

- iPAM (Jackson et al., 2007b): Este dispositivo desarrollado por la Universidad de Leeds está diseñado para terapias de rehabilitación en posición de sedentación (Figura 2.9). Emplea dos brazos robóticos para asistir al brazo y a la muñeca del paciente, y posee un subsistema de RV para visualizar tareas. En (Jackson et al., 2007a) se muestran los resultados clínicos con este tipo de configuración.

- GENTLE/S (Amirabdollahian et al., 2007): Este sistema implementado por la Universidad de Reading permite completar movimientos

Articulac.	Sistema	Institución	Tipo
Codo	MARIONET	Instituto de Rehabilitación de Chicago	Efector final
	Myomo e100	Myomo Inc.	Exoesqueleto
	MEM-MRB	Universidad de Osaka	Efector final
Muñeca	ASSIST	Universidad de Okayama	Exoesqueleto
	PolyJbot	Universidad de Hong Kong	Efector final
Dedos	Amadeo	Tyromotion	Efector final
	Hand of Hope	Rehab-Robotics Comp.	Exoesqueleto
	HEXORR	CABRR	Efector final
	HandCARE	Universidad de Singapur e Imperial College de Londres	Efector final
	InMotion HAND	Instituto Tecnológico de Massachusets	Efector final
Hombro y codo	ARM Guide	Instituto de rehabilitación de Chicago y Universidad de California	Efector final
	REHAROB	Universidad de Budapest	Efector final
	ARMIN	UETH Zurich	Exoesqueleto
	PUPARM	Universidad Miguel Hernández	Efector final
	NeReBot	Universidad de Padua	Efector final
	Limpact	Universidad de Twente	Exoesqueleto
	BONES	Universidad de California	Exoesqueleto
	InMotion ARM	Instituto Tecnológico de Massachusets	Efector final
Antebrazo y muñeca	InMotion WRIST	Instituto Tecnológico de Massachusets	Exoesqueleto
	Supinator extender	Universidad de California	Exoesqueleto
Muñeca y dedos	HWARD	Universidad de California	Exoesqueleto
	AMES	Universidad de Portland	Exoesqueleto
Hombro, codo y antebrazo	GENTLE/S	Universidad de Reading	Efector final
	iPAM	Universidad de Leeds	Efector final
	NJIT-RAVR	Instituto Tecnológico de New Jersey	Efector final
	RehabExos	Universidad de Pisa	Exoesqueleto
	L-Exos	Universidad de Pisa	Exoesqueleto
	ADLER	Instituto de Rehabilitación de Chicago	Efector final
Hombro, codo y dedos	PneuWrex	Universidad de California	Exoesqueleto
	T-WREX	Universidad de California	Exoesqueleto
Codo, antebrazo y muñeca	MAHI	Universidad de Houston	Exoesqueleto
	WOTAS	Consejo Superior de Investigaciones Científicas	Exoesqueleto
Hombro, codo, antebrazo y muñeca	Hybrid-PLEMO	Universidad de Osaka	Efector final
	CADEN-7	Universidad de Washington	Exoesqueleto
	MIME-RiceWrist	Universidad de Houston	Exoesqueleto
	RUPERT	Universidad de Arizona	Exoesqueleto
	AUPA	Universidad Miguel Hernández	Efector final
	UHD	Instituto de rehabilitación de Eslovenia	Efector final
Hombro, codo, antebrazo, muñeca y dedos	ArmeoSpring	Hocoma	Exoesqueleto
	GENTLE/G	Universidad de Reading	Exoesqueleto
	HEnRIE	Universidad de Liubliana	Exoesqueleto
	MUNDUS	Politécnica de Milán	Exoesqueleto

Tabla 2.2: Sistemas de rehabilitación robótica del miembro superior (Maciejasz et al., 2014).

Figura 2.8: Sistemas InMotion (extraídas de la página web de Interactive Motion).

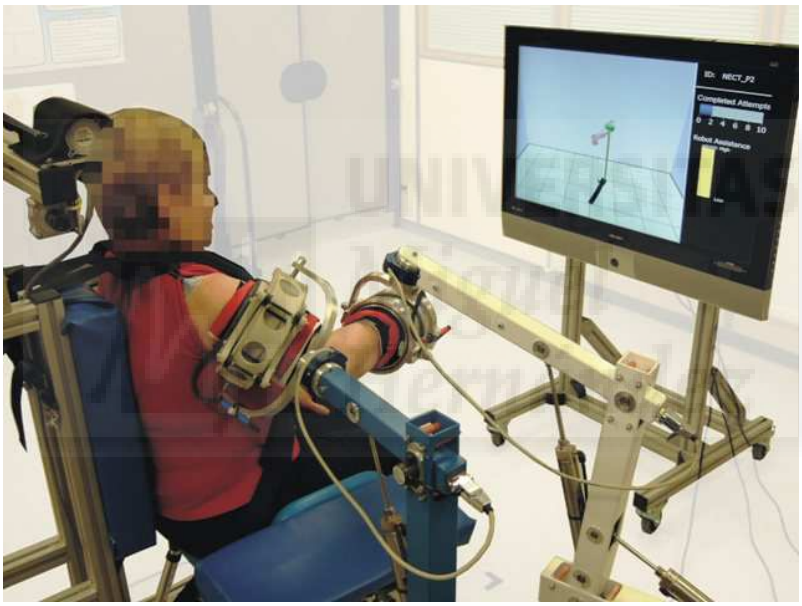


Figura 2.9: Sistema iPAM (Culmer et al., 2010).

de alcance y posiciones arbitrarias de la mano a través de dos puntos de fijación: codo y antebrazo. Dicho sistema se complementa con un exoesqueleto de mano para realizar movimientos de agarre. El sistema completo se llama GENTLE/G (Loureiro y Harwin, 2007). Gracias a la utilización del dispositivo robótico HapticMaster (Van der Linde et al., 2002) se obtiene un lazo de renderización háptico para una realimentación táctil entre el efector final y los elementos del escenario virtual. En la Figura 2.10 se muestra la estructura del sistema

GENTLE/S y el módulo de mano para convertirlo en GENTLE/G.

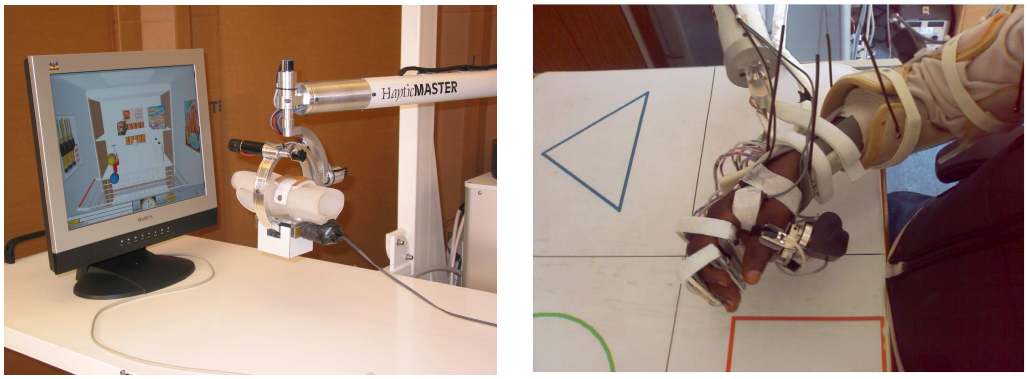


Figura 2.10: En la izquierda, el sistema GENTLE/S (extraída de la página web del proyecto GENTLE). En la derecha, el módulo de mano (Loureiro et al., 2009).

2.3. Sistemas auto-adaptativos de rehabilitación virtual

Con la intención de mejorar las técnicas de rehabilitación robótica, en los últimos años se han realizado nuevos enfoques en lo que respecta al paciente para atribuirle nuevas competencias bio-coperativas dentro de su proceso de rehabilitación (Mihelj et al., 2011; Koenig et al., 2011). De esta manera, el paciente ya no se considera sólo como el receptor pasivo de las trayectorias generadas por los dispositivos robóticos, sino que es el integrante encargado de cerrar el lazo de control del sistema aportando información fisiológica sobre su estado actual. Este hecho permite a los mecanismos de control bio-cooperativo la adaptación del nivel de intensidad de la terapia y maximizar la participación del paciente (Novak et al., 2011). El análisis de este tipo de señales puede permitir a los profesionales clínicos tomar decisiones que permitan mejorar la eficiencia y la efectividad de la rehabilitación motora en los pacientes. En (Simonetti et al., 2016) se proporciona un resumen de sistemas bio-cooperativos en terapias de neuro-rehabilitación que utilizan interfaces adaptativas multimodales.

Aunque actualmente no existen muchos sistemas de rehabilitación asistidos por robots con la integración de este nuevo enfoque de control, hay que destacar los resultados obtenidos en el proyecto MIMICS (*Multimodal*

Immersive Motion rehabilitation with Interactive Cognitive Systems) (Munih et al., 2009). El proyecto se centra en mejorar la rehabilitación sensorial y motora mediante el incremento del compromiso por parte del usuario utilizando sistemas cognitivos y visualizaciones multimodales de entornos virtuales inmersivos. Así, el compromiso del paciente puede aumentar, llevando a un entrenamiento mas intensivo y a obtener mejores resultados terapéuticos. Para cumplir este objetivo, el sistema se encarga de adquirir en tiempo real datos fisiológicos (movimiento, fuerzas, voz, actividad muscular, conductancia de la piel, frecuencia cardíaca,...) y estimar el estado del paciente, en general, la condición psico-fisiológica para adaptar dinámicamente el sistema de RV. Esta información fisiológica se combina con el sistema RV inmersivo de gráficos y sonido en 3D para aumentar el realismo y la motivación del paciente en el entrenamiento de rehabilitación, así como controlar la presencia y la atención dentro de este entorno. El sistema MIMICS tiene dos posibles configuraciones hardware dependiendo del miembro a rehabilitar. Para el miembro inferior se utiliza el dispositivo robótico de rehabilitación LOKOMAT (Jezernik et al., 2003; Riener, 2012), mientras que para el miembro superior se utiliza HapticMaster (Van der Linde et al., 2002). En la Figura 2.11 se muestran las dos posibles configuraciones de hardware robótico que comparten el sistema de recogida de señales fisiológicas.

Figura 2.11: Configuración del sistema para rehabilitación de miembros inferiores y superiores (extraídas de la página web del proyecto MIMICS).

En España e Italia se han llevado a cabo otros estudios clínicos preliminares en centros de medicina y rehabilitación para validar el sistema MAAT (Zollo et al., 2011b). En este proyecto se ha desarrollado un sistema robótico

(Zollo et al., 2011a) para la administración temprana de terapias de rehabilitación en pacientes que han sufrido un ACV con el fin de maximizar la motivación y la participación del paciente en la terapia. Permite la evaluación continua del progreso de recuperación desde un punto de vista funcional y neurológico, con especial atención sobre la cuestión de la seguridad en la interacción humano-robot. Para ello, se registran datos multi-sensoriales (Badesa, 2014) como las fuerzas, el movimiento y las señales fisiológicas para adaptar la complejidad de la terapia y mostrar en tiempo real un entorno de realidad virtual inmersivo conforme a los requisitos específicos del paciente. Por lo tanto, el control bio-cooperativo se encarga de actualizar la terapia en función de las necesidades del paciente gracias a la información sobre su estado anímico. En la Figura 2.12 se muestra el sistema robótico con las capacidades multimodales de detección de las señales fisiológicas.

Figura 2.12: Sistema MAAT (extraída de la página web del ECHORD).

El Grupo de Neuroingeniería Biomédica en la Universidad Miguel Hernández de Elche ha implementado un control bio-cooperativo para terapias asistidas por dispositivos robóticos con el objetivo de adaptar de forma automática el nivel de dificultad de las tareas a partir de la información de las señales fisiológicas (Badesa et al., 2014c). En la Figura 2.13 se muestra el diagrama del control bio-cooperativo donde se enlaza el dispositivo robótico, el software de rehabilitación y realidad virtual, el sistema de procesamiento

y extracción de las características de las señales fisiológicas y el método de clasificación para estimar el estado emocional del usuario.

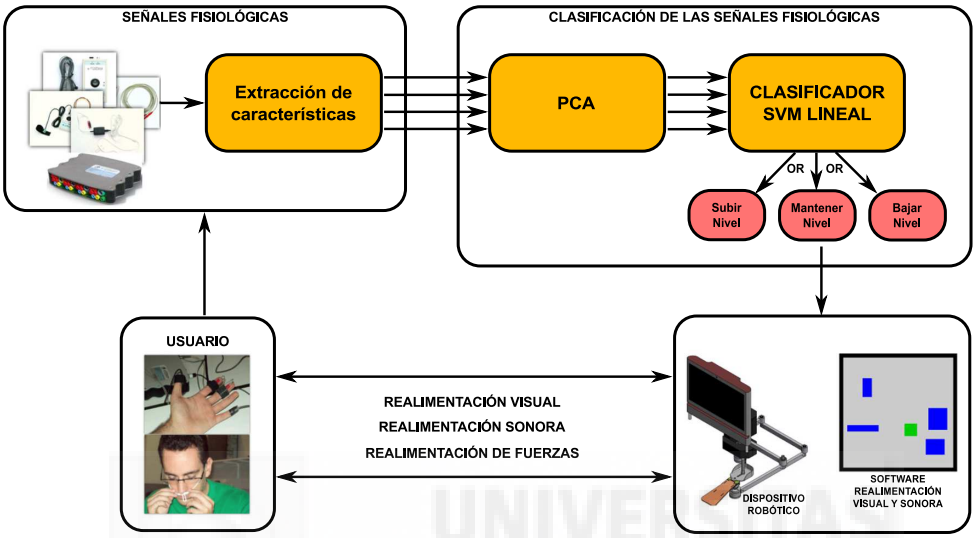


Figura 2.13: Diagrama del control bio-cooperativo.

También cabe destacar el nuevo sistema robótico de agarre terminal desarrollado por el equipo de Ingeniería Biomédica de la Fundación CARTIF llamado PHYSIOBOT (Guerrero et al., 2013; Fraile Marinero et al., 2013) para entrenar movimientos de miembro superior en pacientes con ACV a partir de terapias adaptadas a las necesidades de cada paciente (paradigma *assist as needed*). Permite administrar terapias orientadas a tareas y además dispone de un conjunto de sensores que registra la actividad del paciente en cada momento.

UNIVERSITAS
Miguel
Hernández

Capítulo 3

Diseño de un motor de tareas para neuro-rehabilitación asistidas por dispositivos robóticos

En este capítulo se presenta toda la metodología utilizada para el análisis, diseño, implementación y documentación del motor de tareas propuesto para neuro-rehabilitación asistida por dispositivos robóticos, con el objetivo de obtener una herramienta que proporciona la generación de tareas virtuales de manera rápida y enfocadas a la rehabilitación de miembro superior de personas con daño cerebral adquirido.

En la primera parte del capítulo se describen los componentes principales del sistema software. Posteriormente, se realiza la especificación de los requerimientos y los requisitos del sistema, seguido del proceso de análisis con los casos de uso, la estructuración de la arquitectura, una explicación de la imple-

mentación del código fuente, así como la creación de una serie de tareas para realizar una evaluación de integración.

Para finalizar, se va a realizar una experimentación de algunas de las tareas virtuales implementadas junto con diferentes dispositivos robóticos para comprobar el funcionamiento de la interacción robot-usuario en tiempo real.

3.1. Introducción

Durante la época de los 90 surgió un nuevo concepto de modulación de software llamado motor de juegos (Gregory, 2009) para el desarrollo aplicaciones interactivas en tiempo real, definiendo rutinas de programación que permiten su diseño, creación y visualización. De esta manera, la arquitectura quedaba divida en módulos fundamentales, como puede ser el sistema de renderizado visual, el sistema de simulación dinámica o la reproducción de sonidos (Fernandez et al., 2011). La separación de los componentes facilita un paradigma orientado a la reutilización de código con el objetivo de generar aplicaciones interactivas con gráficos en tiempo real, de un mismo tipo sin tener que modificar el núcleo de programación. Por lo tanto, permite dirigir la mayor parte del esfuerzo a añadir nuevos elementos aprovechando la independencia del gestor de los recursos o modificar algunas reglas propias.

Este concepto ha surgido gracias a la aparición de nuevas herramientas para automatizar tareas con cierto nivel de complejidad, abstrayendo muchos procesos de bajo nivel. La utilización de un motor de juegos puede ser importante debido a varias razones: aportar facilidad en el desarrollo del software o proporcionar portabilidad en varias plataformas dependiendo de las librerías utilizadas. En función de lo portables que sean las librerías, el motor puede funcionar en un cierto número de Sistemas Operativos. Una mayor modularidad beneficia a la colaboración de varios grupos de trabajo que funcionen en paralelo, reduciendo la carga de trabajo en programación al aumentar el grado de abstracción. Sin embargo, puede que algunos elementos específicos sean imposibles de reutilizar, debido a que existe una dependencia entre entre el género del juego y el tipo de motor utilizado. Tampoco existe un motor que pueda generar cualquier tipo de situación. Una manera de mejorar la reutilizacion de los módulos del motor es evitar la implementación de comportamiento de juego dentro del código fuente a partir de un sistema de *Scripts*, lo que permite una optimización del tiempo de desarrollo. Normalmente, el género de los juegos o simuladores dependen del motor de juego utilizado. Sin embargo, su arquitectura está formada por módulos de bajo nivel independientes del género.

Actualmente, existe una gran variedad de motores de juego diseñados para todo tipo de géneros (Vasudevamurt y Uskov, 2015). Los motores co-

merciales tienen la ventaja de ofrecer mejores prestaciones como un soporte oficial o herramientas de desarrollo adicionales. Sin embargo, el precio de las licencias suelen ser un inconveniente. Por otro lado, los motores de código libre permiten la modificación del motor para adaptarlo a las necesidades de desarrollo de manera gratuita.

Por lo tanto, el objetivo de este capítulo es desarrollar un motor de tareas, utilizando la modularidad de los motores de juego y herramientas de software libre, capaz de facilitar la generación de tareas virtuales de manera rápida y adaptable para pacientes con ACV en terapias de neurorehabilitación del miembro superior asistida por dispositivos robóticos de efector final. Debido a la configuración de este tipo de dispositivos robóticos donde el movimiento del efector final corresponde al movimiento de la extremidad humana, el motor de tareas permitirá la simulación de entornos que se ajusten a este tipo de movimiento. Estos ejercicios consistirán básicamente en realizar patrones de movimientos diagonales (Dickstein et al., 1986) para trasladar objetos de un lugar a otro o colocar el efector final en ciertas posiciones. Como el sistema está orientado a pacientes con ACV de diferentes niveles de lesión, también debe ser capaz de adaptar la escena a los parámetros de la terapia añadidos por el terapeuta, como el número de repeticiones, nivel de asistencia, amplitud de movimiento, tiempos y pausas, sin necesidad de modificar el código fuente. Otro punto de interés de este capítulo es la incorporación de un subsistema de adaptación del nivel de dificultad dentro del motor de tareas para actualizar el estado de la simulación del entorno, en función del estado emocional del usuario obtenido por el procesamiento de sus señales fisiológicas (Badesa et al., 2014c).

3.2. Métodos y Componentes

En esta sección se plantea una visión general de la estructura modular del motor de tareas propuesto, con el objetivo de separar las diferentes funcionalidades. En la Figura 3.1 se muestra el diagrama con los principales módulos que forman parte del motor de tareas:

Para el desarrollo de este tipo de software es necesario la utilización de herramientas que faciliten la implementación, automatizando los procesos de construcción de proyectos, integración de diferentes componentes, compilación, enlazado con bibliotecas existentes, gestión de memoria y depuración. Estas herramientas dan soporte al sistema de simulación para integrar los

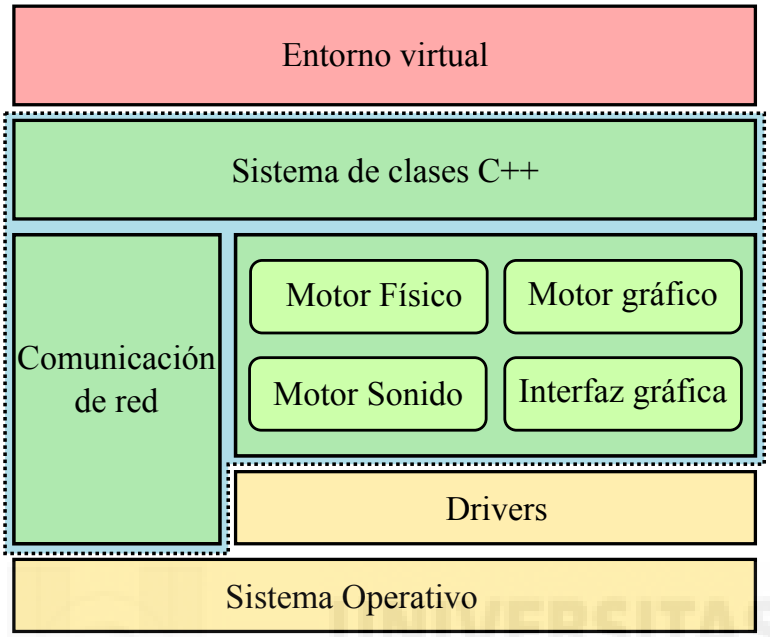


Figura 3.1: Arquitectura modular del motor de tareas propuesto.

elementos pertenecientes al control interno de las tareas. El primer elemento que se necesita es una plataforma encargada de la gestión del motor de tareas que soporte los elementos software de bajo nivel y permita el manejo de los dispositivos externos.

El Entorno de Desarrollo Integrado (IDE) utilizado en esta tesis para crear el núcleo de programación y el control del motor de tareas ha sido *Microsoft Visual Studio 2010*. Es una herramienta software que proporciona todo lo necesario para facilitar el desarrollo de todo tipo programas dirigidos por datos (editor de código fuente, compilador, interpretador, herramientas de construcción automática y depurador). Con esta herramienta se unifica todo el desarrollo del sistema a través de una programación orientada a objetos, beneficiándose de la ventajas que aporta este paradigma como son la reusabilidad, la modificabilidad, la fiabilidad y facilidad de mantenimiento.

Se ha utilizado la metodología llamada Proceso Racional Unificado (RUP) (Kruchten, 2004) como un conjunto de métodos adaptable a las necesidades del sistema y no como un proceso de desarrollo de software con pasos establecidos. Este proceso de ingeniería de software define tareas y responsabilidades dentro del desarrollo organizado de un sistema, y constituye la

metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Por otro lado, se ha utilizado el Lenguaje Unificado de Modelado (UML) (Rumbaugh et al., 2004) para documentar y describir aspectos o métodos del sistema software.

3.2.1. Motor gráfico de visualización

El elemento principal para desarrollar aplicaciones basadas en realidad virtual es el motor de renderizado gráfico o librería gráfica. Su misión principal consiste en generar imágenes bidimensionanes a partir de una cámara virtual que visualiza modelos o escenas con una geometría, puntos de luz, texturas y muchos elementos más, los cuales se representarán en el dispositivo de salida visual mediante una Unidad de Procesamiento Gráfico (GPU). De esta manera, el usuario puede visualizar el escenario y tomar las correspondientes decisiones basándose en la imagen final renderizada.

Existe gran cantidad de herramientas destinadas a este objetivo, sin embargo en esta tesis se ha utilizado Oriented Graphics Rendering Engine 3D (OGRE3D) (Steve, 2013) (Figura 3.2). Esta herramienta aporta una gran cantidad de funcionalidades con características técnicas avanzadas lo que permite tener una calidad de diseño bastante amplia. OGRE3D es un motor de renderizado de gráficos en 3D que facilita el diseño, desarrollo e implementación de aplicaciones de realidad virtual con entornos bastantes realistas, proporcionando interactividad y ejecución en tiempo real. Además, proporciona una gran variedad de estructuras de datos evitando la necesidad de librerías externas para complementar la funcionalidad del renderizado.

Figura 3.2: Logotipo del motor gráfico de visualización - OGRE3D.

Las razones de utilizar este motor son una serie de características destacables para la implementación del sistema encargado de generar entornos de rehabilitación virtual y se enumeran a continuación:

- **Lenguaje orientado a objetos**: OGRE3D está desarrollado en lenguaje C++ permitiendo una programación sencilla y eficiente. Por lo

tanto utiliza una arquitectura extensible de objetos.

- **Licencia LGPL**: El motor es libre y de código abierto bajo licencia Lesser GNU Public License (LGPL).
- **Multiplataforma**: Ofrece soporte para el desarrollo de aplicaciones en las plataformas de Windows, Linux, Mac OS X, iOS y Android.
- **Interfaz de programación de alto nivel**: Abstrae las funcionalidades de bajo nivel de Direct3D (Kovach y Richter, 1999) y OpenGL (Woo et al., 1997) para comunicarse con el hardware mediante métodos intuitivos que permiten la manipulación de todos los elementos sin necesidad de realizar una gestión manual de la geometría o sus propiedades de transformación.
- **Arquitectura modular extensible**: Como este motor se centra exclusivamente en la parte gráfica, permite utilizar una gran variedad de plugins para extender las funcionalidades en caso de ser necesario.
- **Grafos de escena** Utiliza grafos de escena basados en estructuras Octree para almacenar información y organizar los elementos del entorno, aportando una manipulación automática de la gestión del estado de visualización y aplicando optimizaciones para decidir la geometría que se va renderizar.
- **Materiales y Composición**: Gestiona un sistema de interpretación de scripts para definir los materiales encargados de describir las propiedades de las superficies o implementar efectos de postprocesado en tiempo de ejecución.
- **Aceleración por hardware**: Permite una aceleración por hardware, gracias a la funcionalidad de OGRE3D para definir *Shaders* utilizando técnicas programables de GPU de alto nivel y los lenguajes de desarrollo de *Shading* como Cg, High Level Shader Language (HLSL) y OpenGL Shading Language (GLSL).

Además de las razones anteriores, OGRE3D incluye una serie de bibliotecas de clases con una amplia gama de funciones matemáticas para trabajar con puntos de coordenadas, vectores, matrices, cuaternios, y otras entidades matemáticas. Esto es necesario en todo desarrollo de aplicaciones gráficas

para manipular las transformaciones de posición, rotación y escalado de objetos descritos en el espacio 3D.

3.2.2. Motor físico de simulación dinámica

PhysX (Li-xin, 2009) es un motor de cálculo de físicas desarrollado por NVIDIA (Figura 3.3) para simular objetos con un alto grado de realismo utilizando aceleración basada en hardware. Proporciona algoritmos de detección de colisiones para calcular interacciones entre los elementos de una escena simulada en tiempo real y genera respuestas de dichos elementos virtuales ante la influencia de estas fuerzas, determinando su movimiento y dirección. Sus módulos físicos resuelven los comportamientos dinámicos de cuerpos rígidos, cuerpos deformables (Maciel et al., 2009) o tejidos y sistemas de partículas para fluidos (van der Laan et al., 2009; Winkler et al., 2005), liberando a la Unidad Central de Procesamiento (CPU) de realizar una gran carga de cálculos físicos, por lo que aumenta mucho más la rapidez de la simulación gracias a la programación de la GPU (Fernando, 2004). Se puede ejecutar en multitud de plataformas y su Kit de Desarrollo de Software (SDK) es de comercialización gratuita pero de código cerrado. Tiene una interfaz C++ implementada como una jerarquía de clases que contienen funcionalidades accesibles para realizar cálculos específicos al hardware utilizado.

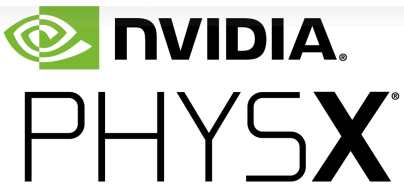


Figura 3.3: Logotipo del motor físico de simulación dinámica - PhysX.

Esta librería también incluye una variedad de funciones útiles cuando se trabaja con simulaciones físicas, aparte de clases matemáticas para la creación de vectores, matrices, cuaternios y otras entidades matemáticas. El SDK incorpora un sistema de creación interna de mallas para desarrollar formas destinadas a la detección eficiente de colisiones. Entre las características que posee el motor físico, se pueden destacar de manera general:

- Generación específica de fuerzas para crear efectos físicos como la

gravedad o fricciones estáticas y dinámicas, e incluso las propiedades del material de las superficies.

- Calcular colisiones y diseñar objetos con formas complejas de manera articulada.
- Utilización de conceptos físicos como puntos de referencia, posiciones, velocidades, aceleraciones, momentos, fuerzas, movimientos rotacionales, energías, fricciones, impulsos, colisiones, restricciones, etc.
- Detección continua de colisiones para evitar que cuando los objetos alcancen velocidades elevadas se atraviesen.
- Tiempos que tardan los objetos dinámicos en ponerse a reposar, lo que permite optimizar el cálculo de físicas descartando los objetos que permanecen parados.

En motor es necesario para aportar un dinamismo gestionado por una detección de colisiones que permita un comportamiento realista de la dinámica de los cuerpos rígidos situados dentro del entorno virtual mientras dichos elementos interactúan con el avatar gestionado por el usuario.

3.2.3. Motor de reproducción de sonidos

Otro aspecto esencial es la incorporación de un sistema de reproducción de audio que dote de realismo y permita una mejora del punto de vista del usuario con respecto a su inmersión dentro de entorno virtual. De esta manera, se añaden elementos sonoros que concuerdan continuamente con eventos producidos durante el ciclo de vida de una simulación, adaptándose a las decisiones y acciones del usuario. Para cumplir con este aspecto se ha utilizado la librería OpenAL (Hiebert, 2005) junto con un *wrapper* llamado OgreOggSound (Figura 3.4) que permite la integración de audio con el motor de renderizado gráfico OGRE3D. El objetivo principal del motor de sonidos es gestionar las operaciones de reproducir, pausar y parar un elemento de audio en cualquier instante durante la simulación, así como modificar el volumen o la posición en el espacio 3D, producir una reproducción indefinida, realizar cambios de puntos dentro de un sonido o saltar a otras pistas de audio a partir de transiciones.

Entre sus múltiples funcionalidades destaca su soporte para formatos de audio OGG y WAV, y una ejecución en multihilo. Su interfaz consiste en

Figura 3.4: Logotipo del motor de sonidos - OgreOggSound/OpenAL.

un número de funciones que permiten especificar objetos y operaciones con el objetivo de especificar varios canales de salida de estructuras de fuentes de sonidos para posicionarlas en un espacio tridimensional. De esta manera, se permite una espacialización razonable de las fuentes para el sistema de audio como pueden ser los auriculares, altavoces 2.1, altavoces 5.1, 7.1, etc.

3.2.4. Interfaz Gráfica de Usuario

El último aporte en el desarrollo del motor de tareas es la incorporación de una Interfaz Gráfica de Usuario (GUI) que se encargue de mostrar información al usuario sobre la acción que está realizando en cada momento. La regla básica a seguir por toda GUI es atraer la atención justa del usuario que permita mantener un nivel de concentración aceptable (Johnson y Wiles, 2003). Este componente permite diseñar los siguientes elementos gráficos útiles para este sistema:

- *Splash Screen*: Es la pantalla que aparece antes de ejecutar un juego para indicar que la carga de recursos puede demorarse un tiempo. De esta manera, el usuario no visualiza directamente el escenario virtual.

- *Heads-Up Display*: Este elemento se encarga de mostrar información de la tarea acerca de las acciones que realiza el usuario y está formado por *Widgets*.

El desarrollo de *Widgets* para desarrollar estos elementos gráficos desde cero puede llevar un largo y costoso trabajo nada trivial, junto con la complejidad que lleva el gestionar cada uno de estos componentes. Por lo tanto, se ha utilizado una librería llamada Crazy Eddie's GUI (CEGUI) (Figura 3.5) (TURNER, 2006), la cual utiliza este tipo de componentes y permite su integración con el motor de renderizado gráfico OGRE3D.

CEGUI es una biblioteca que proporciona un entorno basado en ventanas y *Widgets* para Application Programming Interface (API)s gráficos que

Figura 3.5: Logotipo de la librería gráfica de usuario - CEGUI.

no dan soporte nativo a esta funcionalidad. La librería es multiplataforma y está orientada a objetos, de manera que ofrece potencia y flexibilidad. CEGUI puede considerarse un motor de gestión de *Widgets*.

3.3. Requerimientos del sistema

En la fase de inicio se propone definir cuál es el problema planteado y los objetivos necesarios para solucionarlo, analizando de manera muy general la arquitectura software que se quiere producir e identificando los requisitos funcionales y no funcionales para comprobar los requerimientos que tiene que alcanzar el sistema.

3.3.1. Planteamiento del problema

Una terapia de neuro-rehabilitación está vinculada al tratamiento de personas con la necesidad de recobrar alguna condición o estado que han perdido a causa de alguna lesión cerebral u enfermedad, realizando algún tipo de ejercicio o tarea repetitiva dependiendo de la zona afectada del cerebro. Esto quiere decir que para obtener una completa terapia basada de entornos virtuales y asistidos por dispositivos robóticos se debería de disponer de un número de diferentes tareas virtuales que pueda realizar un paciente, buscando una variedad de ejercicios para motivar e incentivar un poco más su participación durante la terapia evitando que procese un sentimiento de frustración o aburrimiento, los cuales son los factores que más afectan a la hora de abandonar las terapias.

La generación de una tarea implica la necesidad de crear un proyecto único con un fichero ejecutable que contenga la configuración de un entorno virtual y la programación de los objetivos que el paciente debe cumplir con los elementos seleccionables. El implementar un proyecto para cada tarea, añadiendo los componentes descritos en el motor de juego y establecer una

escena manualmente con sus respectivos modelos físicos es un proceso lento y laborioso. Por lo tanto, el objetivo de este capítulo es desarrollar un sistema de arquitectura software que permita generar ficheros ejecutables con tareas virtuales que posean un comportamiento visual y físico realista, destinadas a pacientes con ACV durante la realización de terapias de rehabilitación asistidas por dispositivos robóticos. Adicionalmente, el nivel de dificultad de las tareas virtuales se deberá adaptar automáticamente en función de los comandos recibidos de dispositivos externos de captura y análisis de las señales fisiológicas de los usuarios. Esta arquitectura software permite optimizar el proceso de creación de tareas para la terapia de manera rápida con el menor coste y los mínimos recursos posibles. De esta manera, la parte de programación queda relevada a un segundo plano necesitando la mayor parte del esfuerzo en el diseño de la parte visual y física del escenario, las interfaces gráficas de usuario con sus eventos, la organización o reproducción de sonidos y el planteamiento de los objetivos de la nueva tarea.

Para el diseño de este sistema software se ha propuesto una serie de exigencias con el objetivo de logar la generación de tareas con el suficiente realismo y capacidad de interacción. Estas exigencias son:

1. Los objetos virtuales se deben visualizar de la forma más realista posible mediante una alta densidad de polígonos y calidad de texturas.

2. Los elementos que intervienen en la simulación se deben comportar de una manera similar a la realidad, al aplicarse fuerzas sobre ellos.

3. Las tareas necesitan una interacción en tiempo real (Akenine-Möller et al., 2008).

4. Con el fin de motivar al paciente durante la realización de las tareas terapéuticas se debe incorporar refuerzos sonoros cuando se completan las tareas para indicar de manera sonora la realización de algún objetivo, o sonidos de interacción o incluso música de fondo.

5. La realidad virtual permite simular cualquier tipo de tarea, no realistas y AVD, estableciendo objetivos sencillos de completar mediante la realización de determinados movimientos.

3.3.2. Visión general

De manera general, el sistema propuesto tiene por finalidad el cargar unos ficheros de recursos multimedia y organizarlos para generar un fichero ejecutable con una tarea virtual cuyo nivel de dificultad se actualice automáticamente. Esto quiere decir, que el motor de tareas es un plantilla con módulos software interconectados de manera que al ejecutarlo se encarga de decodificar los recursos multimedia, crear una ventana donde muestra todo el contenido virtual y gestionar el comportamiento físico, visual y acústico. En la Figura 3.6 se presenta una visión general del objetivo de este apartado mostrando el contexto donde se aplica el motor de tareas y cuando se realiza la creación del contenido multimedia.

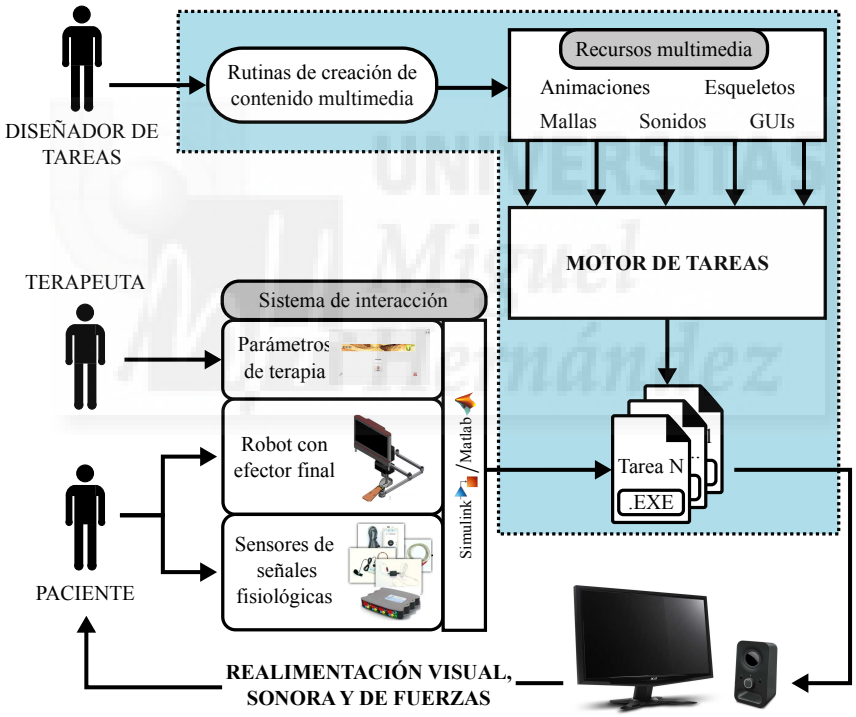


Figura 3.6: Visión general de los elementos que interactúan en el sistema.

Como complemento al desarrollo del motor de tareas se establece un proceso de producción de contenido multimedia para completar el sistema de RV. Este proceso no es una arquitectura de software unificada sino que es una serie de rutinas de diseño utilizando programas de edición *Open Source* (Initiative, 2015), siempre siguiendo un patrón de desarrollo.

Este sistema ofrece al diseñador de la tarea, teniendo en cuenta las indicaciones del terapeuta, la posibilidad de agregar cualquier tipo de escenario junto con diferentes objetos virtuales. Ya que los pacientes pueden sufrir desordenes físicos o psicológicos los entornos deben ser sencillos y claros para que pueda comprender los objetivos rápidamente. La potencialidad principal reside en que una vez que se diseña un objeto o un escenario virtual junto con sus respectivos componentes físicos, éstos se pueden reutilizar dentro de cualquier tarea implementada posteriormente sin necesidad de añadir más líneas en el código fuente. Este hecho mejora la eficiencia, la velocidad y la eficacia del sistema propuesto para generar más tipos de tareas para las terapias virtuales. Existe gran cantidad de juegos libres diseñados con OGRE3D que han guiado el diseño de este motor de tareas (Saltares Márquez, 2011).

Con respecto a los objetivos o desafíos que el paciente debe realizar para completar las tareas se tiene en cuenta las posibilidades que ofrece el dispositivo de control, que en este caso son los dispositivos robóticos de efector final. Debido a la estructura mecánica formada por el efector final, la única interacción que puede realizar el paciente es el movimiento repetitivo de este efector final desde un punto del espacio de trabajo hasta otro, para realizar trayectorias o ejercicios de rehabilitación con movimientos específicos de la extremidad superior. Bajo esta premisa, los objetivos dentro de las tareas virtuales se traducen en el desplazamiento de un objeto controlable buscando un objetivo alcanzable para completar la tarea. Teniendo en cuenta este aspecto es necesario incorporar un gestor automático de objetivos de las tareas para automatizar por completo el sistema.

En conclusión, este sistema es una plantilla de software que gestiona un proceso optimizado de programación para generar tareas cuya organización del entorno virtual, asignaciones de comportamiento físico y la asignación de memoria se gestionan de forma automática. La única acción necesaria para generar diferentes tareas es la incorporación de diversos recursos multimedia. En definitiva, todas las tareas compartirán la misma finalidad: la realización de trayectorias para completar los objetivos en diferentes entornos virtuales.

3.3.3. Requisitos del sistema

En esta sección se van a describir los servicios proporcionados por el software relacionados con el funcionamiento esperado por parte de los usua-

rios, definiendo todos los aspectos funcionales que debe cumplir el motor de tareas y los no funcionales como restricciones operativas. Hay que destacar, que no se realiza una captura de requisitos para las interfaces gráficas debido a que estos elementos sólo mostrarán información sobre la tarea ejecutada.

3.3.3.1. Requisitos funcionales

Los requisitos funcionales describen la funcionalidad que se espera del sistema software (más profundamente es lo que se espera de las tareas virtuales), registrando la interacción entre el sistema y los que se consideran como usuarios que son el Terapeuta y el Paciente. Las tareas generadas contarán con la siguiente lista de requisitos funcionales:

- Crear y organizar de manera automática todos los elementos de la escena virtual
- Mover el elemento controlable formado por un avatar para interactuar con el escenario virtual.
- Salir de la tarea en cualquier momento, en caso de dificultad elevada para el Paciente.
- Visualizar una GUI con información actualizable de objetivos.
- Establecer objetivos predeterminados de alcanzar zonas o recoger objetos con el avatar principal de control.
- La aplicación debe ser capaz de recibir una serie de parámetros de la terapia para adaptar la tarea a las necesidades del paciente.
- Recibir datos desde el sistema de interacción formado por el dispositivo robótico y los sensores de registro de datos fisiológicos.
- Actualizar la cantidad de elementos que aparecen en escena junto con sus movimientos y los parámetros de terapia para modificar el nivel de dificultad de la tarea en función de los comandos recibidos por el clasificador implementado en el sistema de interacción.

Con respecto a la generación de contenido multimedia para complementar las tareas virtuales se pueden definir los siguientes requisitos funcionales a realizar por el Diseñador de tareas:

- Modelar un escenario y objetos virtuales.
- Editar formas físicas solapadas con los elementos virtuales.
- Generar una interfaz gráfica de usuario para mostrar información.
- Crear ficheros de los recursos multimedia, con sus respectivas extensiones, compatibles con el sistema.
- Obtener sonidos correspondientes a cada objeto de la escena y los que sonarán cuando se cumplan los objetivos.
- Escribir un fichero de objetivos en función de los elementos de la escena y los sonidos.

3.3.3.2. Requisitos no funcionales

Estos requisitos representan a las características que de alguna manera pueden afectar al funcionamiento del sistema, restringiendo algún aspecto. En todo diseño y generación de aplicaciones con entornos virtuales el aspecto más importante que se debe tener en cuenta es la fluidez del sistema a la hora de generar las imágenes que se presentarán al usuario con una tasa de fotogramas por segundo (fps). Si esta tasa es demasiado baja se produce una ralentización en la visualización provocando una disminución del impacto en la percepción del usuario. Esta característica depende del uso de la CPU, debido a que si se encarga de la parte funcional del código y de la parte gráfica puede llegar a dar problemas.

Por lo tanto, estos procesos serán optimizados de manera que la parte gráfica se llevará a cabo en la GPU para intentar disminuir la carga gráfica provocada por un elevado número de elementos. Entonces, los entornos generados deberán presentar un escenario atractivo para el usuario sin mucha carga de componentes, limitando las funcionalidades que necesitan mucha carga computacional. Con respecto, a la implementación física del entorno se debe realizar con formas básicas para que la administración y el mantenimiento del comportamiento físico se puedan gestionar fácilmente.

3.4. Análisis del sistema

Una vez establecidos los objetivos iniciales con una visión general del formato de sistema, se realiza la fase de análisis para identificar las necesi-

dades que debe cumplir el sistema a partir de los requisitos obtenidos en la sección anterior. En primer lugar, se define el modelo de dominio, el cual contiene los tipos de objetos conceptuales más importantes dentro del contexto del sistema. Después, se declaran los actores que interactuarán con el sistema junto con una explicación de los casos de uso de cada uno. Para finalizar se modela el comportamiento de las dos partes del sistema.

3.4.1. Modelo de dominio

El modelo de dominio representa los conceptos clave dentro del dominio del problema. Muestra una representación de los objetos conceptuales necesarios para entender de forma abstracta el dominio del problema. La información mostrada se puede expresar mediante sentencias, pero de manera visual permite una mejor compresión de los distintos elementos y sus relaciones naturales. En la Figura 3.7 se puede apreciar el diagrama de dominio del motor de tareas.

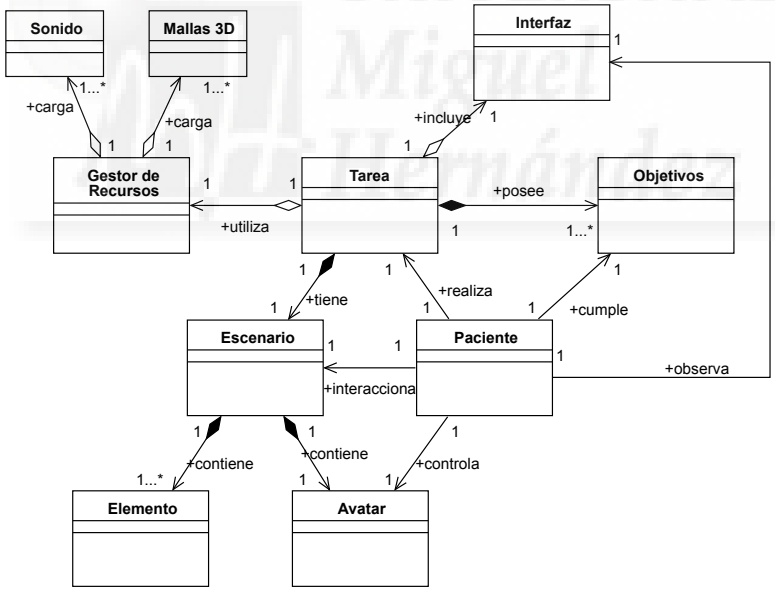


Figura 3.7: Diagrama de dominio del motor de tareas.

El paciente realiza la tarea que estará formada por un escenario compuesto por una serie de elementos visuales de interacción y un avatar controlado por el dispositivo robótico. Los recursos multimedia se cargan a partir de un

gestor de recursos y el núcleo de programación de la tarea se encargará de distribuir estos elementos para estructurar el entorno. Durante el desarrollo de la tarea, el paciente tiene la posibilidad de observar una interfaz con información sobre los objetivos que debe completar para finalizar la sesión de rehabilitación. En este caso el modelo de dominio presenta de manera visual las clases más importantes dentro del contexto del motor de tareas, las cuales se utilizarán en el diseño del modelo software. Durante la fase de diseño aparecen muchas otras clases que se pueden extraer del modelo de dominio. Una de estas clases puede representar un subsistema definido por un número de otras clases internas.

3.4.2. Casos de uso

En este tipo de análisis se definen cuáles son los posibles usuarios o actores del sistema y los distintos objetivos que pueden cumplir interactuando con él. Describe la secuencia de interacciones entre actores y el sistema para llevar a cabo algún procedimiento, definiendo relaciones de comportamiento. En resumen, los casos de uso estructuran de forma natural los requisitos funcionales del sistema.

Como se ha comentado, la funcionalidad principal del sistema es la generación de tareas virtuales a partir de la carga de una serie de recursos diseñados. Por consiguiente, se puede considerar que existen dos tipos diferentes de interacciones: por un lado está el proceso de creación de recursos, y por otro la rutina de programación para simular las tareas virtuales, también llamado motor de tareas.

3.4.2.1. Identificación de los actores

Debido a la estructura del sistema se pueden identificar tres posibles perfiles de usuario con unos roles propios. Los roles son: Diseñador de Tareas, Terapeuta y Paciente. Cada actor interactúa de manera diferente con el sistema. El actor principal del sistema será el Paciente, el cual siempre estará supervisado por el Terapeuta. Sin embargo, como también existe un proceso de creación de contenidos multimedia se puede definir el rol de Diseñador de Tareas:

- *Paciente*: Representa a la persona que ha sufrido un ACV y realiza la terapia virtual de rehabilitación asistida por dispositivos robóticos. Es

el rol que debe interactuar con los escenarios virtuales generados por el motor de tareas a través del efector final del dispositivo robótico.

- *Terapeuta*: Es el rol que deriva de la persona encargada de llevar el seguimiento de la terapia y establecer los objetivos necesarios para intentar conseguir una recuperación notable del Paciente.

- *Diseñador de Tareas*: Se trata del rol de la persona que implementa todo el escenario y los elementos virtuales de las tareas, siguiendo las indicaciones del Terapeuta para establecer qué elementos permitirán cumplir los objetivos de la tarea.

3.4.2.2. Descripción de los casos de uso

Un caso de uso representa una interacción entre un actor y el sistema en forma de secuencia de acciones. De manera que la descripción de los casos de uso se centra en detallar con claridad cuál es el objetivo y no cómo debe cumplirse. En este apartado se detallan los casos de uso de cada actor a partir de un formato de tabla compuesto por una serie de propiedades:

1. Caso de uso: Nombre del caso de uso.

2. Actor: Nombre de los actores que participan en el caso de uso.

3. Descripción: Breve descripción del caso de uso especificado.

4. Precondición: Condiciones cumplidas antes de realizar el caso de uso.

5. Postcondición: Situación del sistema cuando finaliza el caso de uso.

6. Secuencia: Lista numerada de los pasos a seguir por el actor para interactuar con el sistema.

Como se ha comentado anteriormente los casos de uso del sistema se van a clasificar en dos grupos: casos de uso durante la utilización de la tarea y los casos de uso en el diseño del contenido multimedia. En primer lugar se exponen los casos de uso que puede realizar el Paciente y el Terapeuta en la parte más importante del sistema, que es el motor de tareas encargado de simular los entornos. Como el motor de tareas genera diferentes tareas

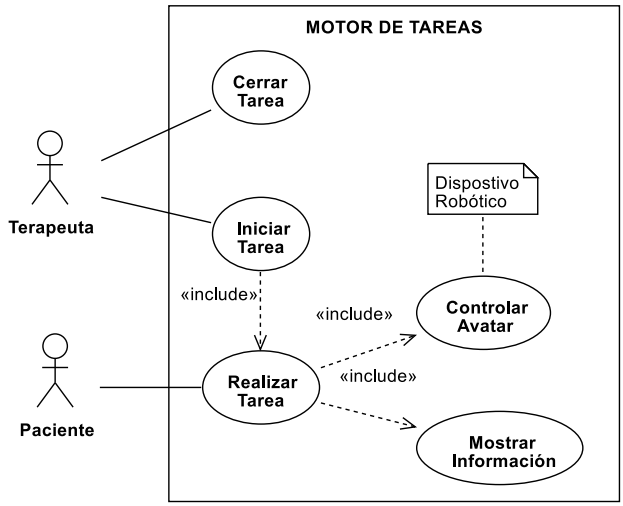


Figura 3.8: Diagrama de casos de uso del motor de tareas.

en función de los elementos virtuales cargados, estas tareas tienen el mismo formato de diagrama de casos de uso. En la Figura 3.8 se muestra el diagrama de casos de uso de una tarea general.

El motor de tareas permite al Paciente realizar la tarea virtual siempre que el Terapeuta inicie el ejecutable, e incluso debe permitir el cierre de la tarea. Mientras el paciente realiza la tarea a través de la interacción con el efector final del dispositivo robótico, puede observar información adicional sobre los objetivos a cumplir en cada momento. A continuación se describen los casos de uso expuestos en este grupo:

Caso de uso	Iniciar Tarea
Actor	Terapeuta
Descripción	El Terapeuta selecciona la tarea que tiene que realizar el paciente para completar una parte de la terapia de RV asistida por dispositivos robóticos.
Precondición	El Terapeuta aporta unos parámetros de terapia.
Postcondición	Genera una ventana con la tarea virtual.
Secuencia	1. El Terapeuta inicia la aplicación. 2. El Sistema despliega una ventana de carga de recursos. 3. El Sistema inicia el motor y muestra el escenario.

Tabla 3.1: Caso de uso: Iniciar Tarea.

Caso de uso	Cerrar Tarea
Actor	Terapeuta
Descripción	El Terapeuta puede cerrar la tarea en cualquier momento.
Precondición	Se debe estar realizando la tarea.
Postcondición	Registra las estadísticas de aciertos y errores de la sesión.
Secuencia	1. El Terapeuta selecciona la opción de cerrar. 2. El Sistema cierra la aplicación.

Tabla 3.2: Caso de uso: Cerrar Tarea.

Caso de uso	Realizar Tarea
Actor	Paciente
Descripción	El Paciente inicia la terapia interactuando en tiempo real con los elementos de la tarea, con la posibilidad de que cumplan o no los objetivos.
Precondición	La tarea debe estar iniciada.
Postcondición	Se completa una sesión de la terapia virtual, registrando el rendimiento del Paciente.
Secuencia	1. El Sistema carga los elementos de la escena. 2. El Paciente y el Sistema interactúan. 3. El Sistema indica que el Paciente ha finalizado los objetivos a través un efecto sonoro.

Tabla 3.3: Caso de uso: Realizar Tarea.

Caso de uso	Controlar Avatar
Actor	Paciente
Descripción	El Paciente mueve el avatar por el escenario a través del dispositivo robótico de asistencia.
Precondición	Se está realizando la tarea.
Postcondición	El avatar se desplazará por el escenario virtual.
Secuencia	1. El Paciente mueve el efector final de sistema robótico. 2. El Sistema desplaza el avatar de control según los movimientos del robot. 3. El Sistema genera colisiones entre los elementos.

Tabla 3.4: Caso de uso: Controlar Avatar.

Caso de uso	Mostrar Información
Actor	Paciente
Descripción	El escenario virtual puede tener una GUI encargada de aportar información adicional sobre los objetivos a cumplir.
Precondición	Se debe iniciar un objetivo de la tarea.
Postcondición	Muestra información sobre todos los objetivos.
Secuencia	1. El Sistema despliega la GUI. 2. El Sistema muestra información sobre los objetivos. 3. El Paciente completa un objetivo. 4. El Sistema indica el fin del objetivo con un efecto sonoro.

Tabla 3.5: Caso de uso: Mostrar Información.

El otro grupo de casos de uso son los correspondientes al diseño del contenido multimedia. Esta parte del sistema se refiere a las rutinas de programación y diseño que debe seguir el Diseñador de Tareas para generar los recursos que se utilizarán durante la ejecución de la tarea. En la Figura 3.9 se muestra el diagrama de casos de uso con las acciones a realizar por el Diseñador de Tareas y el Terapeuta:

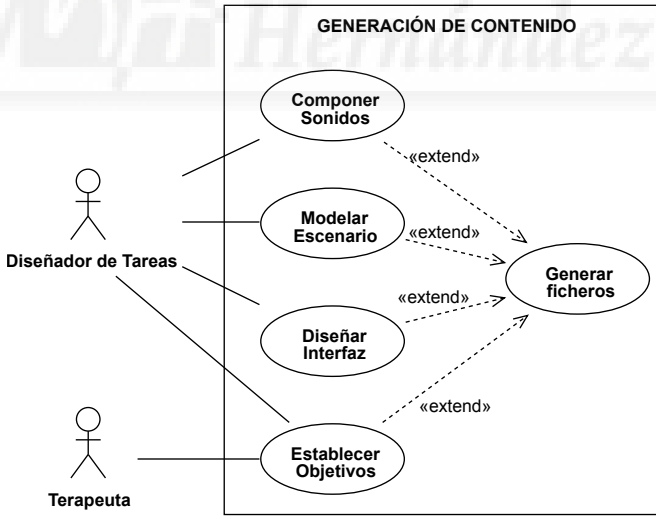


Figura 3.9: Diagrama de casos de uso para la generación de contenido multimedia.

Básicamente todas las acciones de diseño son realizadas por el Diseñador de Tareas, pero siempre bajo la supervisión del Terapeuta ya que éste se

encarga de decidir qué ejercicios son más beneficiosos. A continuación se describen los casos de uso expuestos en este grupo:

Caso de uso	Modelar Escenario
Actor	Diseñador de Tareas
Descripción	El Diseñador de Tareas debe utilizar una herramienta de modelado para obtener una representación del escenario correspondiente a una tarea.
Precondición	Se deben tener conocimientos de modelado con software.
Postcondición	Genera mallas 3D con sus materiales de superficie.
Secuencia	1. El Diseñador modela el escenario. 2. El Diseñador genera las formas físicas de los objetos. 3. El Sistema obtiene la parte visual y física de la tarea.

Tabla 3.6: Caso de uso: Mostrar Escenario.

Caso de uso	Diseñar Interfaz
Actor	Diseñador de Tareas
Descripción	El Diseñador de Tareas debe utilizar la herramienta de diseño de interfaces para aportar una GUI.
Precondición	Tener conocimientos de diseño de intefaces.
Postcondición	Genera los elementos necesarios para visualizar la GUI.
Secuencia	1. El Diseñador genera una interfaz gráfica. 2. El Sistema obtiene una GUI.

Tabla 3.7: Caso de uso: Diseñar Interfaz.

Caso de uso	Obtener Sonidos
Actor	Diseñador de Tareas
Descripción	El Diseñador tiene que obtener sonidos para simular los objetos o para cuando se cumpla algún objetivo.
Precondición	Saber que objetos componen la tarea.
Postcondición	Se obtienen los sonidos.
Secuencia	1. El Diseñador genera o descarga sonidos para los objetos. 2. El Sistema obtiene una lista de sonidos.

Tabla 3.8: Caso de uso: Obtener Sonidos.

Caso de uso	Establecer Objetivos
Actor	Diseñador de Tareas, Terapeuta
Descripción	En este caso de uso se definen los objetivos a cumplir en la tarea, indicando los elementos que participan.
Precondición	Se tienen que tener en cuenta los consejos de Terapeuta.
Postcondición	Establecer los objetivos de la tarea.
Secuencia	1. El Diseñador y el Terapeuta deciden que objetivos se buscan con la tarea diseñada. 2. El Diseñador establece los elementos implicados. 3. El Sistema obtiene una finalidad para la tarea.

Tabla 3.9: Caso de uso: Establecer Objetivos.

Caso de uso	Generar Ficheros
Actor	Diseñador de Tareas
Descripción	Todos los contenidos que se han creado son recogidos en ficheros Lenguaje de Marcas eXtensible (XML) de configuración para la generación del entorno de la tarea.
Precondición	Crear todos los recursos multimedia.
Postcondición	Se obtendrán los ficheros necesarios para la tarea.
Secuencia	1. El Diseñador genera un fichero con la organización de los elementos dentro de la escena. 2. El Diseñador establece el número de formas físicas. 3. El Diseñador desarrolla un fichero con los objetivos a realizar para cumplir la tarea. 4. El Diseñador añade los ficheros al directorio de recursos. 5. El Sistema registra todos los ficheros.

Tabla 3.10: Caso de uso: Generar Ficheros.

3.4.3. Modelado del comportamiento

En este apartado se detalla el modelo de comportamiento del sistema diferenciando los dos grupos de casos de uso definidos en el apartado anterior. Para el caso del motor de tareas se presenta un diagrama de secuencia, mientras que para la generación de contenido se muestra un diagrama de actividad.

3.4.3.1. Diagrama de secuencia del motor de tareas

El diagrama de secuencia se encarga de identificar como se comunican internamente las clases principales para la realización de los casos de uso. Normalmente, se aplica un diagrama de secuencia por cada caso de uso, pero en este caso se pueden representar todos los casos de uso en un diagrama de secuencia. En los diagramas de secuencia, los mensajes deben corresponder a funciones que contienen las clases que los intercambian, sin embargo en esta etapa donde aún se busca analizar el problema planteado aparecen mensajes textuales que en etapas posteriores se corresponderán a métodos concretos de cada fase.

En la Figura 3.10 se representa un diagrama general del motor de tareas que define los casos de uso: Iniciar Tarea, Salir Tarea, Realizar Tarea, Controlar Avatar y Mostrar Información. Define de manera global las clases iniciales y la interacción entre actores y el sistema.

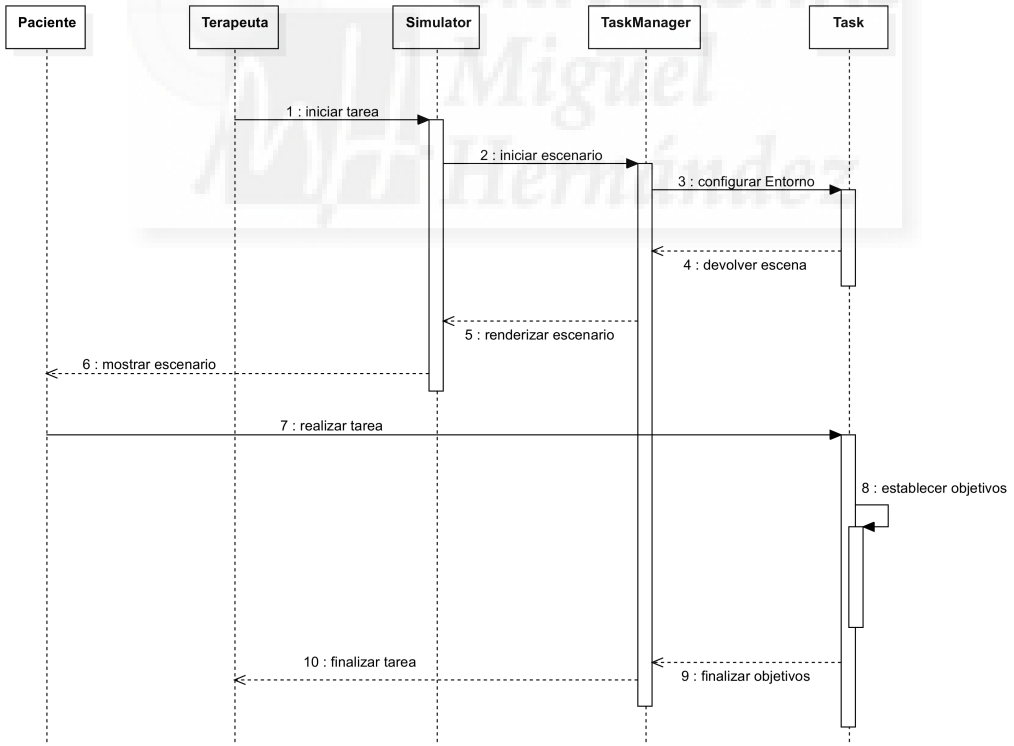


Figura 3.10: Diagrama de secuencia de las tareas simuladas por el motor.

Por consiguiente, el Terapeuta inicia la tarea para llamar a la clase *Simulator* para inicializar el estado interno de la tarea con *TaskManager* y crear una instancia de la clase *Task*. Esta última clase se encarga de generar el escenario de la tarea, estableciendo todos los objetivos e incluso actualizando el nivel de dificultad. Entonces, el Paciente ya tiene una visualización del escenario con el objetivo a cumplir mediante el movimiento del avatar y posee información sobre los pasos que debe que realizar para completar la tarea. Una vez finalizado todos los objetivos, la aplicación se cierra eliminando todo el contenido visible.

3.4.3.2. Diagrama de actividad para la creación de contenido.

El proceso de creación de contenido multimedia puede representarse como un diagrama de actividad debido a que no consiste en un subsistema de programación, sino en una rutina de procedimientos que debe seguir el Diseñador de Tareas para poder generar recursos de una manera correcta y complementar al motor de tareas. Este diagrama describe el procedimiento de desarrollo de recursos a través de una serie de acciones. La Figura 3.11 representa el protocolo de creación de contenido en un diagrama de actividades separado en cuatro subactividades dependiendo del tipo de recurso.

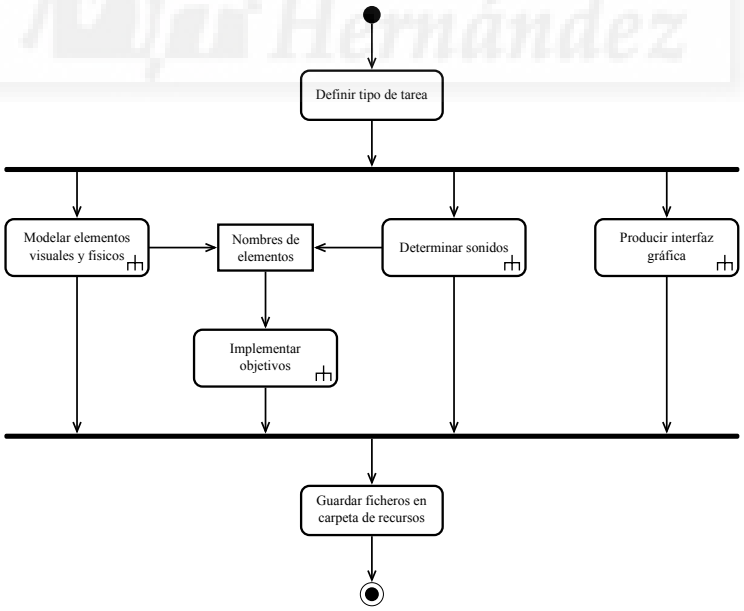


Figura 3.11: Diagrama de actividades general para la creación de contenido multimedia.

La primera decisión que se debe tomar es elegir el tipo de tarea a completar por el Paciente. Cada tarea implica la necesidad de implementar cuatro tipos de recursos diferenciados dentro del diagrama de actividad. El primer tipo de recursos corresponde a la parte visual y física de la tarea formados por mallas 2D o 3D, materiales de superficies, texturas y *scripts* de datos. Estos elementos se obtienen utilizando un software de modelado, el cual permite crear entornos virtuales con cualquier tipo de elemento decorativo y el avatar que controlará el Paciente. Todos estos elementos visuales se pueden desarrollar desde cero o pueden ser descargados desde sitios webs donde se encuentran modelos predefinidos de manera gratuita. Sin embargo, estos modelos descargados suelen tener muchos vértices innecesarios que reducen el rendimiento de las aplicaciones virtuales en tiempo real, por lo que necesitan un proceso de optimización de vértices y caras para reducir ese número. En la Figura 3.12 se muestran las acciones que debe realizar el Diseñador de tareas para obtener este tipo de recursos.

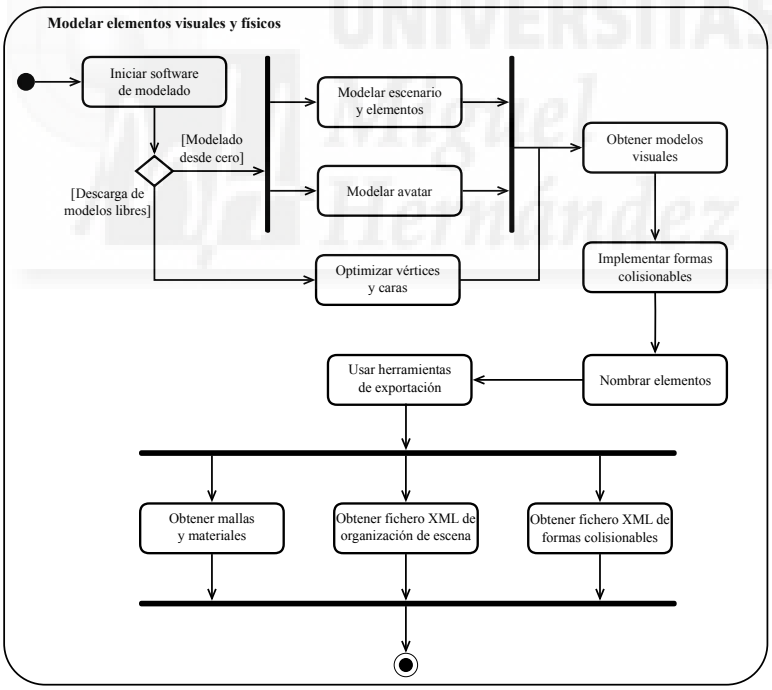


Figura 3.12: Diagrama de actividades para modelar los elementos visuales y físicos.

Un vez implementados todos los elementos visuales, se modelan las formas colisionables que actuarán dentro de la simulación física. Para com-

pletar este punto, se solapan formas como cajas, cápsulas, planos, esferas, cuerpos convexos y todo el tipo de formas soportadas por el motor físico, justo encima de los componentes visuales dentro de la herramienta de modelado hasta completar todas las zonas que necesitan tener comportamiento físico. De esta manera, se puede ajustar visualmente el tamaño colisionable de los elementos. Todos los elementos deben estar nombrados de una manera especial para que el módulo de lectura de ficheros XML sea capaz de reconocer a qué tipo de objeto y simulación física pertenece cada uno. Por consiguiente, se utiliza una herramienta de exportación especial para obtener los ficheros de mallas en el formato nativo que utiliza el motor de gráficos y sus correspondientes materiales de superficies. Mediante esta herramienta también se pueden extraer los ficheros XML que contienen la organización espacial de todos los elementos implementados, así como sus nombres, orientaciones y tamaños.

En la Figura 3.13 se muestran el diagrama de actividades que describen las acciones de generación del resto de recursos. El siguiente tipo de elemento necesario para aportar un poco más de realismo son los sonidos. Por consiguiente, se pueden obtener de dos maneras diferentes: importarlos directamente desde sitios web con sonidos libres o utilizar un programa de edición. Por otro lado, la creación de una GUI que aporte información sobre la tarea es completamente optativo dependiendo del tipo de tarea, ya que algunas pueden ser sencillas y no necesitar este recurso o pueden ser más complicadas al tener la necesidad de seguir una serie de pasos.

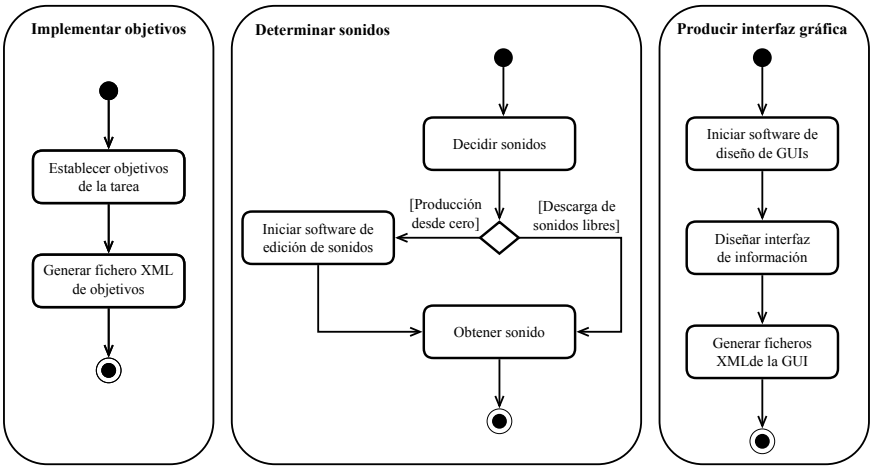


Figura 3.13: Diagrama de actividades para modelar elementos complementarios.

El último tipo de recurso contiene los objetivos a realizar dentro de la tarea. Este recurso recolecta en un formato XML los nombres de los elementos visuales que participan en cada objetivo, señalando el tipo de sonido que debe realizar al completarse, mostrando información complementaria o indicando como se debe actualizar la tarea al cambiar el nivel de dificultad. Para finalizar, todo el contenido multimedia desarrollado con este protocolo debe ser importado al directorio de recursos que utiliza el motor de tareas para cargar contenidos.

3.5. Fase de diseño general

En esta sección se va a presentar el diseño arquitectural modular del sistema para crear una estructura software que soporte todos los requisitos analizados. Esta fase se va a centrar más en el diseño del motor de tareas, debido a que es la única parte del sistema que posee una arquitectura software. En primer lugar, se va definir un modelo de arquitectura modular para tener una organización de las clases del motor de tareas. Para finalizar, se mostrará la jerarquía de clases obtenida.

3.5.1. Arquitectura modular

En primer lugar, se define un modelo de arquitectura general para obtener un diseño de alto nivel que permite organizar el sistema en una serie de módulos y relaciones existentes entre ellos. De este modo, se puede tener una estructuración de las futuras clases que compondrán el motor de tareas. En la Figura 3.14 se muestra la arquitectura general diferenciando los componentes principales que servirán de referencia para crear las clases del proyecto. La parte de la creación de contenido no necesita un diseño previo, ya que consiste en utilizar rutinas de programación con herramientas libres.

El motor de tareas posee una arquitectura estructurada por varios módulos que se pueden organizar según su funcionamiento.

- Núcleo: Es el módulo principal que sustenta toda la ejecución del ciclo de vida de la tarea gestionando la organización del escenario y el control del avatar. Se encarga de inicializar el motor haciendo de enlace con el resto de los módulos para ejecutar los bucles de renderizado del motor gráfico, de la interfaz gráfica y del motor de sonido, así como

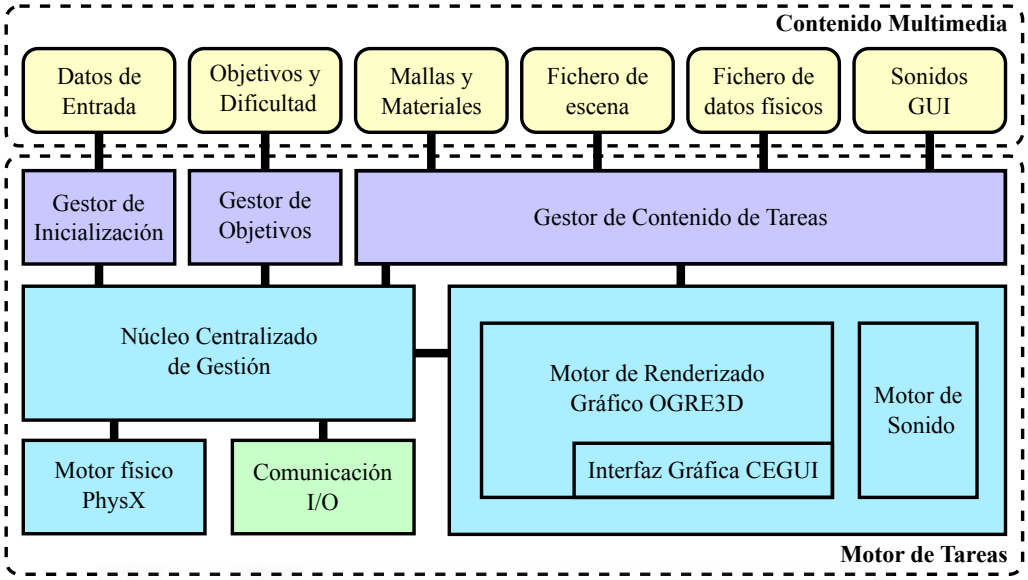


Figura 3.14: Arquitectura modular del motor de tareas y los recursos multimedia.

gestionar la simulación de comportamientos dinámicos y actualizar el nivel de dificultad. Además, establece todas las reglas lógicas de la ejecución del software.

- Gestores de entradas de contenido: Estos módulos se encargan de gestionar la carga de los recursos necesarios para ejecutar la tarea a partir de la lectura de los ficheros de mallas, materiales y *scripts* de datos XML.

- Comunicación I/O: Este módulo de entrada/salida gestiona la iteraciones entre el Paciente y el dispositivo robótico, permitiendo una interacción con el entorno virtual a través del movimiento del avatar. También, obtiene la comunicación con el clasificador de señales fisiológicas para recibir el comando que indica el cambio de nivel de dificultad.

Este tipo de estructura proporciona una segmentación del software que permite separar la lógica de la aplicación en diferentes capas. La Figura 3.15 muestra la distribución de las capas: capa de presentación, de aplicación y de datos.

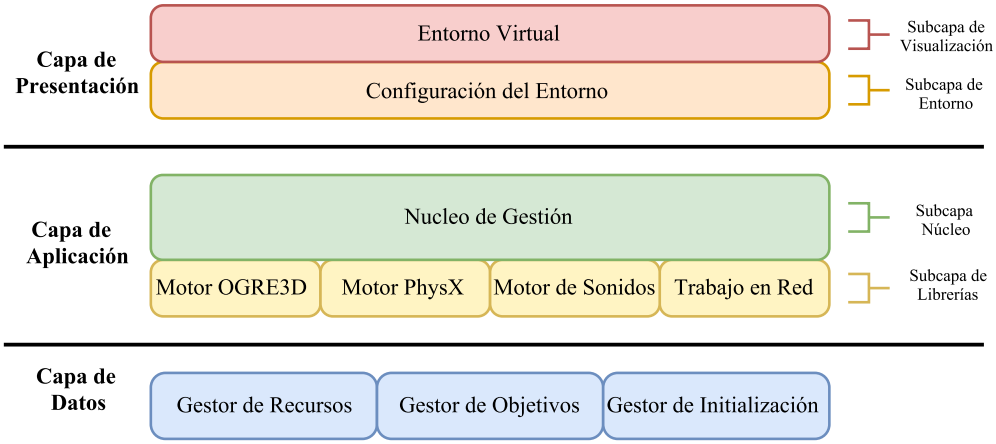


Figura 3.15: Distribución de las 3 capas del sistema.

3.5.2. Diagramas de clases

En este apartado se definen las clases que componen cada uno de los módulos que forman la arquitectura del motor de tareas y se presentan los diagramas para una mayor claridad. Con cada diagrama se aportarán las clases mostrando los principales métodos y atributos. Para una mejor compresión de estas clases, se realiza una breve explicación sobre el módulo que representa cada una.

El primer diagrama mostrado en la Figura 3.16 define las clases encargadas de inicializar los motores de renderizado gráfico, físico, sonoro y el de despliegue de GUIs. Ademas, estas clases ofrecen soporte al resto de clases del proyecto.

La clase *Simulator* inicializa los motores de acción y los gestores de datos de entrada, así como también controla el funcionamiento interno llamando al bucle general de renderizacion que se ejecuta hasta la finalización de la tarea. *InputManager* es el gestor que almacena los datos de entrada que aporta el Terapeuta para modificar el comportamiento de la tarea, como el número de repeticiones de un movimiento, el tiempo, la lateralidad de la lesión, etc. *PhysicsManager* es la clase gestora que lleva a cabo la lectura de fichero de formas físicas correspondientes a las que puede utilizar el motor físico, para guardar en memoria instancias de la clase *Body*. Esta clase *Body* representa de manera global la parte gráfica y dinámica de los cuerpos colisionables de la escena, así como sus valores internos de comportamiento físico.

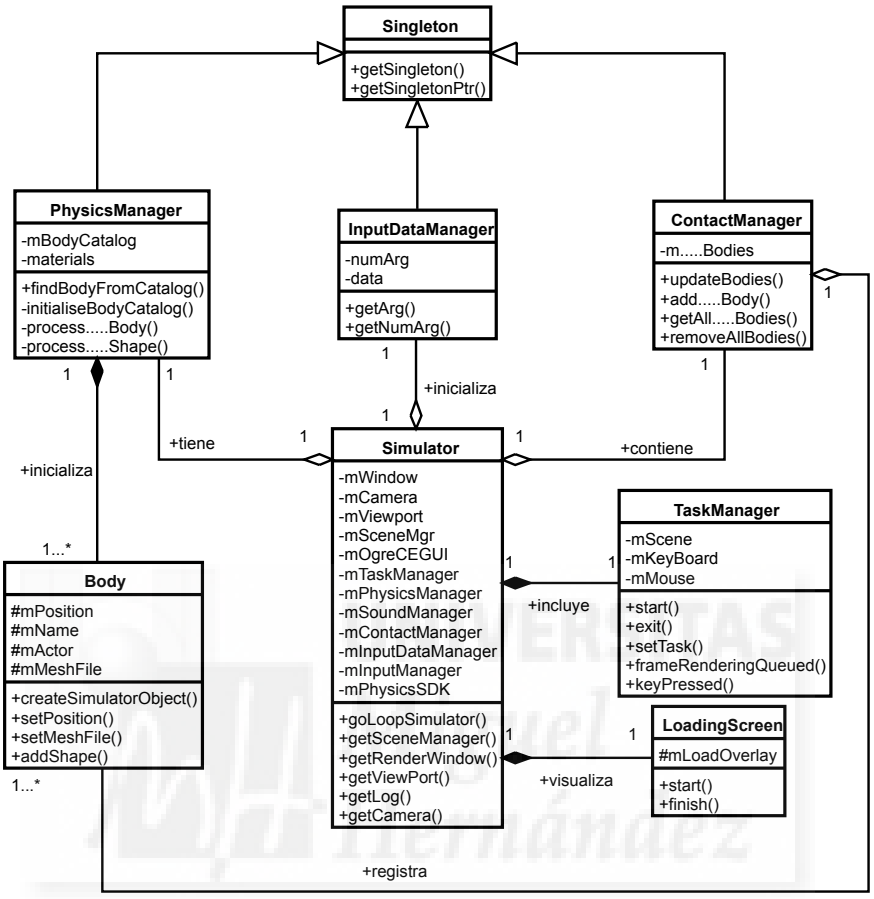


Figura 3.16: Diagrama de clases del núcleo central del sistema.

ContactManager se encarga de registrar todos los elementos colisionables que forman parte de la escena para indicar cuál de estos cuerpos tiene que ser actualizado en pantalla, realizando operaciones complementarias de colisión entre el avatar de control y el cuerpo en el caso de elementos como de cuerpos blandos, fluidos, articulaciones o tejidos. Todos estos gestores heredan de la clase *Singleton* para sólo tener una única instancia de estos objetos. Por otro lado, la clase *LoadingScreen* mostrará una pantalla de carga de recursos antes de iniciar la visualización del escenario virtual de manera que el Paciente no se encuentre de repente con todos los elementos de acción.

El siguiente diagrama (Figura 3.17) representa los posibles objetos colisionables que se pueden incorporar dentro de las escenas virtuales organi-

zados en una jerarquía de clases que heredan del objeto *Body*. Cada clase contiene los elementos necesarios para gestionar la creación de la parte gráfica y física de cada elemento, así como la parte encargada de actualizar su comportamiento o movimientos dentro del núcleo de ciclo principal.

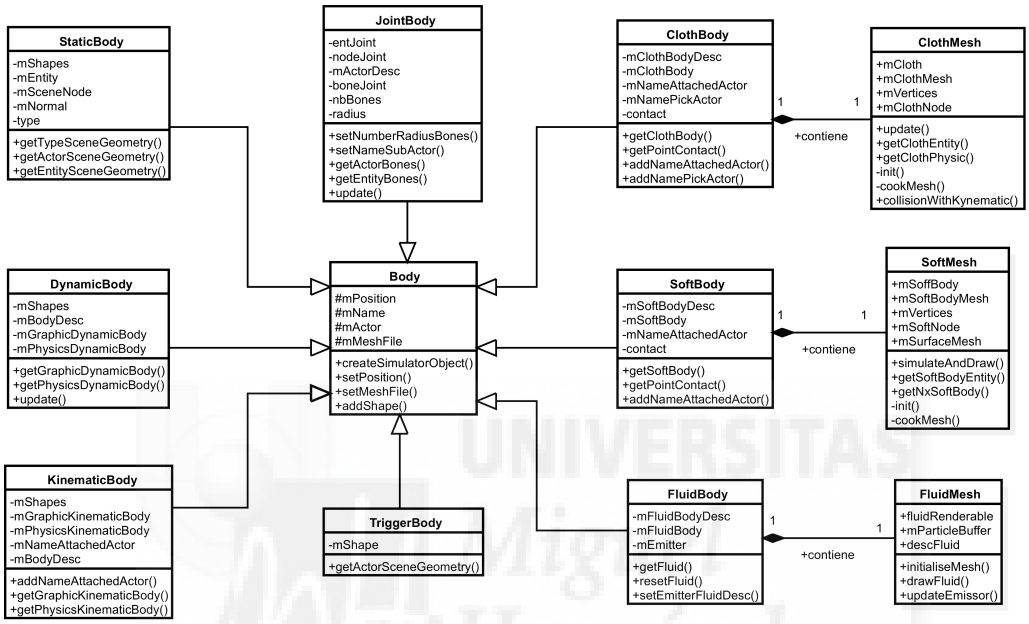


Figura 3.17: Diagrama de clases de los objetos virtuales con comportamiento físico.

En el diagrama de la Figura 3.18 se puede observar las clases relacionadas con el control del escenario junto con su organización y el avatar de control. También se encargan de gestionar la lógica de juego de la tarea para actualizar el estado de los distintos elementos del escenario y gestionar el tratamiento de los eventos externos como la pulsación del teclado o administrar los datos de entrada del dispositivo robótico. La clase *TaskDefatult* es el gestor que controla el estado interno de la tarea y hace de puente entre el usuario y el escenario representado por la clase *Scene*, la cual es el gestor de datos de escena que registra la organización de los elementos aportando la parte visual a los objetos *Body* creados con el *PhysicManager*. De esta manera, se crean las entidades visuales de la escena aportando el comportamiento físico con las formas colisionables, mientras que *KinematicManager* se encarga de modificar la posición y orientación del avatar de control formado por un cuerpo cinemático. En este bloque actúan los motores de renderizado visual, de colisiones y de sonidos. La clase *Task* modela

la parte general de la tarea envolviendo las clases principales que actuarán sobre *TaskDefault* y realiza todas las operaciones de la lógica de juego.

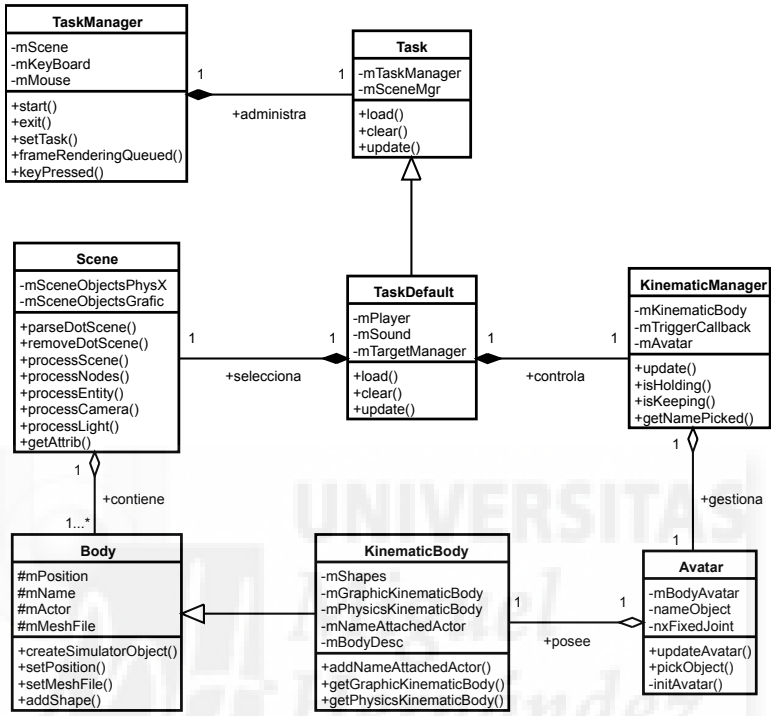


Figura 3.18: Diagrama de clases sobre la interacción y la organización de la escena.

El último diagrama (Figura 3.19) presenta las clases que se ocupan de administrar toda la gestión del cumplimiento de los objetivos, incorporando la clase *TargetManager* para leer los ficheros de objetivos y transformarlos en instancias de la clase *Target*. La clase *Target* controla el estado de cada objetivo. Como para cumplir los objetivos se necesitan realizar colisiones entre elementos de la escena, se añade una clase de captura de eventos llamada *TriggerCallback* que indica cuando un objeto se introduce dentro de un volumen establecido como elemento de interacción del objetivo mientras que la clase *Target* actualiza los marcadores con los elementos que tienen que actuar con el volumen de contacto. Para completar el diagrama se agrega la clase *ComSock* para comunicar la información del sistema de interacción robótico con el software a través de una conexión User Datagram Protocol (UDP), transmitiendo la posición y orientación del efector final al gestor de control cinemático del avatar, así como transferir el comando encargado de

indicar el cambio de nivel de dificultad debido a la variaciones ocasionadas por las señales fisiológicas para la adaptación de los elementos del entorno con las clases *TargetManager* y *TaskDefault*.

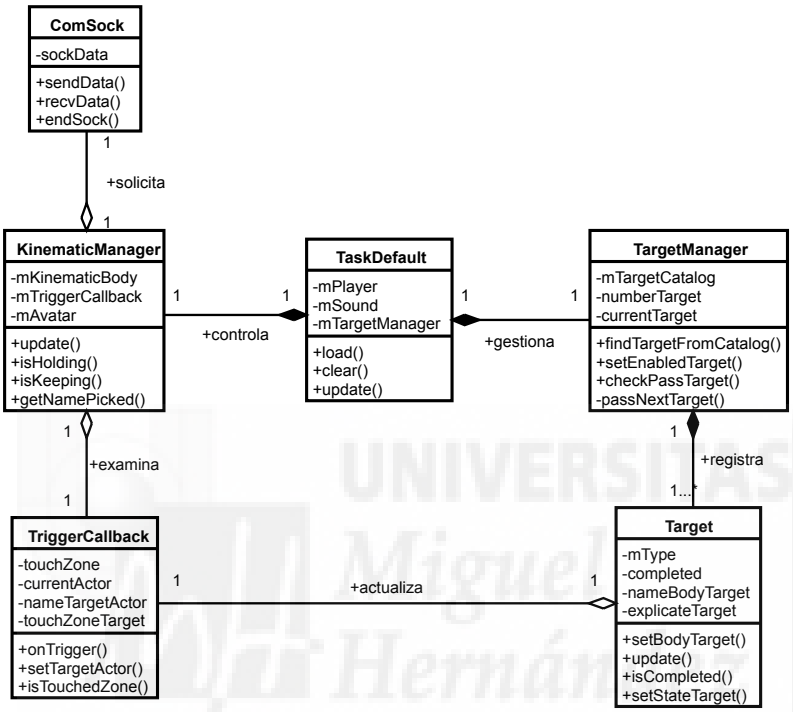


Figura 3.19: Diagrama de clases del cumplimiento de objetivos y comunicación UDP.

3.6. Fase de implementación

El proceso de implementación conlleva la elaboración de la arquitectura del sistema en términos de componentes como pueden ser ficheros de código fuente, *scripts*, ejecutables y similares. Se realiza después de las fases de análisis y diseño, siendo la que necesita de muchas horas de trabajo. Siempre existe la posibilidad de encontrar problemas que dificulten el desarrollo por lo tanto pueden provocar cambios en la estructura u organización de las clases establecidas en la fase de diseño. No obstante, en este apartado se van a comentar los puntos y funcionalidades más importantes del proceso de implementación del sistema de manera detallada, evitando la presentación

de código fuente y omitiendo las partes más sencillas para no extender de manera innecesaria el documento.

A lo largo del proceso de análisis y diseño han surgido dos líneas de desarrollo para implementar tareas destinadas a terapias virtuales de neurorehabilitación asistida por disposivos robóticos. Entonces, esta sección se va a dividir en dos apartados dónde se describen los detalles de implementación más significativos de las diferentes líneas: en la primera se detallan instrucciones para la creación de contenido multimedia y la otra trata sobre la generación de ficheros con rutinas de programación para el despliegue del motor de tareas.

3.6.1. Proceso de creación de contenido multimedia

El contenido que utiliza el motor de tareas incluye mallas de estructuras en 3D o 2D, materiales, texturas, animaciones, *scripts* de datos o sonidos. Todos estos elementos son conocidos también como recursos. La metodología está orientada a trabajar con herramientas *Open Source* para facilitar el acceso al proceso de creación de estas tareas.

3.6.1.1. Exportación de modelos y materiales

Antes de empezar a implementar el contenido visual del escenario de la tarea, hay que tener en cuenta que OGRE3D utiliza un formato de recursos propio y abierto de extension *.mesh* para optimizar el rendimiento a la hora de organizar las escenas. Debido a esto, aumenta la necesidad de utilizar una herramienta de modelado junto con un exportador compatible con OGRE3D.

En este caso, la herramienta elegida para diseñar todos los modelos visuales del escenario virtual de las tareas ha sido *Blender* (Blender, 2016). Esta herramienta permite modelar desde cero cualquier tipo de malla en 3D-2D o importar un modelo adquirido por Internet, y junto con el exportador llamado *io_export_ogreDotScene*(Blender2Ogre, 2016), proporciona los ficheros necesarios para cargarlos en el motor de renderizado. *Blender* también permite obtener un esqueleto asociado a los vértices de la malla para poder controlar la forma visual de un elemento en tiempo real durante la simulación o generar una serie de animaciones guardando la distribución temporal de los vértices en un *buffer* de *frames*. De esta manera, se pueden reestructurar las posiciones de los vértices durante cada *frame* de la simu-

lación a partir de una variable de tiempo. Además, con *Blender* se pueden definir los materiales y texturas que simulan las características superficiales.

El exportador de *Blender* se encarga de generar tres tipos de ficheros escritos en texto plano. El primer tipo es un fichero XML por cada malla implementada con *Blender* para definir la información del par posición-orientación de los vértices y las caras del objeto dentro de su espacio local. Además, hace referencia a los materiales y animaciones de dicho objeto. Sin embargo, para poder utilizar este tipo de ficheros con extension XML en el motor de renderizado es necesario obtener el formato binario *.mesh* ejecutando el *script OgreXMLConverter* (OgreXmlConverter, 2015). Este *script* es una herramienta de conversión por linea de comandos que transforma los ficheros XML en el formato binario nativo. El segundo tipo son los ficheros de extensión *.material* donde se detallan, a partir de una nomenclatura especial, las características de brillo, color, texturas y más opciones. El tercer tipo de recurso proporcionado por el exportador se trata del fichero de datos de escena con extension *.scene* que será comentando en el siguiente apartado.

3.6.1.2. Datos de escena

Los datos con la organización espacial de la escena se recogen en el *script* de datos XML de extension *.scene*, donde se define la posición, orientación y escalado de todos los elementos del escenario virtual, e incluso propiedades adicionales como el nombre de la malla, el nombre del nodo, la proyección de sombras, la configuración de luces de entorno, etc. Sin embargo, antes de exportar los elementos de mallas y materiales es necesario el establecimiento de una convención de nombres para tener una nomenclatura adecuada que permita el entendimiento entre el motor de tareas y este recurso definido, de manera que el sistema sepa en cada momento a qué se refiere cada elemento. La nomenclatura se estructura en diferentes grupos, dependiendo del comportamiento físico, del siguiente modo:

- **Escenario**: Este grupo se refiere a los elementos estáticos del escenario. La nomenclatura es *scene-nombre*. Si el elemento es un plano se nombra con *plane-nombre*. De esta manera se establece un nombre único a cada elemento.
- **Objetos**: Los elementos dinámicos con movimiento libre en función de las fuerzas generadas por el motor físico pertenecen a este grupo.

La nomenclatura es *actor-nombre*.

- **Avatar**: Para añadir un componente cinemático controlable por el usuario se realiza con la nomenclatura *avatar-nombre*.
- **Luces**: Se pueden establecer tres tipos diferentes de luces utilizando cualquier nombre.
- **Cámara**: Este elementos indispensable para capturar parte de la escena también se añade con cualquier nombre.
- **Elementos especiales**: Las funcionalidades del motor físico permiten la incorporación de elementos con un comportamiento dinámico especial como pueden ser fluidos, cuerpos blandos, articulaciones y tejidos. Las nomenclaturas son *fluid-nombre*, *soft-nombre*, *joint-nombre* y *cloth-nombre* respectivamente.
- **Volumenes**: Las zonas de la escena que serán objetivos de posición con volúmenes se definen con la nomenclatura *trigger-nombre*.
- **Elementos decorativos**: También se pueden añadir algunas funcionalidades que aporta el motor de renderizado como sistemas de partículas con *particleSystem-nombre* u opciones de cielo con *skyBox-nombre*, *skyDome-nombre* o *skyPlane-nombre*.
- **Geometría adicional**: Si alguno elementos no sigue los criterios establecidos, se considerará como una geometría no colisionable.

En la Figura 3.20 se muestra la representación gráfica del fichero de datos de escena como un árbol de posibles etiquetas para apreciar la estructura jerárquica correspondiente a la organización de una escena tipo que debe seguir la nomenclatura establecida anteriormente.

A partir del elemento raíz *scene* se puede definir dos tipos de objetos: con *environment* se establece el aspecto de la escena como el color de ambiente o el color de fondo, mientras que con la etiqueta *nodes* se crean los nodos encargados de estructurar el grafo de escena del escenario virtual. Cada nodo registra un objeto virtual definiendo la posición y la escala con atributos de triple tupla (x-y-z), y la rotación a partir de un cuaternio. Teniendo en cuenta el tipo de objeto, se puede elegir una de estas cuatro posibilidades: *light*, *camera*, *plane* o *entity*. Cada una posee diferentes atributos. La etiqueta *entity* tiene atributos para aportar el nombre del fichero de la malla

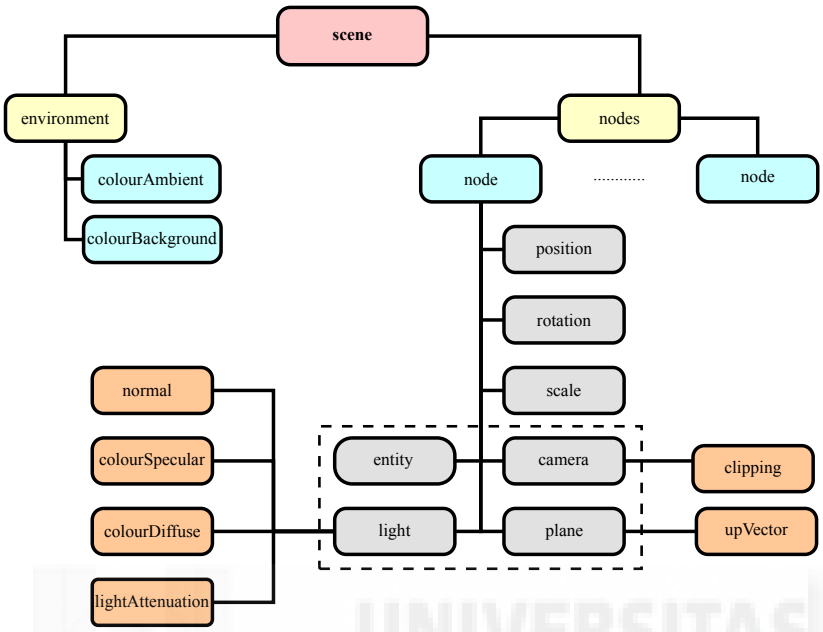


Figura 3.20: Árbol de etiquetas y atributos XML del fichero de datos de escena.

3D e indicar la posibilidad de proyectar sombras en la escena. La etiqueta *plane* posee atributos para configurar su tamaño, texturizado, e incluso para definir su material y proyectar sombras. *Camera* tiene parámetros para configurar la estructura interna del frustum de captura de la cámara virtual. Por último, la etiqueta *light* recoge datos sobre el tipo de luz, la intensidad de potencia, si permite la generación de sombras sobre los objetos visuales alcanzados por sus rayos o propiedades de color. La Figura 3.21 adjunta un fragmento de este tipo de fichero a modo de ejemplo.

3.6.1.3. Datos de cuerpos físicos

La generación de mallas poligonales con *Blender* permite tener una representación visual de la estructura de la escena. Por consiguiente, se pueden obtener mallas destinadas a la funcionalidad de detección de colisiones reduciendo los vértices de la malla visual o generando un conjunto de formas que en conjunto representan todas las partes del objeto. Este proceso se puede realizar con el escenario y con todos los objetos que necesiten detección de colisiones.

Los ficheros de datos de cuerpos físicos se crean siguiendo el mismo

```
<scene formatVersion="1.0.1" >
  <nodes >
    <node name="scene-sphere" >
      <position y="0.500000" x="4.000000" z="-4.000000" />
      <rotation qw="1.000000" qx="0.000000" qy="0.000000" qz="0.000000" />
      <scale y="1.000000" x="1.000000" z="1.000000" />
      <entity meshFile="scene-sphere.mesh" castShadows="True" />
    </node>
    ..............
    <node name="Camera" >
      <position x="0.000000" y="8.179058" z="10.502390" />
      <rotation  qw="0.852640" qx="0.522499" qy="0.000000" qz="-0.000000" />
      <scale x="1.000000" y="0.999996" z="0.999996" />
      <camera projectionType="perspective" fov="0.5033799435163735" >
        <clipping near="0.10000000149011612" far="100.0" />
      </camera>
    </node>
  </nodes>
  <environment >
    <colourAmbient r="0.0" g="0.0" b="0.0" />
    <colourBackground r="0.050876" g="0.050876" b="0.050876" />
  </environment>
</scene>
```

Figura 3.21: Fragmento de ejemplo de un fichero de datos de escena.

principio utilizado para los ficheros de datos de escena. Con *Blender* se modelan los objetos que representan las formas físicas superpuestas justo encima de los elementos visuales y el fichero de datos de cuerpos físicos recoge la posición, la orientación y la escala de cada una de esas formas. También agrupa valores adicionales de los parámetros de comportamiento físico de los elementos. En un mismo *script* se reúnen todas las formas colisionables de todos los elementos que componen el escenario de la tarea. De esta manera, durante la organización del escenario en el motor de tareas, el nombre del objeto indicado en el fichero de datos escena recupera el modelo colisionable dentro de este catálogo. En la Figura 3.22 se presentan las etiquetas a partir de un árbol jerárquico.

Este fichero de formato *.physics* tiene un elemento raíz llamado *bodies*. A partir de la etiqueta *body* se pueden referenciar nueve tipos de comportamiento físico: material, estático, dinámico, cinemático, fluido, cuerpo blando, tejido, articulación o volumen. Los objetos de tipo estático, dinámico y cinemático utilizan la etiqueta *shape* para registrar la posición, orientación y escalado de cada una de las formas que componen ese objeto. Por lo

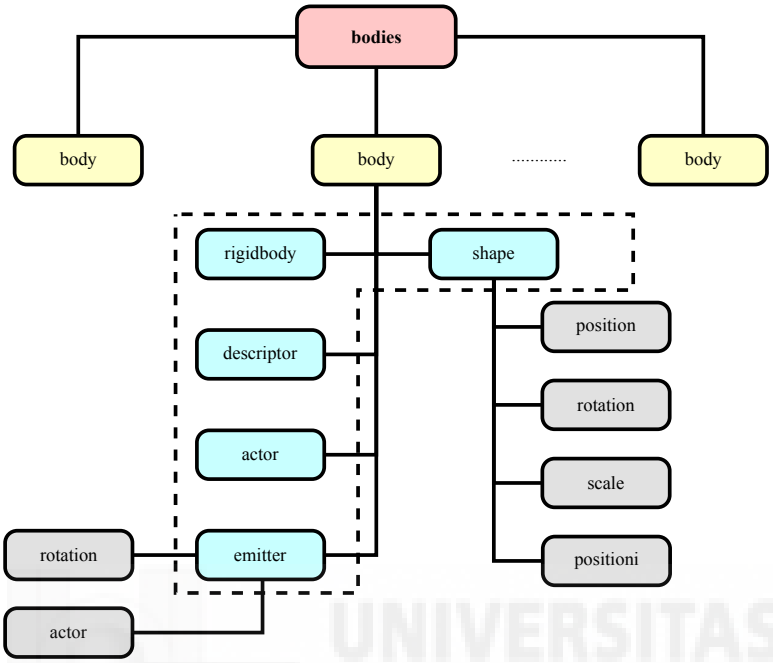


Figura 3.22: Árbol de etiquetas y atributos XML del fichero de formas físicas.

tanto, pueden tener tantas etiquetas *shape* como formas físicas posea. Adicionalmente, los cuerpos dinámicos, cinemáticos y articulaciones necesitan la etiqueta *rigidbody* para indicar datos como el peso o la masa. Los cuerpos blandos, tejidos y fluidos utilizan la etiqueta *description* para registrar los parámetros que caracterizan el comportamiento físico de estos elementos. También pueden utilizar la etiqueta *actor* para indicar si deben estar adjuntos a otro elemento. Los fluidos utilizan *emitter* si necesitan un punto de inicio para empezar su simulación. Sin embargo, existe una gran lista de atributos que se pueden utilizar con cada etiqueta según el tipo de objeto, por lo que se han omitido para no extender demasiado este apartado.

En conclusión, este fichero es un catálogo de modelos colisonables de cada uno de los elementos de la escena, cuyos nombres deben coincidir con los nombres definidos en el fichero de datos de escena. Entonces, cuando el gestor de creación de elementos de escena va incorporando los objetos, con un sólo nombre es capaz de recuperar los modelos de colisión y asignarlos a un malla visual. La Figura 3.23 adjunta un fragmento de este tipo de fichero a modo de ejemplo.

```
<bodies>
  <body type = "dynamic" name = "multicompounddd" >
    <rigidbody type = "custom" mass = "5"/>
    <shape type = "sphere" name = "mcd1">
      <position x="0.000000" y="6.500000" z="-5.500000" />
      <rotation qz="-0.000000" qx="0.000000" qy="0.000000" qw="1.000000" />
      <scale x="0.500000" y="0.500000" z="0.500000" />
      <positioni x="0.000000" y="6.500000" z="-5.500000"  />
    </shape>
    <shape type = "capsule" name = "mcd2">
      <position x="0.000000" y="5.500000" z="-5.500000" />
      <rotation qz="-0.000000" qx="0.000000" qy="0.000000" qw="1.000000" />
      <scale x="0.250000" y="0.500000" z="0.250000" />
      <positioni x="0.000000" y="6.500000" z="-5.500000"  />
    </shape>
    <shape type = "box" name = "mcd3">
      <position x="0.000000" y="4.500000" z="-5.500000" />
      <rotation qz="-0.000000" qx="0.000000" qy="0.000000" qw="1.000000" />
      <scale x="0.500000" y="0.500000" z="0.500000" />
      <positioni x="0.000000" y="6.500000" z="-5.500000"  />
    </shape>
  </body>
  ..............
  <body type = "soft" name = "soft">
    <descriptor volumeStiffness="1.0" stretchingStiffnes="1.0"
      dampingCoefficient="0.5" friction="0.5" particleRadius="0.025"
      solverIterations="5" density="1.0" attachmentResponseCoefficient="0.1"
      collisionResponseCoefficient="0.1" flagHardware="true" flagVisual="true"
      flagVolumeConser="true" flagColliTwoWay="true"/>
    <!--- Indicar si se tiene que adjuntar a un actor estatico o dinamico -->
    <actor name = "NULL"/>
  </body>
</bodies>
```

Figura 3.23: Fragmento de ejemplo de un fichero de datos de cuerpos físicos.

3.6.1.4. Datos de objetivos

Uno de los elementos clave para dar una funcionalidad de rehabilitación a las tareas es el fichero de datos de objetivos. En este punto, se tienen los recursos que representan la parte visual y física de la tarea, pero con sólo esos elementos no se pueden realizar un ejercicio completo, ya que sin objetivos las tareas carecen de utilidad. Por lo tanto, se debe generar otro fichero XML que recoge todos los objetivos que se pueden cumplir durante la realización de una tarea.

En primer lugar, se debe definir qué tipo de objetivos se pueden cumplir con el sistema de rehabilitación propuesto basándose en el mecanismo de efector final controlado por el usuario. Como el efector final sólo proporciona un desplazamiento sin opción de interacción con botones o algo por el estilo,

los objetivos se fundamentan en el hecho de alcanzar ciertos puntos en el espacio de trabajo. Traducido a los elementos del escenario virtual, los objetivos consistirán en:

1. Seleccionar un objeto dinámico con el avatar controlable.
2. Acercar el avatar controlable a cierto punto del escenario.
3. Depositar el objeto seleccionado en algún punto estático del escenario o encima de otro elemento dinámico.
4. Mantener el objeto seleccionado en algún punto del escenario o encima de otro elemento dinámico.

Todos estos objetivos están orientados de cara a la programación del gestor de objetivos, pero básicamente consisten en realizar trayectorias con el avatar controlable por encima de objetos, seleccionándolos o trasladándolos de unos puntos del escenario a otros. Por consiguiente, el fichero de formato *.targets* se tiene que implementar teniendo en cuenta el nombre de los elementos que se han utilizado en el fichero de datos de escena. Además, en este fichero también se añaden las especificaciones de los niveles de dificultad de la tarea pudiendo modificar cada atributo de manera independiente. En la Figura 3.24 aparece el árbol jerárquico de etiquetas que se han definido para implementar estos objetivos y los niveles.

El elemento raíz es *targets* y permite generar una lista de todo tipo de objetivos con la etiqueta *target*. Dependiendo del tipo de objetivo, se configuran los atributos de la etiqueta *object* para indicar los elementos que interactúan y como se deben comportar durante la realización del objetivo actual. Por otro lado, con la etiqueta *design* se selecciona el tipo de marcador que señalará el elemento de acción dentro del escenario virtual y si se debe ejecutar algún sonido o animación complementaria. Por último, la etiqueta *caption* recoge la información adicional a mostrar por la interfaz de información en caso de existir. En el caso de los niveles de dificultad con la etiqueta *level*, existe la posibilidad de asignar aspectos referentes a los tiempos de ejecución de las trayectorias con *time*. El atributo *number* modifica el número de elementos que pueden interactuar con el avatar y *mode* cambia algún aspecto del comportamiento de los elemento en función del tipo de tarea ejecutada. En la Figura 3.25 se muestra un fragmento de ejemplo de este tipo de ficheros.

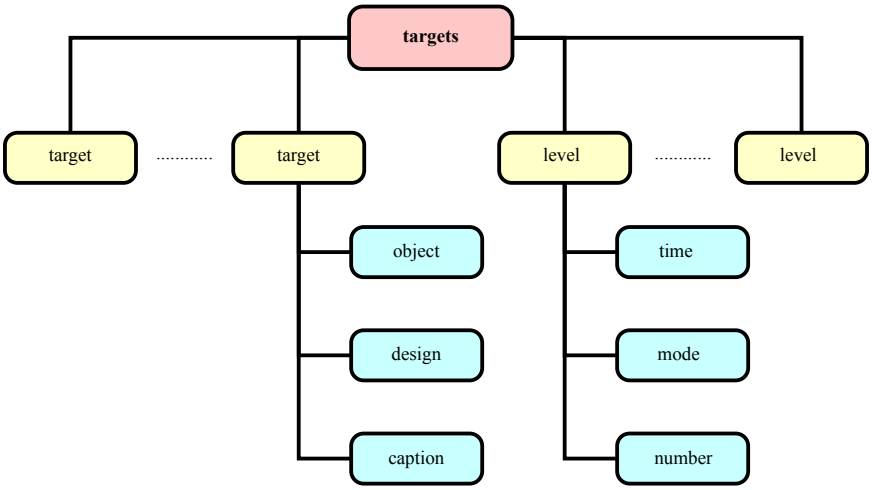


Figura 3.24: Árbol de etiquetas y atributos XML del fichero de objetivos.

```
<targets>
  <target type = "position" name = "target1" >
    <object name = "point1" target = "esfere" maintain = "true" time="3"  />
    <design name = "SystemTarget" dimx = "0.20" dimy = "0.20" dimz="0.10" pto = "false" >
    <caption tittle = "Acerca el objeto esférico al punto 1" />
  <target />
  ..............

  <target type = "marker" name = "target10" >
    <object name = "oilPUPArm" target="null" >
    <design name = "BillboardBlink" dimx = "0.15" dimy = "0.15" dimz="0.15" pto="false"/>
    <caption tittle = "4. Coge la aceitera" />
  </target>
  <level num = "1" name = "level01" >
    <time traj = "5" total = "25" >
    <mode type = "rand" />
    <number elem = "3" />
  </level>
</targets>
```

Figura 3.25: Fragmento de ejemplo de un fichero de datos de objetivos.

Como aporte adicional, se puede considerar que con el diseño de algún tipo de tarea en concreto existe la posibilidad de que surja la necesidad de realizar algunos objetivos no definidos o nuevos tipos de marcadores de elementos. O incluso que cuando se quiera modificar el nivel de dificultad se deban cambiar valores de la tarea que no están implementados en los actuales atributos. Entonces, es necesario realizar una actualización de esta parte del sistema para de complementar este tipo de fichero con esos nuevos fundamentos, hasta el punto de realizar nuevos bloques de código dentro del programa. Sin embargo, una vez realizado este proceso, las nuevas defi-

niciones extenderían el sistema para abarcar más tipos de tareas.

3.6.1.5. Creación de elementos complementarios

Además de todos los recursos comentados en apartados anteriores, se puede incorporar contenido adicional en forma de sonidos para aportar una realimentación sonora de los objetos de la escena o para indicar cuando se ha completado un objetivo. Existe una gran variedad de herramientas para grabar, editar, convertir, generar, grabar y mezclar audio de manera gratuita como *SoundEditor*, *Recording Audio* o *Audacity*, e incluso se pueden obtener desde sitios web con sonidos de libre distribución. Cualquier método es válido siempre que los sonidos tengan un formato *.ogg* o *.wav*.

Por otro lado, se puede añadir una GUI encargada de mostrar la información adicional que necesita el usuario para completar algún objetivo. El programa elegido para diseñar este tipo de componentes ha sido *CEED* (CEED, 2016), el cual es el editor oficial y optimizado para la librería gráfica CEGUI. Este programa se encarga de generar todos los ficheros necesarios para desplegar la GUI dentro del motor de tareas. Lo único a tener en cuenta a la hora de diseñar la interfaz es que debe tener un texto estático con la nomenclatura *infoStaticText*. Este elemento carga el texto almacenado en el fichero de objetivos, correspondiente al objetivo a completar por el usuario. Sin embargo, también puede tener otros elementos para mostrar otro tipo de información como datos de tiempo o aciertos. La posición de la GUI depende de la organización en pantalla de la tarea, de manera que se coloca en la parte de la pantalla donde no cubra elementos clave.

3.6.1.6. Directorios de recursos

El sistema desarrollado utiliza varios tipos de ficheros externos, en consecuencia se almacenan en la jerarquía de carpetas mostrada en la Figura 3.26. Cada carpeta contiene un tipo de recurso:

- interfaces: Directorio que contiene los ficheros de la GUI.
  - fonts: Guarda los ficheros de fuentes de caracteres.
  - imagesets: Almacena las imágenes utilizadas por la GUI y sus correspondientes ficheros de regiones llamados *Imagesets*.
  - layouts: Reúne los ficheros de organización en pantalla de la GUI.

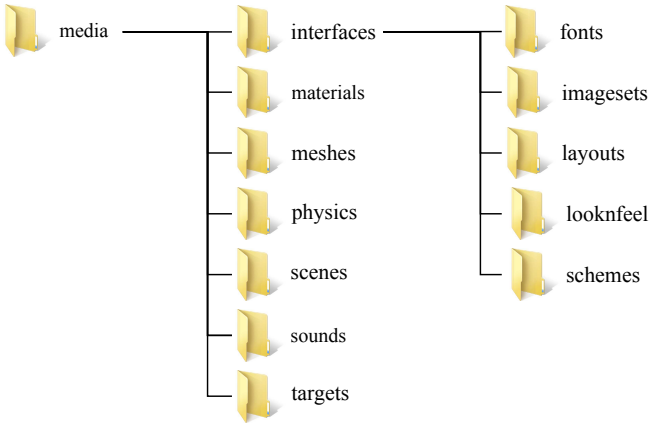


Figura 3.26: Directorio de carpetas para almacenar los recursos multimedia.

  - looknefeel: Carpeta utilizada para guardas los ficheros que definen el comportamiento visual de los elementos de la GUI.
  - schemes: Recolecta los ficheros con el aspecto visual de la GUI .
- materials: Alberga los materiales utilizados por las mallas.
- meshes: Carpeta que agrupa todos los ficheros de mallas visuales.
- physics: Almacena los ficheros con los catálogos de formas físicas.
- scenes: Se guardan los ficheros con la estructura de la escena.
- sounds: Carpeta donde se almacenan los ficheros de audio.
- targets: Lugar donde se recogen los ficheros con objetivos.

3.6.2. Implementación del motor de tareas

La segunda parte de la sección consiste en el desarrollo del motor de tareas. Durante la evolución de este apartado se detallan los aspectos más importantes a nivel de programación de cada una de los módulos del motor, a través de una serie de diagramas de flujo de proceso. Sin embargo, se ha evitado la inserción de código fuente en el documento para no extender demasiado el documento.

El punto inicial de la fase de implementación del motor de tareas consiste en crear un proyecto con una plantilla de aplicación de consola como raíz

del sistema aplicando una configuración inicial en blanco, lo que permite controlar todos los aspectos de construcción y compilación del proyecto. El proyecto se ha decidido generar con este formato de aplicación de consola para generalizar la entrada de datos al ejecutable desde el sistema de interacción con parámetros que modifiquen algunos de los aspectos internos de la tarea. De esta manera se evita la necesidad de cambiar el código fuente del programa al recibir parámetros relacionados con la repetición de los ejercicios o el nivel de asistencia del dispositivo robótico, ya que se necesita indicar diferentes valores para los diferentes pacientes con un distinto grado de lesión.

Con respecto a la parte de programación se ha implementado el cuerpo de las clases diseñadas en secciones anteriores para complementarlas con las clases proporcionadas por los SDK correspondientes al motor de renderizado gráfico, al motor físico y al motor de sonidos, ya que estas librerías proporcionan sus funcionalidades como una jerarquía de clases en C++. En las siguientes secciones se describen los diferentes módulos que forman parte del motor de tareas en una serie de diagramas de flujo separados para una mejor compresión.

3.6.2.1. Núcleo central de inicialización

El núcleo del motor de tareas está construido en torno a la clase *Simulator* y *TaskManager*. *Simulator* se encarga de declarar e inicializar los componentes principales del sistema tales como las librerías, los gestores, los elementos raíz y sus respectivos métodos de inicialización (Figura 3.27).

Básicamente todo el programa se fundamenta del motor de renderizado. Antes de poder utilizar OGRE3D se realiza una instancia de la clase raíz *Root*. Este objeto se crea utilizando el patrón *Factory*, para gestionar con un alto nivel todos los elementos principales del motor de renderizado, como son la creación de la cámara junto con su *viewport*, la carga de *plugins* para extender su funcionalidad, etc. En el siguiente paso se indican las rutas de los directorios de recursos donde el sistema buscará los elementos a cargar, visualizar en pantalla y utilizar durante la ejecución de la tarea. Justo después se crea la pantalla de visualización que contendrá todo el escenario virtual. Entonces, se inicializa el sistema de captura de pulsaciones del teclado con la librería *OIS* (pulsar y soltar tecla son los eventos que interesan en este proyecto). A continuación se comprueba si la tarea necesita alguna GUI complementaria para mostrar información adicional. Si necesita

este paso entonces carga todos los recursos necesarios.

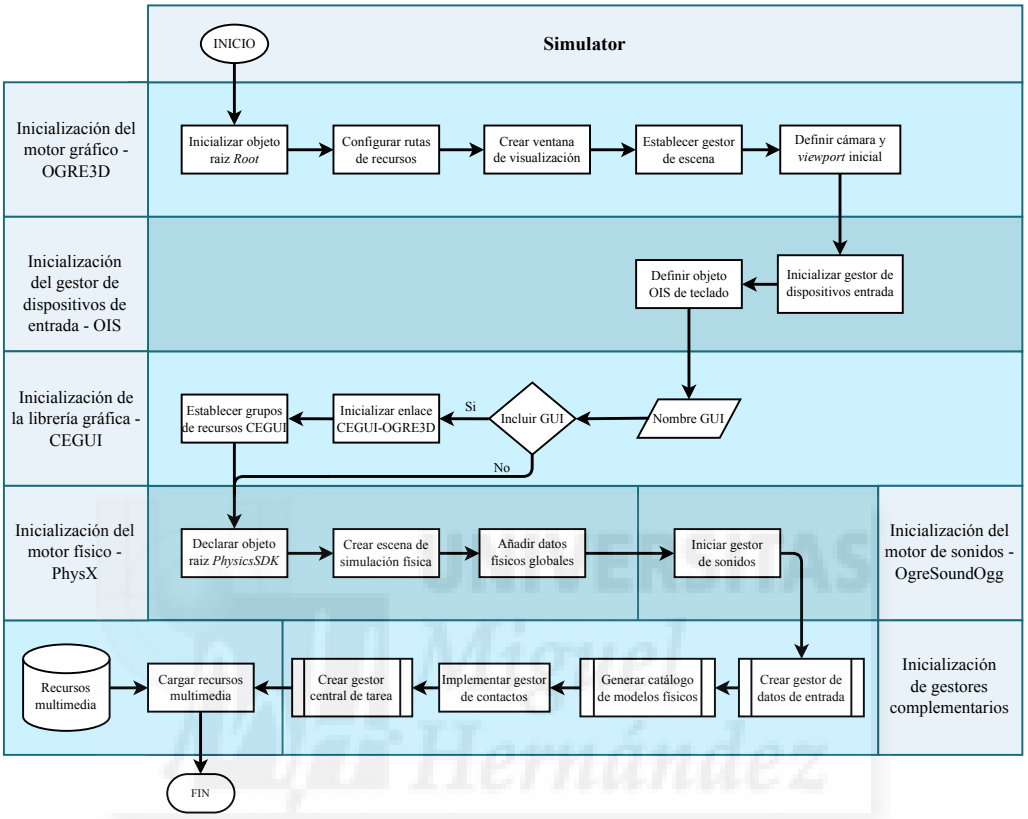


Figura 3.27: Diagrama de flujo de la inicialización general del motor de tareas.

Una vez realizada toda la inicialización de la parte visual, se pasa a la inicialización del motor físico a través de su clase raíz *PhysicsSDK* para generar el escenario de la simulación física y configurar valores físicos globales como la gravedad o las constantes de fricción estática y dinámica. La siguiente inicialización es el motor de sonidos para permitir la carga de sonidos cuando se establezcan los objetivos. Además, también se inicializa el resto de gestores que de momento no disponen de ningún tipo de información. Para finalizar, se cargan todos los recursos mostrando en pantalla una ventana de espera con una barra de progreso.

Por otro lado, *TaskManager* gestiona todo el proceso de *frames*, inicializando el bucle de renderizado que señala el ciclo de vida o simulación del programa. Además controla el estado interno de la tarea y registra los eventos de teclado y los eventos de ventana. En la Figura 3.28 se muestra

el proceso de inicialización del gestor de tareas. Esta clase sigue el patrón *Singleton* para tener una única instancia disponible desde varios módulos del sistema y hereda de las clases *FrameListener* y *KeyListener* del árbol jerárquico de OGRE3D. Las clases *Task* y *TaskDefault* se encargan de modelar el estado de simulación de la tarea haciendo de puente con el resto de clases que deben ejecutar su ordenes. Mas en concreto, *TaskDefault* ordena la carga el escenario y la GUI en caso de ser necesario, así como la actualización de los elementos de escena, gestiona el cumplimiento de los objetivos e incluso envía información sobre los resultados de rendimiento del paciente. La clase *Task* es virtual y *TaskDefault* hereda de ella.

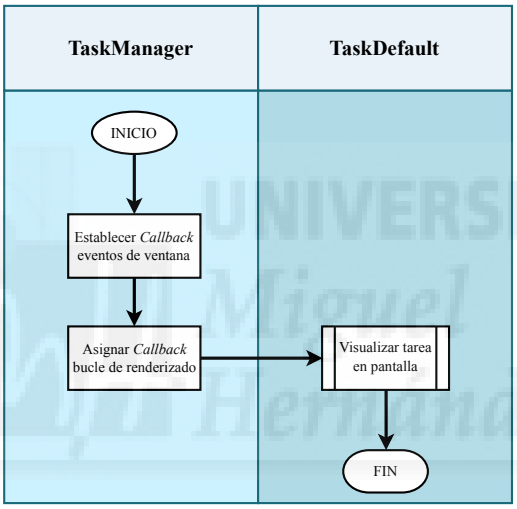


Figura 3.28: Diagrama de flujo de la inicialización del gestor de tareas.

3.6.2.2. Creación del escenario virtual

La implementación del escenario se divide en dos etapas. La primera etapa corresponde a la generación de la parte física. Cuando se inicializa el motor físico, se define un mundo físico simulado para asociar las formas colisionables con los elementos visuales representados en pantalla por el motor de renderizado. Por lo tanto, se puede decir que cada elemento del entorno virtual tiene una forma física que actúa dentro de la simulación física de *PhysX* y un nodo con una malla gestionada por OGRE3D. Esta asociación de nodos y formas físicas permite la visualización de las posibles colisiones generadas en el motor físico. Teniendo en cuenta esto, se ha implementado

la clase *Body* para encapsular la forma física, la malla y el nodo, y sus clases hijo actualiza estos elementos en función de su comportamiento físico. Cuando se inicializa el gestor de entrada de datos físicos (representado por la clase *PhysicsManager* en la Figura 3.29), se genera un catálogo de objetos *Body* con los cuerpos colisionables de cada uno de los elementos que forman parte del escenario.

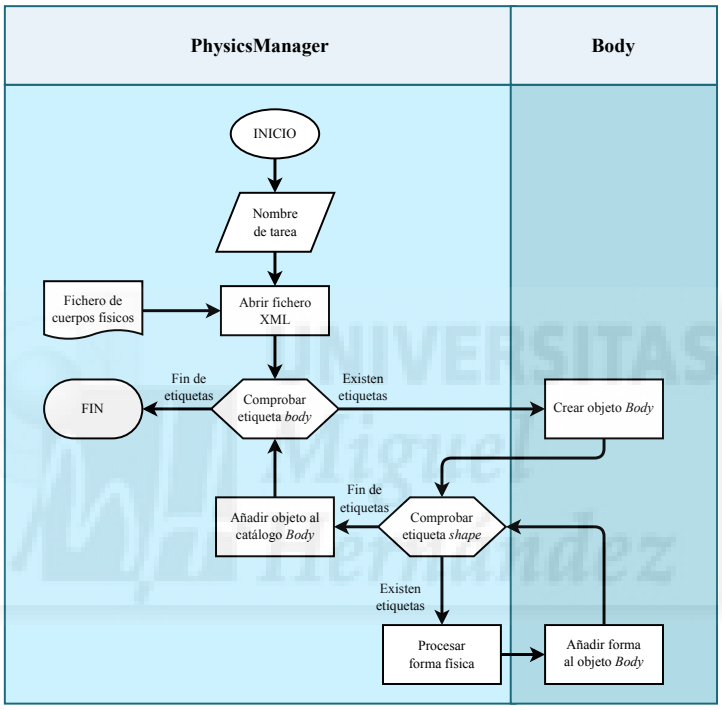


Figura 3.29: Diagrama de flujo de la creación de formas físicas.

PhysicsManager procesa el fichero XML de datos de cuerpos físicos a partir de una serie de métodos para recuperar los conjuntos de datos de las formas físicas definidas en este fichero. De esta forma, cuando se obtiene un conjunto de datos se procesan y se almacenan dentro de su correspondiente objeto *Body*. Sin embargo, esta etapa se realiza antes de inicializar la clase de control de la tarea y antes de la carga de recursos visuales junto a la pantalla de espera, debido a que la creación de formas físicas conlleva una gasto considerable de tiempo de carga al tener que efectuar labores de generación de mallas de simulación física. De esta manera se evita la aparición de un extraño efecto prolongado de espera antes de visualizar la tarea.

Una vez obtenido el catálogo de objetos *Body* con los cuerpos colisionables, se pasa a la etapa de generación del contenido visual utilizando las mallas visuales asociados a las formas físicas. Por consiguiente, desde la clase *TaskDefault* se crea una instancia de la clase *Scene* para completar esta etapa (Figura 3.30).

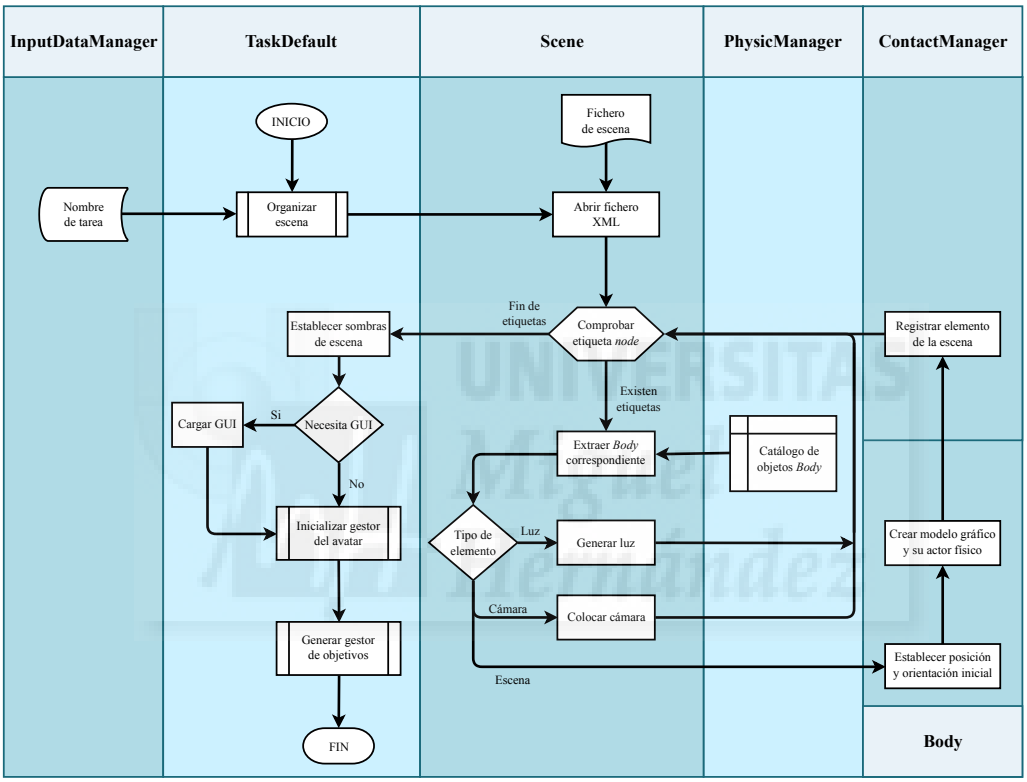


Figura 3.30: Diagrama de flujo de la organización visual del escenario virtual.

Scene es la clase encargada de procesar el fichero XML de datos de escena con la organización espacial de todos los elementos y almacenar un registro de los elementos creados. Cuando lee las líneas del fichero de datos de escena comprueba el tipo de elemento a partir las nomenclaturas explicadas en el apartado 3.6.1.2. Entonces busca el nombre dentro del catálogo de objetos *Body* para asignar la malla y representar visualmente el objeto en una posición inicial dentro de la escena. También se encarga de registrar los elementos dentro de *ContactManager* para la gestión de contactos.

3.6.2.3. Gestor de datos de entrada

Generalmente, los objetivos de todas las tareas son similares pero con diferentes elementos visuales: desplazar un avatar desde un punto a otro para llevar a cabo con éxito una determinada tarea interactuando con el entorno. Sin embargo, no todos los pacientes a los que está orientado este sistema tienen el mismo grado de lesión. Esto conlleva la necesidad de adaptar el número de repeticiones de movimientos o modificar otros valores de terapia sin necesidad de modificar el código fuente. Para solucionar este problema, se ha incorporado un gestor de datos de entrada representado por la clase *InputDataManager*, la cual se encarga de registrar todos los elementos para tener un acceso disponible desde varios módulos del sistema. Los datos de entrada que puede registrar son los siguientes, aunque en algunas de las tareas no hará falta utilizarlas todas:

- El nombre de la tarea a visualizar para tener una referencia del fichero de datos de escena, del fichero de datos físicos, del fichero de datos de objetivos y si se tiene que aplicar una interfaz.
- El número de repeticiones del movimiento.
- La pausa entre repeticiones.
- La lateralidad de la lesión del paciente para ajustar las posiciones de los objetos con respecto a la dirección de la trayectoria.
- El nivel de asistencia del dispositivo robótico.
- El tiempo para realizar una trayectoria o para completar la tarea.
- La amplitud del movimiento que determina el rango de movilidad del usuario.

3.6.2.4. Creación del avatar de interacción

El motor de tareas se ha diseño de manera que el control del avatar dentro del escenario se realiza con un dispositivo robótico. Por esto, se ha implementado la clase *ComSock* con el objetivo crear una comunicación vía socket con el sistema de gestión del dispositivo robótico. Principalmente se intercambian los datos de posición y rotación del efector final al motor de tareas. Otro dato recibido es el comando que indica el cambio del nivel

de dificultad. También se intercambian datos desde el motor al sistema de interacción como posiciones objetivo para la asistencia cuando el paciente necesita llegar a un punto y no consigue alcanzarlo, e información sobre los resultados de rendimiento del usuario al finalizar la tarea. Con los datos de posición y orientacion, la clase *KinematicManager* gestiona el desplazamiento del avatar, la asistencia y las características de sujeción de los objetos decidiendo si selecciona o suelta algún elemento, junto con la clase *Avatar* (Figura 3.31). Más específicamente, la clase hijo de *Body* llamada *KinematicBody* se encarga de la gestión directa de cambio de posición y orientación del avatar.

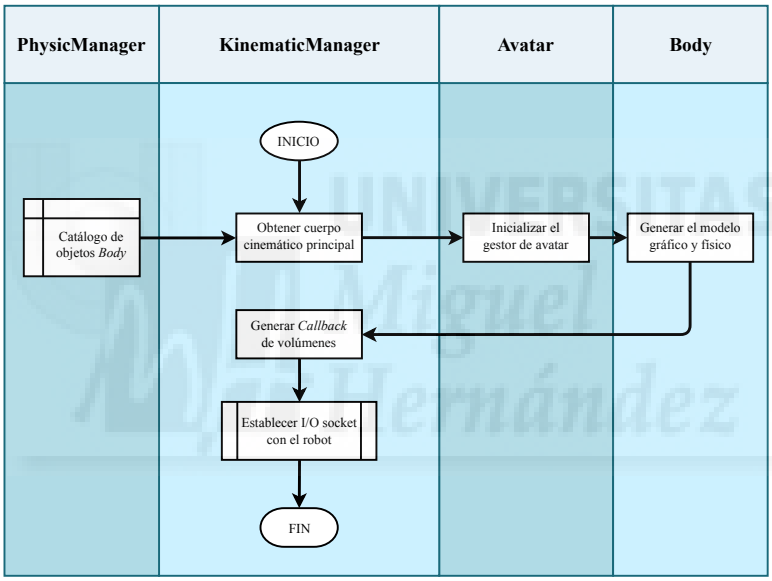


Figura 3.31: Diagrama de flujo de la inicialización del avatar.

Cuando se selecciona el objeto cinemático del catálogo de cuerpos *Body*, se crea el modelo gráfico y el modelo físico del avatar, y justo dentro de *KinematicManager* se implementan las comunicaciones sockets necesarias para intercambiar los datos así como inicializar el *Callback* de volúmenes.

3.6.2.5. Gestión y creación de objetivos

El control de gestión y la creación de los objetivos se lleva a cabo con las clases *TargetManager* y *Target* (Figura 3.32). *TargetManager* es otro

gestor de lectura de ficheros XML pero en este caso se encarga de los datos de objetivos junto con los datos de niveles de dificultad.

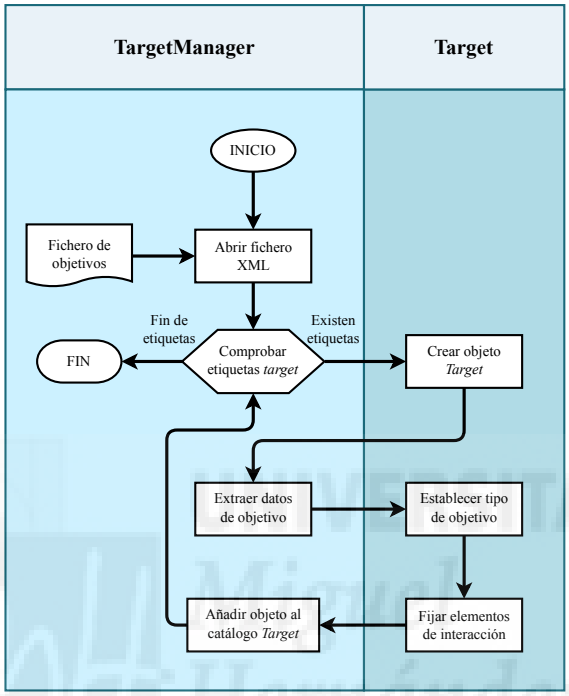


Figura 3.32: Diagrama de flujo de la creación del catálogo de objetivos.

De la misma manera que los gestores de escena y físicas procesaban sus ficheros, *TargetManager* utiliza métodos similares para generar una instancia del objeto *Target* por cada objetivo definido en el fichero. La clase *Target* se encarga de recuperar toda la información sobre los elementos que participan en un objetivo concreto. De esta manera, establece los criterios a seguir por el Paciente para completar los objetivos. A parte de generar los objetos que encapsulan los objetivos de la tarea, se encarga de comprobar el estado actual del objetivo y decide si se ha cumplido las condiciones necesarias para pasar al siguiente objetivo. Además, controla el *TriggerCallback* indicando el nombre de los elementos que deben interactuar en cada momento. Por otro lado, estas clases gestionan la reproducción de sonidos o la ejecución de una determinada animación al cumplir algún objetivo. Además estas clases examinan también los requisitos de control de la tarea a partir de la clase *InputDataManager*.

3.6.2.6. Actualización de la escena virtual

TaskManager es la clase encarga de ejecutar el ciclo de renderizado de la tarea, sin embargo *TaskDefault* manda la orden de actualización de los objetos *Body* a partir de la clase *ContactManager*. *ContactManager* tiene una lista de todos los objetos que deben actualizar su posición y orientación en función de la detección de colisiones dentro de la simulación física. En la Figura 3.33 se muestra el diagrama de flujo con el proceso general de actualización de la escena, donde el ciclo de simulación se ejecuta mientras existan objetivos por cumplir.

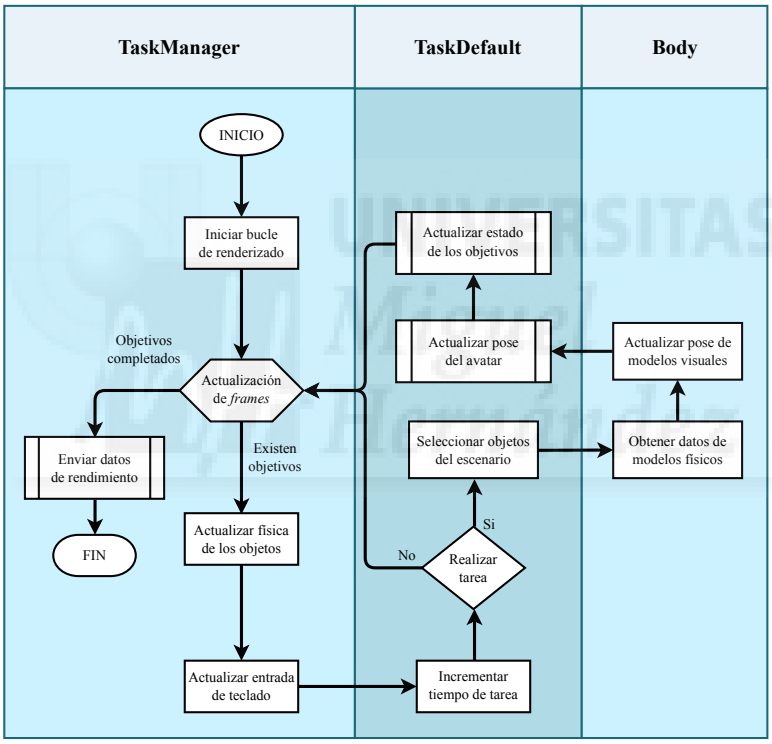


Figura 3.33: Diagrama de flujo del proceso de actualización general.

En cada *frame* se actualiza la simulación física y los objetos *Body* modifican la pose de los todos los elementos gráficos. El cuerpo cinemático también influye en el comportamiento físico de los otros elementos dinámicos al ejercer fuerzas. Por lo tanto, este cuerpo controlable también necesita otro proceso de actualización para traducir el movimiento del dispositivo robótico en un desplazamiento del avatar dentro del entorno virtual. La

clase *KinematicManager* se encarga de todo este proceso de actualización (Figura 3.34) del cuerpo cinemático, comprobando el objetivo actual para gestionar la actividad de seleccionar o depositar objetos a partir de la clase *TriggerCallback*. También realiza la conexión con el sistema de interacción para compartir información con el dispositivo robótico, como los datos de posición del efector final o asignar el nivel de asistencia.

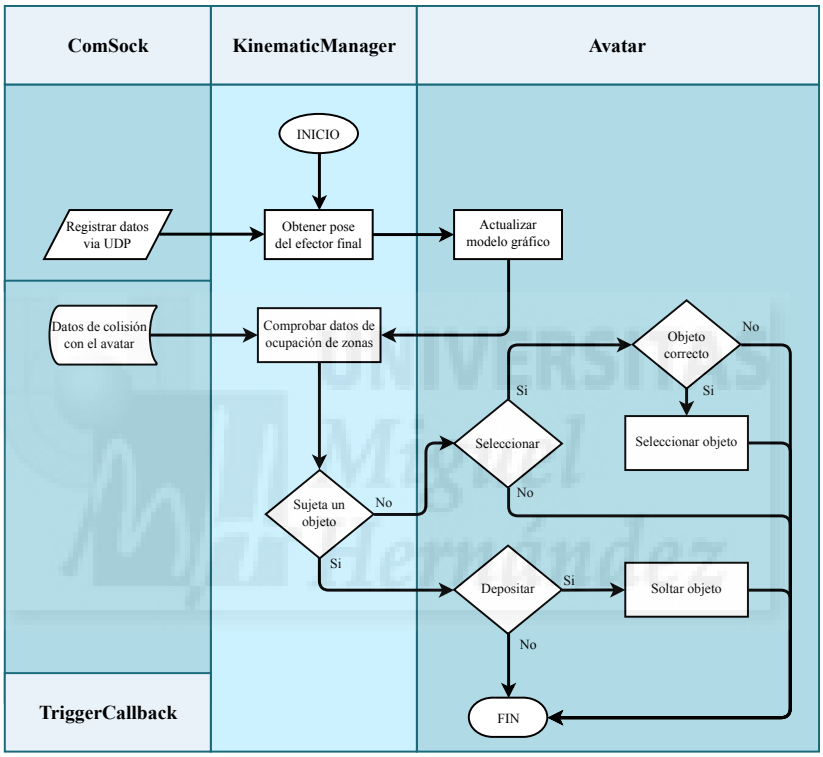


Figura 3.34: Diagrama de flujo del proceso de actualización del avatar.

Para finalizar, se realiza la actualización del estado de los objetivos (Figura 3.35). Una vez actualizado el movimiento en pantalla del avatar, se comprueba el estado del objetivo actual en cada iteración del bucle de renderizado. Entonces, se analizan los elementos que participan para verificar si se cumple algunas de las condiciones específicas. Cuando se cumple alguno de los objetivos se pasa al siguiente estableciendo los nuevos datos de ocupación de zonas en *TriggerCallback*. En este punto, se actualiza la GUI en caso de ser necesario. Cada cierto tiempo se crea una conexión con el módulo de clasificación del sistema de interacción para recibir los coman-

dos que indican la modificación del nivel de dificultad dependiendo de las señales fisiológicas del Paciente. Entre *TargetManager* y *TaskManager* se realiza este proceso.

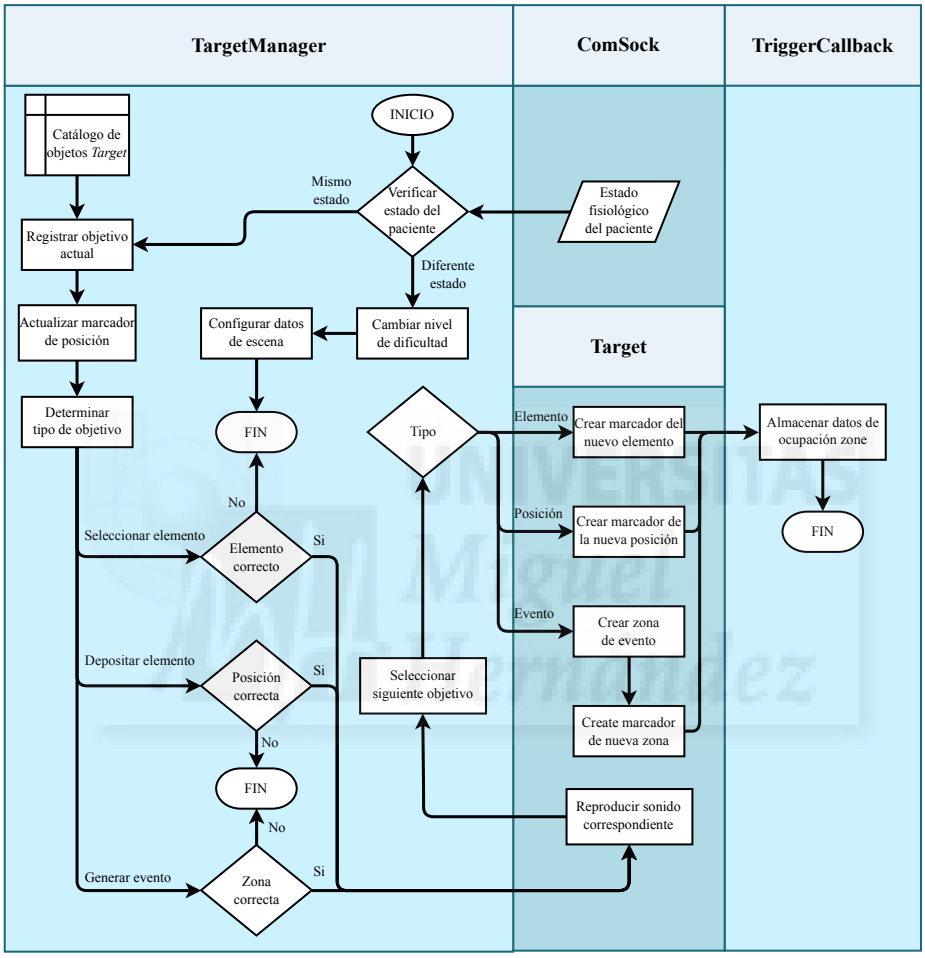


Figura 3.35: Diagrama de flujo del proceso de actualización del estado de los objetivos.

3.7. Pruebas de integración

Después de tener en cuenta los requisitos iniciales, el proceso de análisis, diseño e implementación, el siguiente paso consiste en verificar el funcionamiento de los componentes desarrollados para comprobar si son capaces

de interactuar correctamente entre ellos. Con este propósito se ha implementado una serie de tareas virtuales de ejemplo para examinar la eficacia del sistema software. Durante el proceso de implementación se han probado todos los módulos prestando especial atención a todas las posibles bifurcaciones propensas a fallos. En esta sección se van presentar seis diferentes tareas, cuya complejidad va aumentando conforme se han ido diseñando. Esta complejidad viene dada por la cantidad de módulos que actúan durante la ejecución de la tarea, simulando diferentes cantidades de elementos virtuales y objetivos. Todas estas tareas cumplen la premisa que debe seguir este tipo de tareas destinadas a personas con ACV: los entornos virtuales debe ser sencillos y claros para que los pacientes puedan comprender fácil y rápidamente los objetivos de la tarea. Con este proceso se comprueba la potencialidad que puede tener este sistema para desarrollar tareas virtuales cuyo nivel de dificultad es auto-adaptable.

3.7.1. Actividad motora para seleccionar vasos

La primera tarea es una actividad simple basada en AVD con pocos elementos. En este caso el escenario está formado por dos elementos estáticos representados por una mesa y un posavasos, un elemento dinámico simbolizado por un vaso y un elemento cinemático de control configurado por un brazo virtual. La Figura 3.36 muestra una captura de pantalla del entorno.

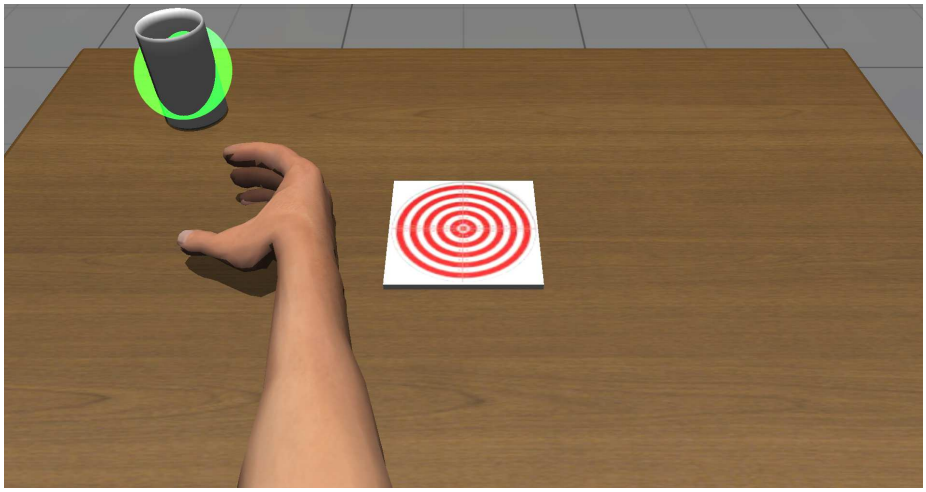


Figura 3.36: Entorno virtual de la tarea de seleccionar vasos.

El objetivo principal consiste en simular la acción de sujeción y depósito de un vaso. Como objetivos particulares se han establecido dos acciones:

1. Seleccionar el vaso: El vaso se sujeta con la palma de la mano del brazo virtual. Si este paso no se realiza correctamente, el vaso puede llegar a tambalear o volcar debido a las fuerzas simuladas por el motor físico en caso de colisión inesperada con alguna parte del brazo.

2. Depositar el vaso: Una vez realizado el primer objetivo, lo siguiente consiste en depositar el vaso justo encima del posavasos. Para evitar una imprevista caída del vaso cuando colisiona con el borde del posavasos, se debe dejar en el centro.

Al iniciar la tarea, el vaso aparece encima de la mesa en un posición aleatoria. El Paciente tiene que seleccionar el vaso y dejarlo en el posavasos. Esta rutina se puede realizar el número de veces que el Terapeuta considere adecuado. Cada vez que se deja el vaso, aparece en otra posición distinta a la anterior. Para obtener una mejor percepción de la situación del brazo dentro de la escena durante la realización de la tarea, se utiliza la generación de sombras proporcionada por el motor gráfico. De esta forma se pueden llegar a cumplir los objetivos con una mejor precisión. El nivel de dificultad se adapta modificando el tiempo que tiene el Paciente para completar las dos acciones. Con esta tarea sencilla se prueban las funcionalidades más básicas del sistema correspondientes a la lectura y procesamiento de los ficheros de recursos XML y la simulación de unos objetivos simples.

3.7.2. Actividad de colocación de piezas

La siguiente tarea pretende simular el ejercicio de colocar piezas de diferentes formas en sus correspondientes huecos dentro de un tablero. Por tanto, el escenario contiene un tablero estático y piezas dinámicas con diferentes formas: una estrella, un cono, una capsula, una esfera, un anillo, un cubo, un cilindro, una letra y una cruz. Para seleccionar las piezas se incorpora un avatar representado por una flecha. En total existen nueve diferentes piezas, sin embargo, el tablero está constituido por seis huecos donde se colocan los porta-piezas para indicar cuál de las piezas se debe poner en el correspondiente hueco. De esta manera, el usuario deberá seleccionar sólo las piezas que se muestran en pantalla. En la Figura 3.37 se muestra el entorno virtual.

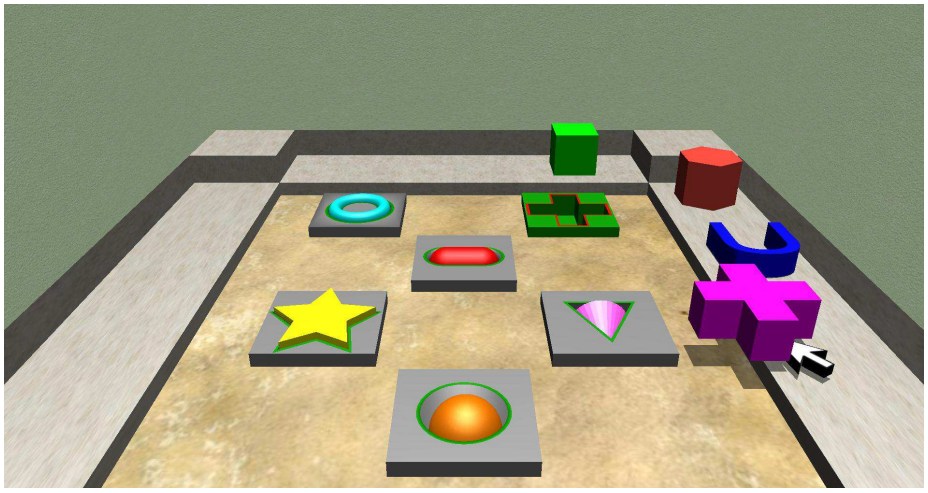


Figura 3.37: Entorno virtual de la tarea de colocar piezas.

Como se ha indicado anteriormente, el objetivo principal de la tarea es seleccionar las piezas y colocarlas en los correspondientes huecos. Cada vez que se inicia la tarea, los porta-piezas se distribuyen por el tablero de manera aleatoria para que el usuario no deba colocar siempre las mismas piezas en los mismos lugares. De esta forma, no se pueden interaccionar con tres piezas que permanecerá en sus posiciones iniciales. Por consiguiente, el fichero de objetivos tiene que contener 18 posibles acciones: nueve de los objetivos consisten en la posibilidad de seleccionar todas las piezas y los nueve restantes son para dejar las piezas en los porta-piezas. Sin embargo, el sistema es capaz de seleccionar los objetivos correspondientes a los porta-piezas mostrados en pantalla a partir de sus posiciones en el escenario. Con esta tarea se prueban las siguientes funcionalidades del sistema:

- Los módulos básicos de lectura de ficheros XML y almacenamiento de los diferentes elementos del escenario.
- Sujeción y depósito de elementos dentro de los lugares asignados.
- Comprobar sistema de señalización de los elementos que deben interactuar durante el objetivo, mostrando un anillo de partículas encima de la posición de la pieza o la posición donde se debe dejar.
- Selección de objetivos a partir de los elementos mostrados en escena.
- La reproducción de sonidos al completar objetivos.

3.7.3. Actividad de golpear elementos situados al azar

La siguiente tarea virtual consiste en localizar elementos que aparecen aleatoriamente en pantalla con el objetivo de golpearlos con el avatar de control. El escenario se ha diseñado de manera que representa un jardín formado por cuerpos estáticos como son una valla de madera, césped y nueve madrigueras. Incluso se han añadido elementos decorativos como herramientas de jardinería para ofrecer al paciente un mayor grado de realismo dentro del entorno virtual. El cuerpo cinemático en este caso es un martillo. Durante la ejecución de la tarea aparecen conejos por las madrigueras. La Figura 3.38 muestra el escenario virtual de la tarea.

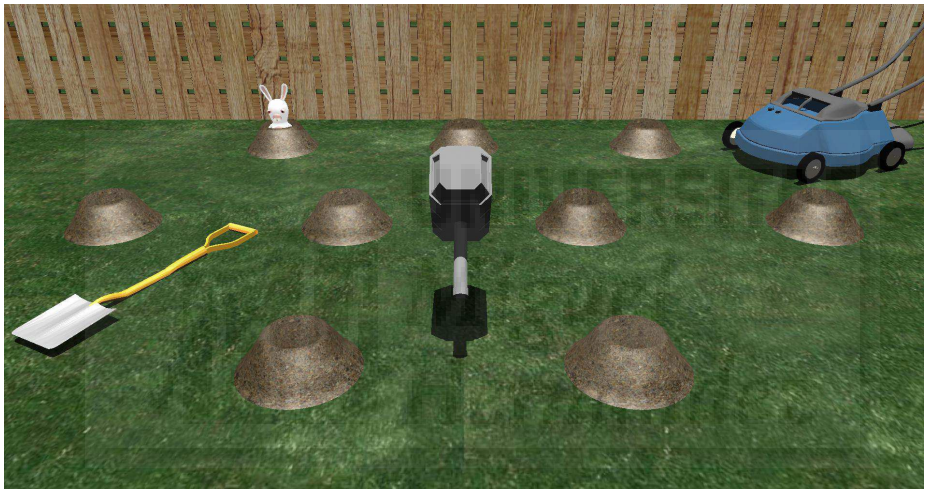


Figura 3.38: Entorno virtual de la tarea de golpear elementos.

El objetivo principal de la tarea es golpear a los conejos con el martillo, los cuales aparecen aleatoriamente en cualquiera de las nueve madrigueras del escenario. Cada vez que el usuario acerca el martillo a un objetivo, se ha programado una simulación de golpeo con su correspondiente sonido. El conejo se ha diseñado como un cuerpo blando, por lo tanto, su topología elástica se deformará al entrar en contacto con el martillo teniendo en cuenta la dirección de las fuerzas recibidas. Como dicho elemento permanece un tiempo limitado en la pantalla, el usuario tiene la posibilidad de golpearlo para tener éxito o si el elemento desaparece se considerará un error. En esta tarea, el fichero de datos de objetivos sólo tiene un objetivo y el nivel de dificultad viene determinado por el tiempo total que permanece el conejo en pantalla y la cantidad de veces que se repite el objetivo, siempre cumpliendo

una aleatoriedad a la hora de mostrar el elemento de contacto. Con esta tarea se prueban las siguientes funcionalidades del sistema:

- Los módulos básicos de lectura de ficheros XML y almacenamiento de los elementos de la escena, tanto partes gráficas como físicas.
- El módulo de datos de entrada para almacenar valores de terapia.
- Ejecutar simulaciones al cumplir objetivos.
- La funcionalidad de repetición de tareas de manera aleatoria.
- La reproducción de sonidos al completar o fallar algún objetivo.
- La capacidad de mantener el objetivo un tiempo limitado, evitando que permanezca activo durante toda la simulación.

3.7.4. Activad de selección de cajas

Esta tarea simula un fábrica de cajas con vista en perspectiva donde la escena converge al punto central de la pantalla. El escenario gráfico estático consiste en ocho plataformas y un depósito central. Las ocho plataformas están colocadas uniformemente alrededor del depósito, siempre manteniendo la misma distancia. Todos los elementos estáticos son soportes para el elemento dinámico representado por una caja, la cual interactúa con la herramienta virtual cinemática definida por una llave inglesa. Además se han añadido unos elementos decorativos. En la Figura 3.39 se puede observar una captura de pantalla de la tarea.

La estructura del escenario permite realizar dos objetivos principales: coger la caja situada en cualquiera de las ocho plataformas y soltarla dentro del depósito central. Por consiguiente, el fichero de objetivos debe estar formados por dos bloques de etiquetas XML. La caja puede aparecer en escena de tres maneras diferentes sobre las ocho plataformas: i) secuencialmente en las agujas del reloj; ii) en contra de las agujas del reloj; iii) aleatoriamente. Entonces, es necesario aportar la cantidad de veces que se repite el objetivo y el tiempo limite para cumplir el movimiento. De este modo, cuando la caja aparece en cualquiera de las ocho plataformas se avisa cambiando el color del soporte. Al recoger la caja con el elemento cinemático, el depósito central será el que cambia de color a modo de señalización. Disponer de un tiempo máximo para completar los objetivos permite la posibilidad de

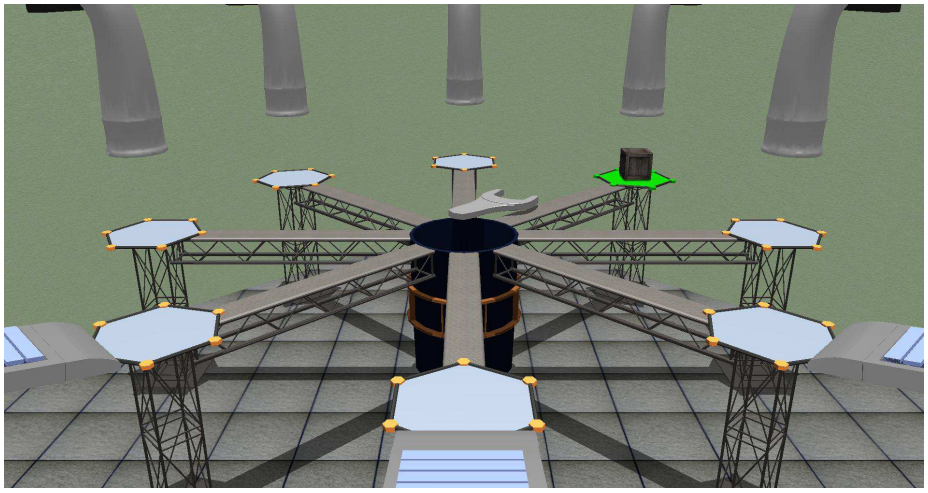


Figura 3.39: Entorno virtual de la tarea de seleccionar cajas.

obtener un fallo. Por tanto, el sistema puede realizar un seguimiento del rendimiento de éxito del usuario. Con esta tarea se prueban las siguientes funcionalidades del sistema:

- Los módulos básicos de lectura de ficheros XML y almacenamiento de los elementos de la escena.
- El módulo de captura de datos de entrada para almacenar valores de terapia.
- La funcionalidad de repetición de tareas de manera aleatoria y secuencial.
- La reproducción de sonidos al completar o fallar algún objetivo.
- La capacidad de mantener la posición objetivo un tiempo limitado. En caso de terminar este tiempo, se pasa al siguiente objetivo mostrando la caja en otra posición.

3.7.5. Actividad para servir líquidos

En esta prueba del sistema se ha implementado una tarea con más elementos de interacción para completar más objetivos que en las pruebas

anteriores. La tarea consiste en simular el trabajo de un camarero encargado de proporcionar diferentes tipos de bebidas en función del pedido de los clientes. Por tanto, el escenario estático gráfico está formado por una mesa, un dispensador de bebidas, un dispensador de vasos y tres posavasos con diferentes colores para indicar el tipo de bebida. Como elemento decorativo sin funcionalidad física se ha añadido una pila de vasos encima de una toalla de cocina. El avatar de control es un brazo virtual, el cual interactuará con un vaso que tiene comportamiento dinámico. En esta tarea también se ha añadido un brazo extra con una bandeja a modo de objetivo final a la hora de depositar el vaso. En la Figura 3.39 se visualiza el escenario gráfico de la tarea implementada.

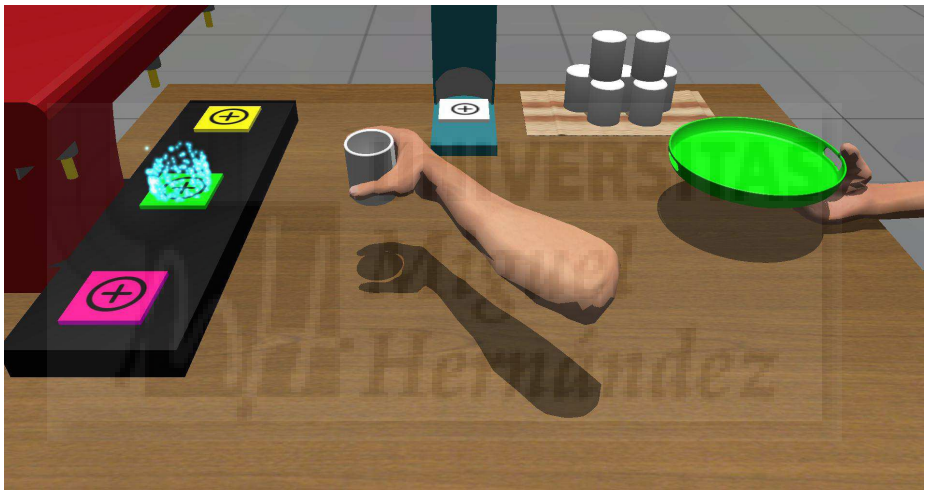


Figura 3.40: Entorno virtual de la tarea de servir bebidas.

La finalidad principal de la tarea consiste en coger un vaso, rellenarlo con un líquido y depositarlo encima de una bandeja. Esto da como consecuencia la incorporación de tres objetivos concretos dentro del fichero de objetivos. Al empezar la tarea, el usuario espera la aparición de la bandeja encargada de indicar el color de la bebida a servir. Entonces el usuario acerca el brazo al vaso situado en el dispensador de vasos para seleccionarlo y acercarlo al dispensador de bebidas justo en la posición del posavasos cuyo color coincide con el color de la bandeja. Para aumentar un poco más el realismo a la hora de añadir el líquido se ha utilizado la funcionalidad de fluidos del motor físico. De esta manera, cuando el vaso permanece en la zona superior al correspondiente posavasos se vierte un torrente de líquido

para que el usuario permanezca un tiempo en la misma posición hasta el llenado del vaso. Una vez finalizado este objetivo, el siguiente consistirá en depositar el vaso encima de la bandeja. Estas tres acciones pueden ser repetidas dependiendo de los criterios del Terapeuta. Cada vez que aparece la bandeja, su color es aleatorio. En este caso los objetivos son secuenciales y el rendimiento de la tarea se mide en función de la cantidad de tiempo que el usuario ha necesitado para completar las tres acciones un serie de veces. Cuando se repite la tarea, el sistema vuelve a mostrar los mismos objetivos concretos pero modificando la posición donde se llena el vaso. Con esta tarea se prueban las siguientes funcionalidades del sistema:

- Los módulos básicos de lectura de ficheros XML y el módulo de almacenaje de elementos de escena.
- El módulo de datos de entrada para almacenar valores de terapia.
- Repetición de los objetivos establecidos en el fichero de objetivos en función de la cantidad de repeticiones establecidas.
- Ejecutar simulaciones al iniciar objetivos.
- La funcionalidad aleatoria de elegir entre una serie de diferentes elementos para establecer condiciones del objetivo.
- Señalización de posiciones objetivo a partir de sistemas de partículas.

3.7.6. Tarea de la vida diaria en una cocina

Con esta prueba se pretende comprobar todas las funcionalidades añadidas durante el desarrollo del sistema generando un entorno de AVD (Mehrholz et al., 2008) basado en una cocina para realizar una tarea sencilla como es preparar un huevo frito. El escenario contiene los siguientes elementos dinámicos de interacción: un huevo, una sartén, una aceitera y un depósito de basura. El mobiliario estático de la cocina está representado por una mesa con cajones, una estantería empotrada en la pared, una vitrocerámica y una tabla para cortar. En esta actividad también se ha utilizado un brazo virtual con la mano abierta. En esta prueba se ha incorporado una interfaz gráfica para informar al usuario del objetivo a cumplir en cada momento y una barra de progreso para indicar el tiempo limite para realizar la receta. La Figura 3.41 muestra el escenario virtual completo.

Figura 3.41: Entorno virtual de la cocina donde se realiza la receta.

Aunque la finalidad principal de la tarea consiste en seguir unos pasos para cocinar un huevo frito, existe una secuencia de objetivos a cumplir en cada paso con una orden de ejecución:

1. Agarrar sartén: En primer lugar se selecciona la sartén acercando el brazo virtual a la parte señalada.

2. Depositar sartén: Existen cuatro posibles lugares donde se puede depositar la sartén. Estos lugares vienen determinados por los fogones de la vitrocerámica y sólo uno será señalizado de manera aleatoria.

3. Encender vitrocerámica: En función de la posición donde se ha dejado la sartén, se indica uno de los cuatro botones para que sea pulsado por el usuario acercando el brazo virtual.

4. Seleccionar aceitera: El siguiente objetivo es seleccionar la aceitera situada en la estantería e indicada con un marcador circular.

5. Verter aceite: A continuación, el usuario debe colocar la aceitera encima de la sartén para verter un líquido encargado de simbolizar el aceite.

6. Dejar aceitera: Una vez completado el vertido del líquido, se indica la posición donde se debe dejar la aceitera, situado encima de la estantería.

7. Seleccionar huevo: Lo siguiente consiste en seleccionar el huevo acercando la mano del brazo virtual.

8. Freír huevo: El huevo se debe acercar encima de la sartén para realizar una simulación de rotura de la cáscara y la correspondiente caída de la yema y la clara. Adicionalmente, se ha incorporado una simulación para que la yema y la clara se transforme en un huevo frito.

9. Desechar cáscara: Para finalizar, el último objetivo consiste en dejar la cáscara vacía dentro del depósito de basura.

En esta tarea, se han añadido muchos más objetivos que en los casos anteriores, por tanto el fichero de objetivos es más extenso. Hay que destacar la incorporación de varias simulaciones y de dos tipos diferentes de señalización de los elementos. Por otro lado, el rendimiento del ejercicio se mide en función el tiempo que tarda el usuario en completar todos estos objetivos, el cual tiene un tiempo límite para ello, y el nivel de dificultad se cambia modificando este tiempo limite. En esta tarea se han probado las siguientes funcionalidades del sistema:

- Los módulos de lectura de ficheros XML junto con la organización de la escena y el almacenamiento de los elementos de la escena.

- El módulo de captura de datos de entrada para almacenar valores de terapia y determinar el tiempo que tiene el usuario para completar la receta en cada nivel de dificultad.

- Gestión de la exposición de la información correspondiente a cada objetivo y la acumulación de tiempo mostrada en la barra de progreso.

- Sistema de señalización de posiciones objetivo a partir de sistemas de partículas o marcadores circulares.

- Ejecutar simulaciones durante la realización de los objetivos.

- Selección aleatoria de posiciones objetivo.

3.8. Experimentación

En esta sección se van a definir los sistema de neuro-rehabilitación utilizados para interactuar con las tareas virtuales implementadas por este sistema. El módulo UDP se encarga de recibir la información de la posición y la orientación del efector final de los sistemas de interacción externos para controlar el avatar de las tareas virtuales. La modularidad del sistema diseñado permite incorporar direcciones IP para establecer enlaces con cualquier programa independiente. De esta manera, las tareas virtuales no se especifican para un único dispositivo, sino que se podrían utilizar para múltiples dispositivos robóticos, ya que las tareas están optimizadas para su comunicación con este tipo de elementos. En los siguientes apartados se muestran los tres sistemas robóticos de neuro-rehabilitación y el controlador háptico que se han elegido para la experimentación con las tareas.

3.8.1. Dispositivo háptico Phantom Omni

Antes de distribuir tareas virtuales a los sistemas de neuro-rehabilitación para pacientes reales se debe verificar su correcto funcionamiento. Para ello, se utiliza el dispositivo háptico de tipo *joystick* llamado *Phantom Omni* de *Sensable* (Sensable, 2015) para simular el comportamiento de un efector final con seis Grados De Libertad (GDL), tres para la posición y tres para la orientación. Permite al usuario tener una entrada en 3D y abarcar todas las zonas posibles del entorno virtual. Proporciona características de facilidad en la configuración, diseño portable y tamaño compacto con un espacio de trabajo flexible. En la Figura 3.42 se muestra el dispositivo háptico.

Figura 3.42: Dispositivo háptico Phantom Omni de Sensable.

Aparte de conseguir el punto en el espacio de trabajo, el dispositivo permite tener una realimentación de fuerzas para dar al sujeto la sensación de tacto mientras interactúa con los entornos virtuales (Burdea, 1999). Adicionalmente, se ha implementado un programa que recibe por UDP los puntos de colisión entre el avatar y los objetos del entorno virtual, así como los momentos y la magnitud de la colisión, para renderizar la sensación de contacto a partir las correspondiente fuerzas con sus direcciones, justo en el punto equivalente dentro del espacio de trabajo físico del *joystick*. Gracias a la herramienta software *OpenHaptics* (Itkowitz et al., 2005) se puede efectuar el renderizado háptico de fuerzas utilizando los valores obtenidos del motor físico. En la Figura 3.43 se muestra una diagrama con los procesos necesarios para implementar el renderizado de fuerzas en función de las APIs HLAPI y HDAPI de *OpenHaptics*.

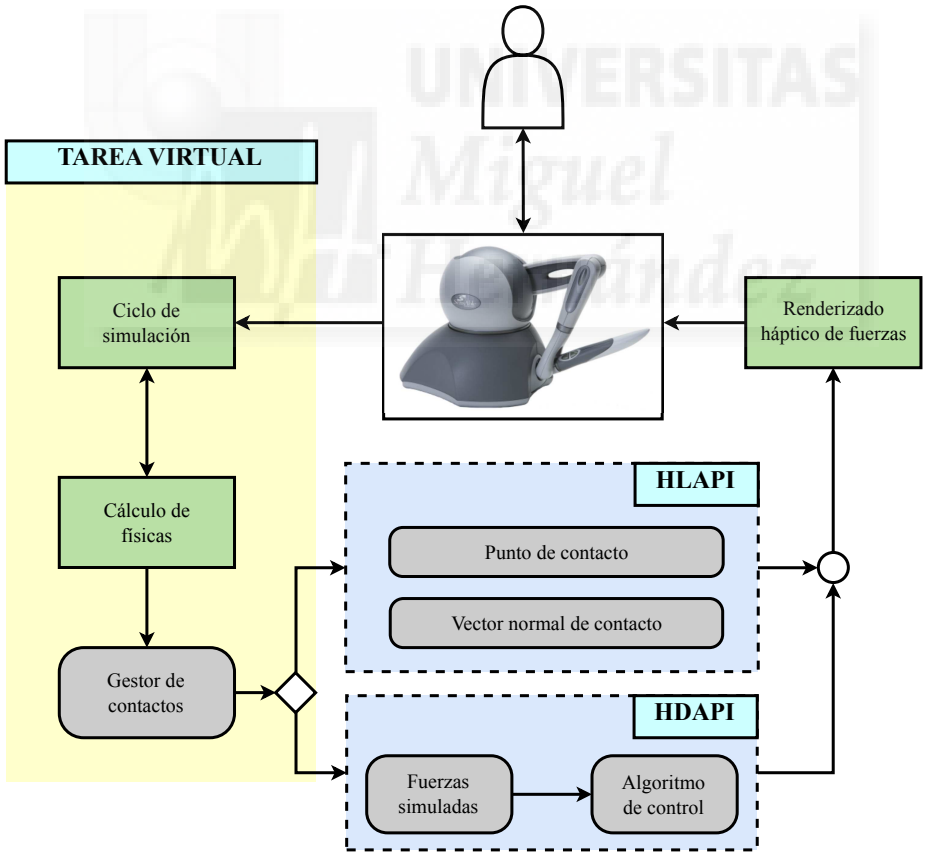


Figura 3.43: Procesos para renderizar fuerzas con Phantom Omni de Sensable.

Con HDAPI se realiza un renderizado directo de las fuerzas simuladas por el motor físico, pero necesita un algoritmo de control para reducir el error entre fuerzas ideales y aplicables, y así evitar cambios bruscos de fuerzas para que exista un control estable del dispositivo. Por otro lado, HLAPI utiliza la geometría de los objetos tridimensionales para extraer puntos de colisión y la dirección de contacto. De esta manera, se realiza el renderizado háptico evitando el hecho de tener que elaborar una gestión eficiente en la representación de fuerzas. En este trabajo se ha implementado un programa con los dos tipos de criterios para simular la renderización de las fuerzas.

En conclusión, este dispositivo permite la simulación de un robot basado en un efector final y se ha utilizado en la experimentación del funcionamiento de las tareas virtuales. En la Figura 3.44 se puede observar un usuario controlando el *joystick* del dispositivo háptico para dirigir la orientación y la posición del avatar dentro de una tarea virtual.

Figura 3.44: Pruebas experimentales del sistema con el dispositivo Phantom Omni de Sensable.

3.8.2. Dispositivo robótico PUPArm

El primer sistema de neuro-rehabilitación utilizado para comprobar el funcionamiento del motor de tareas, con el objetivo de presentar tareas virtuales a pacientes, está formado por el sistema robótico llamado PUPArm (Badesa et al., 2014a) y un subsistema de visualización. Este sistema fue diseñado y desarrollado por el Grupo de Neuroingeniería Biomédica en la Universidad Miguel Hernández de Elche como un robot de rehabilitación para pacientes con ACV u otros trastornos neurológicos. Este sistema de neuro-rehabilitación se muestra en la Figura 3.45.

Figura 3.45: Sistema de neuro-rehabilitación formado por el robot PUPArm.

La estructura robótica consiste en cuatro barras metálicas, de manera similar al robot de rehabilitación MIT-MANUS (Krebs et al., 1998). Estas barras están conectadas como un paralelogramo y se impulsan por módulos giratorios neumáticos de la serie DSMI para aportar un grado más de seguridad en la interacción hombre-robot gracias a la compresibilidad del aire. Esta estructura proporciona un manipulador planar con movimiento en dos dimensiones. Por consiguiente, el sistema sólo permite el movimiento horizontal de la extremidad superior de los sujetos, envolviendo la flexión y extensión del codo y el hombro, la abducción y la aducción horizontal. El efector final se compone de un mango acolchado para un mejor agarre y un soporte para el antebrazo, lo que permite liberar del peso del miembro

superior al paciente. Posee varios niveles progresivos de actuación entre el robot y el usuario, desde la completa asistencia donde el usuario no necesita realizar algún movimiento para llegar a los objetivos, hasta un nivel donde el robot aplica una resistencia al movimiento del usuario.

Por otra parte, el subsistema de visualización se compone de un monitor de ordenador y un software llamado REVIRE que se utiliza como sistema de simulación RV para visualizar actividades en coordinación con los movimientos del robot. El mismo ordenador se encarga de coordinar en tiempo real los actuadores neumáticos, los objetivos de las tareas y la información al usuario. El sistema también se encarga de registrar información sobre el progreso del paciente en la rehabilitación.

Como se ha comentado, el módulo de comunicación mediante el protocolo UDP permite el intercambio de datos entre robot y el motor de tareas. Por lo tanto, se establece una dirección IP y un puerto dentro del sistema para comunicar la posición del efector final del dispositivo con el avatar de control de las tareas y desplazar dicho elemento hasta completar los objetivos de la terapia de neuro-rehabilitación. En la Figura 3.46 se muestra a un usuario utilizando el sistema de rehabilitación mientras completa una de las tareas generadas.

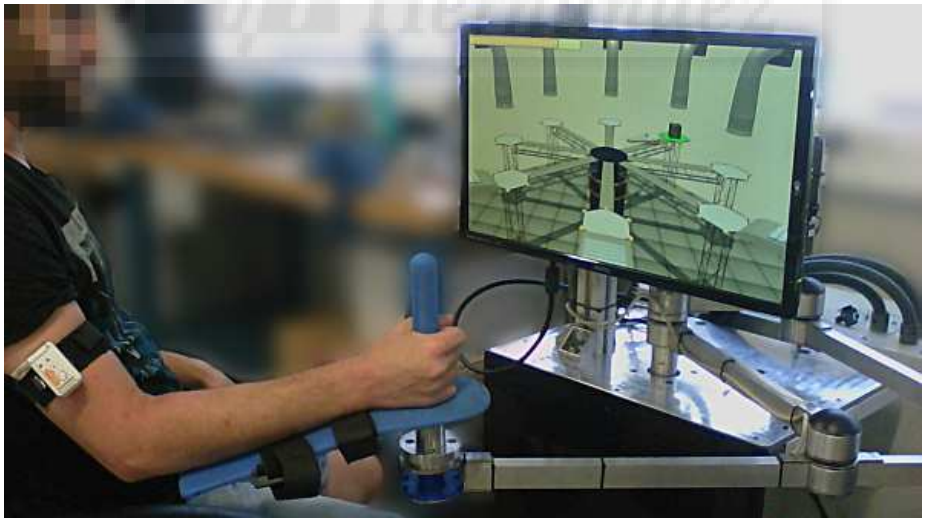


Figura 3.46: Pruebas experimentales del sistema con el dispositivo robótico PUPArm.

3.8.3. Dispositivo robótico HELPER

Al igual que el dispositivo robótico anterior, el sistema HELPER (Díez et al., 2016) ha sido diseñado por el Grupo de Neuroingeniería Biomédica con el objetivo de proporcionar la rehabilitación del miembro superior para pacientes con ACV en ambas posiciones: sedentación y decúbito supino. Es una extensión del dispositivo PUPArm, en el cual se ha añadido un tercer grado de libertad para tener un espacio de trabajo en 3D. Tiene varios modos actuación, el movimiento libre, movimiento asistido y la aplicación de fuerzas en dirección contraria al movimiento del paciente para crear un modo resistivo. En la Figura 3.47 se muestra el sistema de neuro-rehabilitación HELPER.

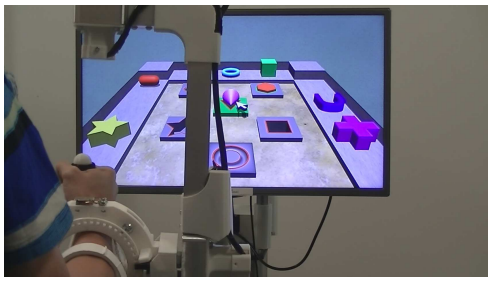
Figura 3.47: Sistema de neurorehabilitación formado por el robot HELPER.

El dispositivo consiste en tres subsistemas y tiene un total de seis GDL, tres para la posición en el espacio de trabajo y tres para la orientación del brazo. Los dos primeros grados de libertad forman un paralelogramo similar al PUPArm situado de manera paralela al suelo permitiendo un movimiento horizontal del brazo. Para el movimiento vertical posee otro grado de libertad, lo que permite una actuación en contra de la gravedad. Estos

tres primeros GDL se utilizan para la posición, mientras que el efector final consta de los tres grados de libertad restantes, los cuales son pasivos y está enganchados a la muñeca para orientar libremente el brazo con una posición cómoda. Por otra parte, también posee un sistema de visualización formado por una pantalla y una columna de elevación para adaptar la pantalla a la altura del paciente. En la Figura 3.48 se muestra el efector final del sistema de neuro-rehabilitación.

Figura 3.48: Efector final del sistema de neurorehabilitación HELPER.

Como este sistema robótico ha sido diseñado para distribuirlo a hospitales donde se realizan terapias a personas que han sufrido un ACV, se debe incorporar una serie de tareas virtuales para maximizar la motivación y participación de los pacientes durante la terapia. Por lo tanto, se ha realizado una prueba experimental entre el robot y el motor de tareas. En la Figura 3.49 se muestran dos tareas incorporadas en el sistema de visualización del dispositivo robótico.



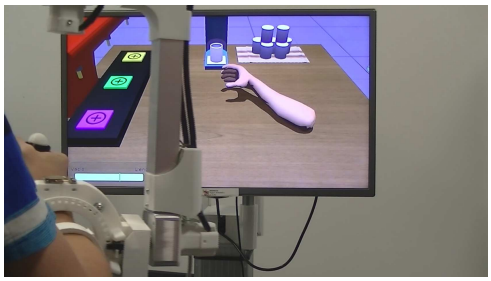


Figura 3.49: Pruebas experimentales del sistema con el dispositivo robótico HELPER.

3.8.4. Dispositivo robótico MAAT

El último dispositivo es una plataforma robótica diseñada para realizar tareas basadas en AVD como beber un vaso de agua, lavarse los dientes, peinarse y similares, por lo que ha sido implementado siguiendo un modelo cinemático del brazo humano. También ha sido diseñado por el mismo grupo para la rehabilitación del miembro superior. Es un robot serial redundante de siete GDL, cuyo lazo de control se implementa con accionamientos eléctricos a través de motores PRL proporcionados por *Schunk*. Dichos motores permiten el control en posición y velocidad, y utiliza un ordenador para obtener el control del robot. Está dentro del marco del proyecto de investigación MAAT (*Multimodal interfaces to improve therApeutic outcomes in robot-Assisted rehabiliTation*), el cual pertenece al proyecto europeo ECHORD (European Clearing House for Open Robotics Development). En la Figura 3.50 se muestra el sistema de neuro-rehabilitación MAAT.

Figura 3.50: Sistema de neurorehabilitación formado por el robot MAAT.

El efector final posee un punto de agarre en la muñeca del paciente a través de un acoplador electromagnético. Este acoplador proporciona una medida de seguridad extra ante posibles fallos, al permitir un desacople bastante rápido entre el robot y el paciente. En la Figura 3.51 se muestra el efector final del sistema, el cual posee un sensor de fuerza-par de seis GDL para gestionar la interacción física entre el hombre y la máquina.

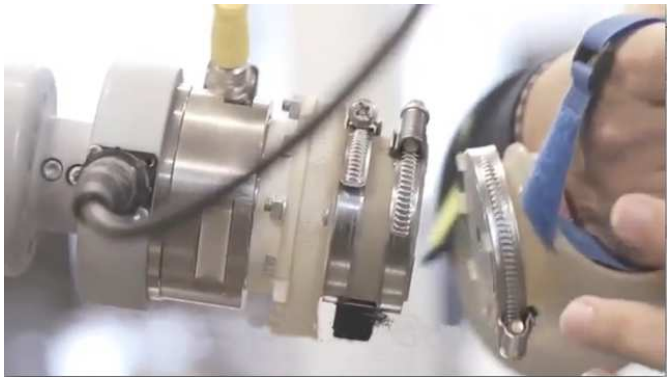


Figura 3.51: Efector final del sistema de neurorehabilitación MAAT.

Para comprobar la comunicación del sistema robótico con el motor de tareas se ha propuesto la realización del ejercicio inspirado en AVD sobre alcanzar un vaso y llevarlo a la boca durante múltiples repeticiones. Esta prueba experimental sólo se realiza para verificar la utilización de este tipo de dispositivo robótico con el motor de tareas en términos de funcionalidad y compatibilidad, y no para realizar una recuperación del paciente con una terapia activa. En la Figura 3.52 se muestra un usuario utilizando el dispositivo a través del efector final mientras visualiza la tarea virtual en una pantalla.

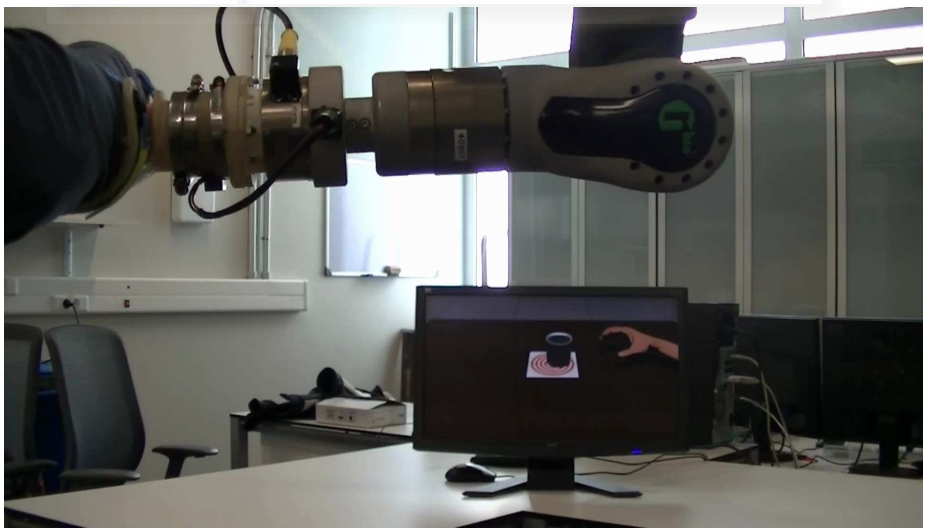


Figura 3.52: Pruebas experimentales del sistema con el dispositivo robótico MAAT.

3.9. Conclusiones

En este capítulo se ha presentado el proceso de captura de requisitos, análisis, diseño, implementación y pruebas experimentales de un sistema de generación de tareas virtuales destinadas a terapias de neuro-rehabilitación del miembro superior asistidas por dispositivos robóticos. En un principio se han detallado los elementos software de libre distribución que lo han fundamentado. El proceso de implementación del sistema ha constado de dos partes bien diferenciadas: la parte de la producción de contenido multimedia y la generación del código capaz de realizar simulaciones basadas en principios físicos. De esta manera, el sistema se encarga de proporcionar objetivos y movimiento a un escenario virtual, siendo posible crear múltiples tareas sin modificar una sola linea de código. Sin embargo, la configuración u objetivos de algún tipo de tarea puede necesitar incorporar algún nuevo bloque de código para controlar algún tipo de objetivo o animaciones. A pesar de ello, una vez establecido el nuevo bloque, ya se podrían diseñar más tipos de tareas en función de los diversos objetivos que pueden surgir durante el diseño de los escenario y los elementos. Simplemente se modifica la clase *Target* para añadir las nuevas funcionalidades que han surgido durante el desarrollo de las pruebas. De esta manera, el sistema se puede extender incorporando los nuevos bloques de programación gracias a su diseño modular.

Como resultado se ha obtenido un método de generación de tareas de manera rápida y de bajo coste, lo que permite minimizar los gastos ocasionados durante el proceso de producción. Crea entornos con el mismo tipo de objetivos para completar trayectorias con los dispositivos robóticos añadiendo simplemente ficheros con etiquetas, las cuales son leídas por los módulos de lectura del sistema para automatizar toda la gestión de objetivos alcanzables y la organización de la escena virtual. Permite la creación de tareas adaptadas a cada paciente comprobando cuáles pueden ser más beneficiosas para su neuro-rehabilitación motora. Además, el sistema desarrollado permite la adaptación del nivel de dificultad de las tareas en función del estado del paciente, característica que favorece la motivación y la atención del usuario durante la terapia. Otro punto importante es la capacidad de actualización del funcionamiento interno de la tarea modificando parámetros de terapia, como el número de repeticiones, el tiempo para completar trayectorias, la longitud máxima de movimiento o el nivel de asistencia. En conclusión, se puede afirmar que se han cumplido todos

los requisitos preestablecidos ofreciendo una herramienta con una elevada versatilidad y flexibilidad para implementar todo tipo de ejercicios virtuales adaptables automáticamente. Por otro lado, cabe destacar que este sistema de simulación de tareas virtuales y los dispositivos robóticos PUPArm y HELPER, actualmente se están utilizando en el Hospital de larga estancia de la Pedrera en Dénia, en la Unidad de Daño Cerebral del Hospital de San Vicente del Raspeig y en el Hospital Vega Baja de Orihuela para terapias de neurorehabilitación de pacientes que han sufrido un ACV.

Capítulo 4

Sistema de interacción auto-adaptativo para terapias virtuales de rehabilitación robótica

En este capítulo se describe la implementación de un sistema multimodal de rehabilitación asistido por dispositivos robóticos utilizando una nueva técnica de estimación del estado psico-fisiológico del paciente, a partir de la medición y el análisis de señales fisiológicas, para auto-adaptar el nivel de dificultad de tareas virtuales dentro de una terapia de rehabilitación robótica.

En la primera parte del capítulo se definen los tipos de señales fisiológicas junto con el proceso de extracción y procesamiento de la señal para obtener los parámetros fisiológicos más destacables. Además, se presenta la técnica de clasificación utilizada para estimar el estado del individuo, a partir de un modelo matemático formado por una red neuronal ART y un sistema de

lógica difusa.

En la siguiente parte, se explica la metodología experimental utilizada para evaluar si la técnica de clasificación neuro-difusa es robusta y estable a la hora de estimar el estado emocional de una serie de usuarios, a los cuales se les ha inducido distintos niveles de estrés mientras realizaban una tarea virtual en 2D con ayuda de un dispositivo de asistencia robótico.

Para finalizar, se integra el proceso de extracción y el procesado de las señales fisiológicas con el método de clasificación neuro-difuso y una de las tareas implementadas en el capítulo 3 para definir el sistema de interacción hombre-robot basado en la adaptación del nivel de dificultad en función del estado emocional del paciente dentro de una terapia virtual con un entorno visual con un alto grado de realismo.

4.1. Introducción

En este capítulo se presenta el desarrollo de un sistema multimodal que combina los procesos de captura, análisis y clasificación de señales fisiológicas para obtener el estado emocional de pacientes con el objetivo de adaptar de manera automática el nivel de dificultad de tareas virtuales dentro de una terapia de neuro-rehabilitación asistida por dispositivos robóticos.

La actual evolución de la tecnología robótica ha demostrado que este tipo de dispositivos tienen un papel muy importante en la rehabilitación neurológica (Miller et al., 2010), sin embargo, aún existen muchos retos por resolver. Por consiguiente y a pesar de la creciente popularidad de la robótica en la neuro-rehabilitación, todavía se discute su eficacia de manera controvertida. Una cuestión importante en este campo es el de promover la participación activa del paciente en el bucle de control, el cual se refiere al concepto de permitir al ser humano actuar cooperativamente en lugar de tratarlo como una fuente de perturbación. Por lo tanto, un dispositivo robótico ideal debería ser capaz de decidir qué nivel dificultad se debe aplicar en diferentes escenarios de rehabilitación teniendo en cuenta la información bio-mecánica, así como la fisiológica y los aspectos emocionales de los pacientes que subyacen en las terapias asistidas por robots.

La emoción es un estado complejo de sentimientos que implica reacciones psicológicas y fisiológicas producidas por las interacciones entre los seres humanos y el entorno. Existe una clasificación de emociones (también llamados estados afectivos) ampliamente aceptada, donde se describen como un modelo circumplejo con dos dimensiones: valencia y excitación (Russell, 1978; Russell, 1980). La dimensión de valencia puede tomar valores desde estados de desagrado hasta placer, y por otro lado, la dimensión de excitación puede tomar valores desde estados de desactivación (entre el estado del sueño hasta la somnolencia) a estados de activación como pueden ser diversas etapas de alerta o una excitación frenética. Los estudios publicados sobre sistemas neuronales implicadas en el uso de técnicas de neuro-imagen sugieren que la dimensión de valencia y la de excitación se pueden asociar con circuitos neuronales separados que contienen las regiones de la amígdala, la ínsula, el tálamo, la corteza cingulada dorsal anterior y la prefrontal (Anderson et al., 2003; Small et al., 2003; Anders et al., 2004; Nielen et al.,

2009; Posner et al., 2009; Gerber et al., 2008; Colibazzi et al., 2010). La mayoría de estos estudios muestran que la amígdala es el núcleo de la región afectiva sugiriendo que esta región puede pertenecer tanto al sistema neuronal de la dimensión de valencia como a la de excitación.

Por otro lado, la Teoría de Redes se puede aplicar también para el cálculo neuronal de las emociones como se describe en la propuesta conceptual de Pessoa sobre los cálculos neuronales y emocionales (Pessoa, 2008). En base a esta hipótesis, Feng (Shu y Tan, 2012) propone una Cognitive Regulated Affective Architecture (CRAA), la cual comprende una red cognitiva, una red afectiva y una capa de valoración. Esta red afectiva se diseñó para simular las funciones de la amígdala, y se construyó utilizando una red neuronal basada en modelos de la Teoría de Resonancia Adaptativa (ART). Por esta razón, en este capítulo se ha propuesto la hipótesis de que las redes neuronales, especialmente las redes neuronales basadas en ART, deberían funcionar mejor que los clasificadores implementados previamente en la literatura científica para estimar el estado emocional de los usuarios en función de las reacciones fisiológicas durante las terapias de rehabilitación asistidas por dispositivos robóticos. Novak en (Novak et al., 2012) proporciona el resumen de una lista de estudios comparando diferentes algoritmos de clasificación determinados por el número de sujetos incluidos en cada estudio, las metodologías utilizadas y el tipo de clasificador acorde a su precisión.

En trabajos previos (Badesa et al., 2014c) se han utilizado nueve técnicas de aprendizaje automático para estimar distintos estados de usuarios, como aburrido, contento y excitado. Los resultados muestran que las Máquinas de Soporte Vectorial (SVM) con Función de Base Radial (RBF) proporcionan los mejores resultados en términos de precisión (91.43 %). Sin embargo, para un uso más amplio de estas tecnologías se necesitan métodos capaces de cubrir pacientes con una amplia variedad de características físicas, así como deterioros cognitivos y que sean capaces de adaptarse automáticamente a las demandas y necesidades especificas del paciente. Por lo tanto, a causa del amplio uso de las redes neuronales en el modelado de procesos neuronales relacionados con las emociones, en esta tesis se propone una investigación sobre la utilidad potencial de las redes neuronales que incorporan conceptos de la teoría de lógica difusa para estimar el estado emocional del usuario y comprobar el rendimiento de esta tecnología a la hora de adaptar el nivel de dificultad de tareas con entornos de RV realistas en 3D. Finalmente, se integran todos los aspectos del capítulo para definir una interfaz de interacción multimodal auto-adaptativa.

4.2. Señales fisiológicas

Las señales fisiológicas son registros de eventos biológicos variantes en el tiempo controladas por el SNC para gestionar músculos, glándulas y órganos del cuerpo humano. Dichos eventos generan actividad química, eléctrica o mecánica de manera inconsciente que puede ser registrada y analizada para obtener la respuesta emocional de un usuario frente a estímulos del entorno.

4.2.1. Tipos de señales

Los procesos fisiológicos originan diferentes tipos de señales: señales bio-eléctricas donde el sistema nervioso genera señales eléctricas entre las células debidos a cambios electro-químicos, cuya estimulación produce un potencial de acción a través de la membrana celular; señales bio-magnéticas asociadas a los campos magnéticos generados por órganos específicos; señales bio-químicas con información de los niveles de los agentes químicos del cuerpo como las hormonas; señales bio-mecánicas calculadas a partir de la tensión, fuerza, flujo o presión de funciones mecánicas.

A continuación se muestra una lista con las señales fisiológicas elegidas para estimar el estado psico-fisiológico del usuario, así como una breve explicación de sus características principales:

1. **Ritmo cardíaco**: El ritmo cardíaco es la cantidad de veces que el corazón llega a latir para bombear la sangre a través de las arterias hacia todos los órganos dependiendo de la situación de esfuerzo o reposo sometida por el organismo. El método encargado de registrar esta señal es el análisis de la onda de pulso, el cual está basado en Fotopletismografía (PPG)(Allen, 2007).

2. **Frecuencia respiratoria**: La frecuencia respiratoria representa el número de veces que un ser vivo absorbe oxigeno y expulsa dióxido de carbono durante un periodo de tiempo. La variación en la frecuencia puede indicar diferentes emociones, ya que una respiración acelerada es signo de ira, miedo o incluso alegría, mientras que una respiración lenta muestra un estado de relajación, reposo o depresión.

3. **Temperatura de la piel**: Otra bio-señal fácil de registrar es la temperatura de la superficie de la piel. Existen mecanismos dentro del organismo, los cuales se encargan de modificar la temperatura bajo

diferentes condiciones externas o internas para mantener un equilibrio entre ganancia y perdida de calor. Por ejemplo, debido a la tensión provocada por el movimiento de los músculos, los vasos sanguíneos se contraen a partir de la circulación de diferentes flujos de sangre y la temperatura disminuye.

4. **Respuesta galvánica de la piel**: De forma similar a la temperatura de la piel, la respuesta galvánica de la piel o respuesta electrodérmica representa un cambio en las propiedades superficiales de la piel, más concretamente en su capacidad para conducir la electricidad a través de los nervios y las glándulas sudoríparas. Esta señal también es conocida como la conductividad de la piel. La señal Galvanic Skin Response (GSR) contiene dos componentes parametrizables: el Nivel de Conductancia de la Piel (SCL) es una medida de la cantidad global de la excitación ante variaciones rápidas y puntuales ante un estímulo, mientras que la Respuesta de Conductancia de la Piel (SCR) refleja ligeros cambios transitorios y se puede adquirir con una frecuencia de muestreo baja (32Hz).

4.2.2. Adquisición

En este apartado se describe el material utilizado para el registro de las señales fisiológicas encargadas de medir y monitorizar las reacciones del cuerpo. En la Figura 4.1 se muestra el esquema general del sistema de adquisición.

Los cuatro tipos de señales fisiológicas se registran mediante una serie de sensores suministrados por la empresa *g.tec*. El sensor *g.PULSEsensor* proporciona el pulso cardíaco como una señal analógica bastante clara que refleja las variaciones en el flujo sanguíneo de los vasos situados en las falanges o en los lóbulos de las orejas (concretamente en la última falange del dedo pulgar), utilizando la técnica PPG para calcular la cantidad de luz que emite el sensor y cuanta luz es absorbida durante el proceso. Para la medición de la respiración se ha utilizado el sensor *g.FLOWsensor*, el cual es un termistor que se sitúa en la zona nariz-boca y se encarga de medir el cambio de temperatura al inspirar o espirar el oxígeno. El sensor de temperatura de la piel se llama *g.TEMPsensor* y provee un tensión de salida en mV proporcional a la temperatura de la piel. La GSR se obtiene con el sensor *g.GSRsensor* y mide la conductancia colocando dos electrodos fija-

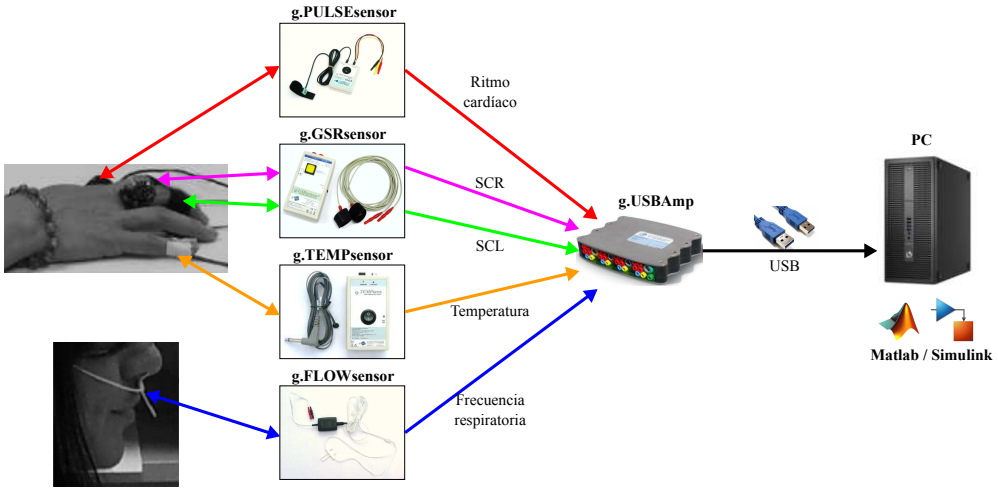


Figura 4.1: Esquema general del sistema de adquisición de señales fisiológicas.

dos por tiras de velcro en las falanges del dedo índice y corazón. Entonces, se distribuye una pequeña corriente entre esos dos electrodos y calcula el voltaje resultante.

Una vez que el usuario tiene todos los sensores colocados, la señales se adquieren a través del amplificador *g.USBAmp* también de la empresa *g.tec*. Este dispositivo es un sistema de adquisición, amplificación y procesamiento de bio-señales de alto rendimiento y precisión. Dispone de 16 canales para muestrear simultáneamente diferentes señales con una precision de 24 bits y están interconectados por conmutadores para las etapas de amplificación internas junto con la aplicación de un filtro *anti-aliasing*. Además, posee un cable de sincronización para garantizar que todas las señales se muestrean con exactamente la misma frecuencia (en este caso se ha elegido 256Hz).

Con respecto al software de adquisición, la empresa *g.tec* proporciona *drivers* para conectar el amplificador con el ordenador y una API de Matlab. El software utilizado para el análisis y el procesado en tiempo real de las señales fisiológicas, junto con su tratamiento de filtrado, ha sido implementado con la interfaz de Simulink a partir de la API de Matlab siguiendo un procedimiento de conexión con el amplificador para reenviar la información resultante al software de control del dispositivo robótico. La Figura 4.2 presenta el esquema de Simulink.

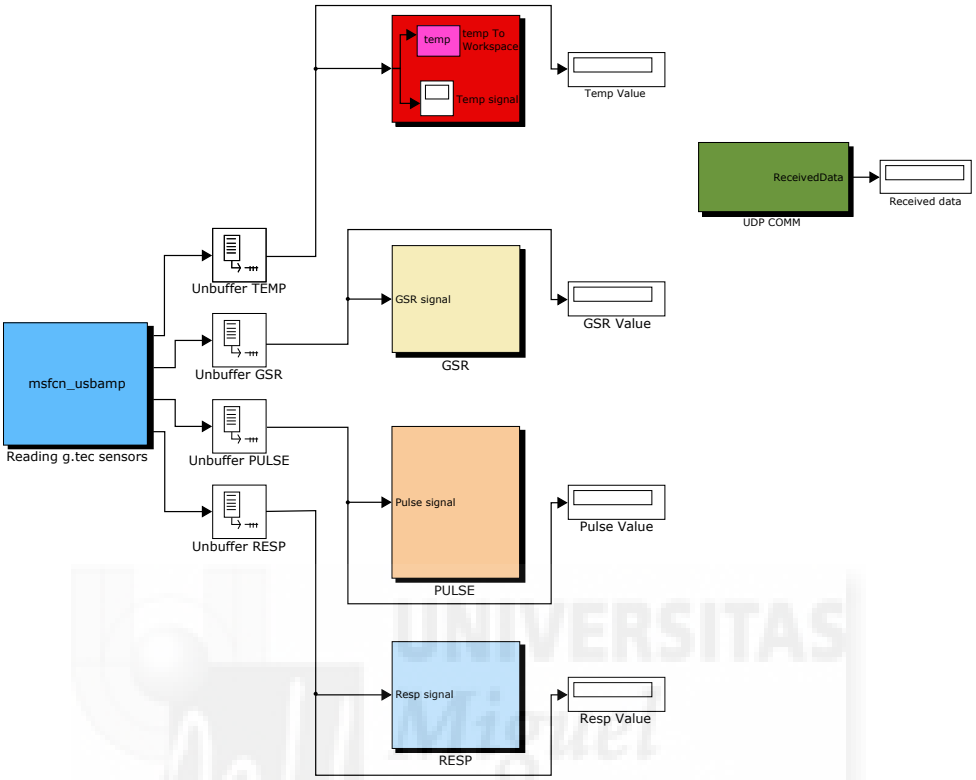


Figura 4.2: Esquema Simulink para la adquisición de señales fisiológicas.

4.2.3. Procesado

Todo proceso de adquisición tiene asociado una tarea de procesamiento encargada de extraer información relevante, o características principales, de cualquier tipo de señales. Por tanto, un análisis de las señales fisiológicas elegidas permite obtener un vector con 5 valores característicos(pulso, respiración, temperatura, SCL y SCR) durante cada medición del sistema. En la Figura 4.3 se muestra el diagrama de bloques de las técnicas de análisis y procesamiento utilizadas. Básicamente, todos los módulos de procesamiento se basan en un análisis temporal de las señales.

El análisis del ritmo cardíaco se realiza calculando el tiempo entre dos picos consecutivos de la señal, a través de un algoritmo de detección de picos, cuyo valor se invierte para obtener el ritmo cardíaco instantáneo, pero primero se debe aplicar una normalización de la señal. Con la señal de frecuencia respiratoria se aplica el mismo proceso que con la señal de pulso

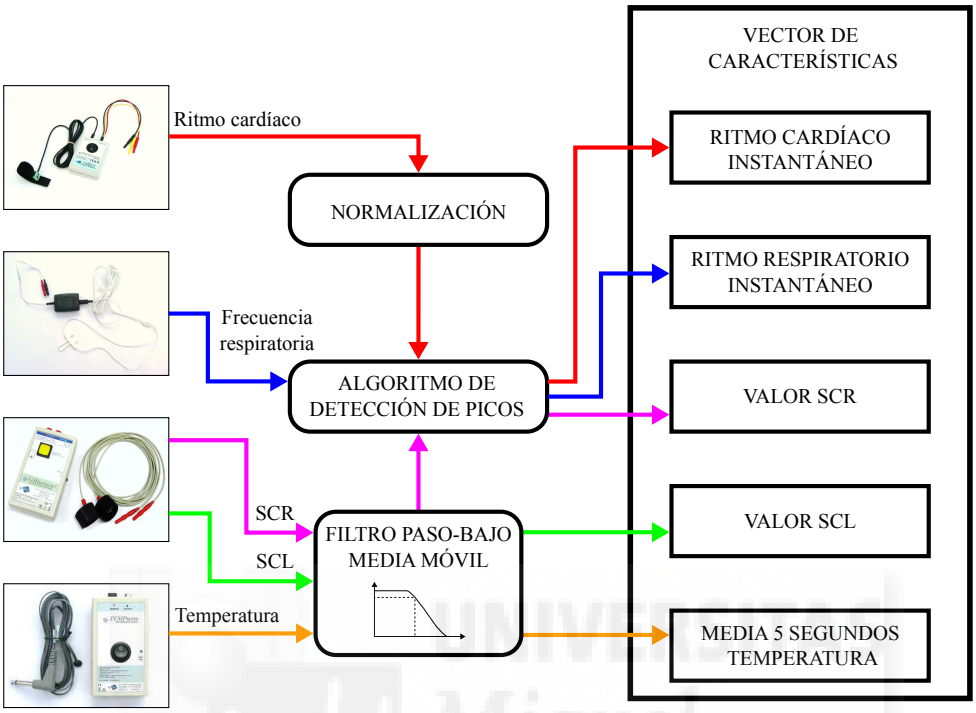


Figura 4.3: Diagrama de bloques de las técnicas de procesamiento.

sin la previa normalización. A la señal de temperatura se le aplica un filtro de media móvil sobre los últimos 5 segundos para eliminar el ruido en alta frecuencia en la medida y utilizar este valor como la característica principal de esta bio-señal.

Como se ha comentado en el apartado anterior, el análisis de la conductividad de la piel obtiene dos posibles valores característicos, uno del SCL y otro del SCR. La SCL es una señal de baja frecuencia y su valor característico se puede obtener de forma sencilla través de una media de la señal en un periodo de tiempo concreto. Por otro lado, para el SCR se emplea, en primer lugar, un filtro paso-bajo que elimina el ruido de alta frecuencia con una media a partir de los dos segundos de señal alrededor de la muestra actual, y después se aplica el algoritmo de detección de picos. El valor característico del SCR es la frecuencia de aparición de los picos, tomada como el número de picos por minuto y para que un pico se considere en el cálculo de la medida, este debe exceder de 0.05 μS y debe ocurrir menos de cinco segundos después del inicio del incremento de la señal.

4.3. Método neuro-difuso de clasificación

En esta tesis se ha propuesto utilizar un método neuro-difuso de clasificación llamado *S-dFasArt* (*Supervised and Dynamic Fuzzy Adaptative System ART-based*)(Cano-Izquierdo et al., 2012) para clasificar los patrones temporales obtenidos de la interpretación de las señales fisiológicas registradas durante terapias de rehabilitación asistidas por dispositivos robóticos y estimar en tiempo real, con fiabilidad, tres posibles estados psico-fisiológicos de los pacientes.

El clasificador *S-dFasArt* es un algoritmo que enlaza los fundamentos de la teoría de Conjuntos Difusos en un modelo de red neuronal artificial basada en ART, más específicamente en la arquitectura difusa ARTMAP (ARTMAP, 1992). De esta manera se aprovecha la capacidad de aprendizaje y adaptación de las redes neuronales, combinándolas con la robustez, la interpretabilidad y la tolerancia a fallos de los sistemas difusos a través de la incorporación de variables lingüísticas capaces de mejorar las propiedades de los algoritmos de aprendizaje, el modo de actualización y la velocidad de convergencia de los pesos de la red neuronal. Este tipo de arquitectura satisface el criterio de estabilidad-plasticidad, ya que el clasificador es capaz de adquirir nuevos patrones de aprendizaje sin perder el conocimiento acumulado comprometiendo su capacidad de adaptación a nuevas condiciones. Las características del *S-dFasArt* permiten un aprendizaje muy rápido con un conjunto reducido de patrones de entrenamiento cuyos valores se presentan a la red sólo una vez.

El modelo *S-dFasArt* establece una dualidad neuro-difusa entre los conceptos de activación y funciones de pertenencia para gestionar una aprendizaje supervisado de la información de entrada utilizando ecuaciones dinámicas para las etapas de procesamiento del algoritmo. La estructura interna de la red neuronal se organiza por nodos o categorías en función de la cantidad de muestras utilizadas en el entrenamiento, y durante el proceso de estimación todos estos nodos reaccionan ante un valor de entrada, pero el clasificador sólo activa el nodo con el nivel de respuesta más alto. Por lo tanto, utiliza una estimación competitiva donde la categoría asociada con el nodo ganador es la clasificación de la red para el patrón de entrada actual. La red decide a qué categoría pertenece cada dato presentado según su parecido con los nodos creados, y si el dato no se clasifica en ninguna categoría existente, la red genera una nueva.

4.3.1. Arquitectura S-dFasArt

La Figura 4.4 muestra la arquitectura general del clasificador propuesto donde se puede comprobar la combinación de operaciones neuronales y difusas. Este esquema consta de los siguientes elementos: una Capa de Entrada, una Capa de Supervisión, un Subsistema de Orientación y una Capa de Categorías.

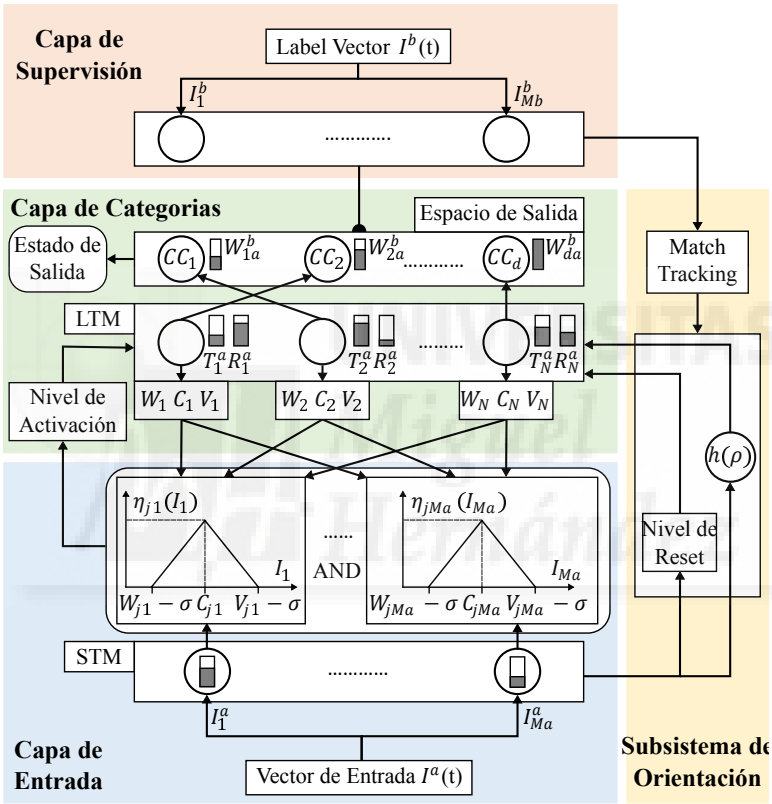


Figura 4.4: Arquitectura del método de clasificación *S-dFasArt*.

La Capa de Entrada está formado por M_a nodos. Cada nodo recibe un elemento del vector de entrada I^a para retener temporalmente los aspectos de entrada más importantes activando la memoria a corto plazo (Short Term Memory (STM)). Cada uno de los nodos de esta capa tiene asociado un bloque difuso integrado por una función triangular de activación-pertenencia η_{ji}. Estos bloques registran el grado de pertenencia de cada una de las características de entrada i con respecto a las categorías difusas j. El tamaño

de la función de pertenencia se puede determinar con el parámetro de diseño σ. Por consiguiente, este parámetro modifica el carácter difuso de las categorías de salida (Izquierdo et al., 2001).

La Capa de Supervisión tiene M_b nodos para registrar el patrón de supervisión con el estado de clasificación correcto I^b asociado a un vector de entrada. Esta capa configura el Espacio de Salida creando un número d de clases con cada una de las posibles clasificaciones que la red neuronal puede codificar. La entrada de los vectores de supervisión sólo se proporcionan a la red durante la etapa de aprendizaje.

La Capa de Categorías está formada por un conjunto de N nodos que representan todas las categorías que se han creado durante el proceso de aprendizaje, dando lugar a un conjunto de unidades difusas. Cada nodo o categoría tiene asociados dos valores principales: T_j indica el grado de activación y R_j indica la capacidad de aprendizaje de esa entrada. Cuando se le presenta al clasificador un patrón de entrada, todas las categorías se activan a un determinado nivel de activación T_j. Cada unidad de salida tiene asociados tres tipos de pesos neuronales: mínimos W_{ji}, centrales C_{ji} y máximos V_{ji}. Estos pesos se encargan de almacenar la memoria a largo plazo (Long Term Memory (LTM)), cuyos valores están vinculados con los bloques difusos para actualizar la función de pertenencia encargada de gestionar el nivel de aceptabilidad de un patrón de entrada dependiendo de la categoría que está reaccionado. La activación de las categorías difusas de salida se calcula utilizando la operación AND con todos los grados difusos de pertenencia de las características del vector de entrada. Cada categoría difusa sólo puede codificar un estado de salida CC de las d-posibilidades que se han generado en la red neuronal durante la fase de aprendizaje a partir de los valores de supervisión. Por lo tanto diferentes categorías pueden apuntar a un mismo nodo de clasificación en el Espacio de Salida. En resumen, esta capa es responsable de correlacionar los pares de secuencia de vectores de entrada con los vectores de supervisión.

Además de las capas anteriores, posee un Subsistema de Orientación para detectar la semejanza del vector de entrada con las categorías aprendidas por la red neuronal. Este porcentaje de semejanza se puede comparar con el parámetro de vigilancia $h(\rho)$ para controlar el número de categorías difusas que se deben crear en la Capa de Categorías. El parámetro de vigilancia determina cómo de estricta debe ser la red neuronal durante el proceso de clasificación de las medidas de entrada generalizando los resultados. El *match tracking* se encarga de gestionar el ajuste automático del parámetro

de vigilancia indicando si la entrada se ha clasificado correctamente en el nodo j o si el modelo tiene que crear otra categoría de salida. Si el nivel de semejanza calculado por el Subsistema de Orientación no es lo suficientemente parecido con una categoría, se produce un nivel de reset Rj para deshabilitar la categoría actual y elegir otra categoría siguiendo el criterio de máxima semejanza.

4.3.2. Algoritmo de aprendizaje

En esta sección se presenta el algoritmo de entrenamiento administrado por el clasificador durante su proceso de aprendizaje (Figura 4.5). Este algoritmo aplica un aprendizaje competitivo para generar nuevas categorías difusas. Los datos de entrada se introducen en la red neuronal sólo una vez siguiendo el orden temporal de cuando se han obtenido y procesado. Debido a este proceso, los pesos de las categorías se actualizan dinámicamente. En (Cano-Izquierdo et al., 2009) y (Toledo-Moreo et al., 2010) se puede encontrar una descripción más concreta del algoritmo de aprendizaje, sus ecuaciones dinámicas y una explicación más exhaustiva de los parámetros.

Los nodos de la Capa de Categorías compiten entre ellos pero sólo se produce la fase de aprendizaje en la categoría con el nivel de activación más alto. El funcionamiento del algoritmo *S-dFasArt* puede describirse a través de los siguientes pasos:

1. En primer lugar, se inicializa el clasificador definiendo los valores permanentes de los parámetros de las ecuaciones dinámicas como la velocidad de activación, la velocidad de crecimiento del nivel de reset, constantes de tiempo, ganancias dinámicas y el parámetro de vigilancia. Estos parámetros afectan directamente al comportamiento del aprendizaje. Inicialmente, como no se ha presentado a la red ningún patrón de entrada, la red neuronal no posee, de momento, ningún nodo asociado con pesos en la Capa de Categorías ni en el Espacio de Salida.

2. Antes de empezar el proceso de aprendizaje, se debe obtener K-número de muestras formadas por las medidas de entrenamiento $I^a(t)$ y sus correspondientes etiquetas de supervisión $I^b(t)$.

3. Un patrón de entrada h se presenta a la red neuronal. Cada nodo de la Capa de Entrada recibe una característica del vector de entrada,

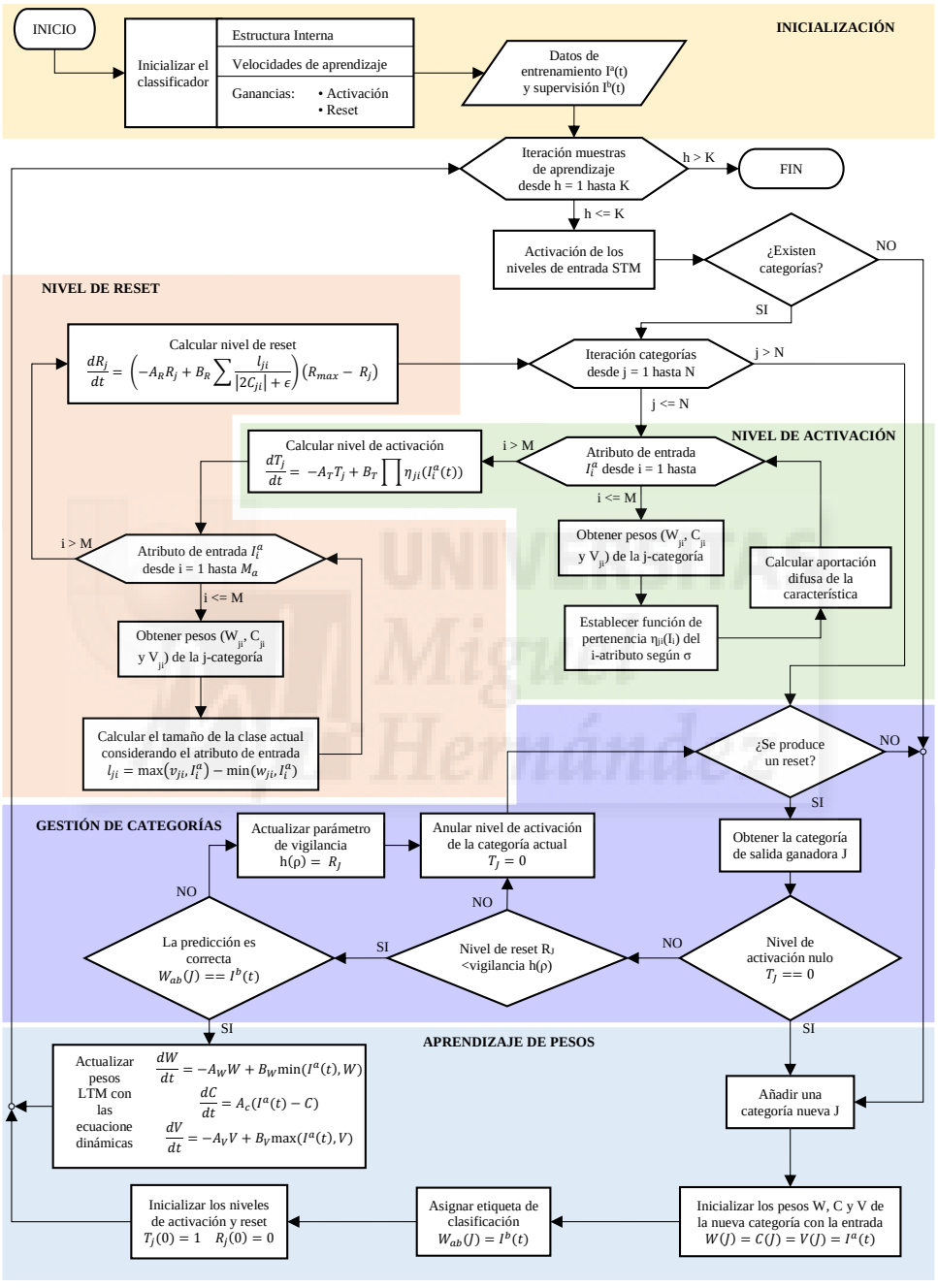


Figura 4.5: Diagrama de flujo del algoritmo de aprendizaje *S-dFasArt*.

activando el STM para indicar el grado de presencia de cada atributo de la señal. Además, el vector de supervisión se presenta a la Capa de Supervisión, activando sus nodos.

4. Cuando la red obtiene un patrón de entrada y una etiqueta de supervisión, se comprueba si existen categorías para calcular los niveles de activación y reset. Si la red no tiene categorías, no se realiza este paso.

5. Si existen categorías se ejecuta la siguiente secuencia de pasos para cada una de las j-categorías existentes:

   a) La categoría actual j envía los pesos W_{ji}, C_{ji} y V_{ji} al bloque de lógica difusa asociado a cada uno de los i-nodos de la Capa de Entrada y actualiza su función de activación-pertenencia, considerando el parámetro σ para controlar el tamaño y el carácter difuso de las categorías.

   b) Entonces, se calcula el valor de pertenencia difuso asociado a cada nodo de entrada que determina el nivel de activación T_j de cada característica del vector de entrada. El nivel de activación de las categorías procesadas se puede calcular a partir de estos valores de pertenencia y la operación AND, utilizando las ecuaciones dinámicas encargadas del proceso de activación del algoritmo *S-dFasArt* (4.1). El parámetro A_T es la velocidad de activación de las categorías y determina la sensibilidad de la red neuronal a la hora de responder a los cambios en las entradas. El parámetro $\eta_{ji}(I_i^a)$ representa la contribución difusa de cada una de las características de la muestra de entrada para calcular el nivel de activación, aplicando la función de activación-pertenencia mientras que el parámetro B_T especifica la ganancia dinámica de la contribución difusa total.

$$\frac{dT_j}{dt} = -A_T T_j + B_T \prod_{i=1}^{M} \eta_{ji}\left(I_i^a\left(t\right)\right) \tag{4.1}$$

   c) A continuación, el tamaño de la categoría difusa se calcula teniendo en cuenta los datos de entrada como un patrón adjunto a esta categoría, es decir, el grado de semejanza que indica si el patrón de entrada es un subconjunto de los pesos de la categoría asociada. Por lo tanto, cada i-nodo de entrada proporciona

una magnitud (4.2) para calcular el tamaño total de la categoría como la suma de todos estos valores (4.3).

$$l_{ji} = max\left(V_{ji}, I_i^a\right) - min\left(W_{ji}, I_i^a\right) \tag{4.2}$$

$$dreset \quad = \sum_{i=1}^{M} \frac{l_{ji}}{|2C_{ji}| + \epsilon} \tag{4.3}$$

d) Este tamaño se utiliza para calcular el nivel de reset la categoría actual a partir de la ecuación de reset del *S-dFasArt* (4.4). R_j puede considerarse como un umbral de semejanza necesario para que un vector de entrada se pueda asociar a la categoría j. A_R consiste en la velocidad de crecimiento del nivel de reset que permite al sistema responder a los cambios dinámicos de los datos de entrada. Esto implica que la categoría puede aprender a partir de la entrada. El parámetro B_R determina la ganancia relacionada con el tamaño total de la categoría. R_{max} establece el valor máximo de reinicio de las categorías.

$$\frac{dR_j}{dt} = \left(-A_R R_j + B_R dreset\right)\left(R_{max} - R_j\right) \tag{4.4}$$

6. En este punto, se han calculado los niveles de activación y de reset de las N categorías que actualmente existen dentro de la red neuronal. Entonces, se determina la categoría ganadora J seleccionando la categoría con el valor más alto de activación (4.5).

$$T_J = max\left\{T_j; \quad j = 1...N\right\} \tag{4.5}$$

7. Si el nivel de activación máximo es nulo ($T_j = 0$), se incorpora una categoría no comprometida con pesos nulos. Entonces se aplica un rápido aprendizaje (4.6) para establecer los pesos de este nodo no comprometido como un prototipo del patrón de entrada, y se asigna a la etiqueta que recibe la Capa de Supervisión como el resultado de clasificación de la categoría actual (4.7) en el Espacio de Salida. Además, se inicializan los niveles de activación y reset para el primer instante de tiempo (4.8).

$$W = C = V = I^a\left(t\right) \tag{4.6}$$

$$W_{ab} = I^b(t) \tag{4.7}$$

$$T_j(0) = 1 \qquad R_j(0) = 0 \tag{4.8}$$

8. Sin embargo, si la categoría ganadora tiene un nivel de activación no nulo se compara el nivel de reset con el parámetro de vigilancia. Si el valor del reset supera el umbral de vigilancia se produce un estado de reset debido a que el nodo ganador no representa apropiadamente la categoría a la que pertenece el patrón de entrada actual. Entonces, el Subsistema de Orientación deshabilita temporalmente el nodo J ($T_J = 0$) y selecciona la categoría con el siguiente nivel de activación de mayor valor.

9. Si el nivel de reset es menor que el parámetro de vigilancia, se compara la etiqueta de clasificación de la categoría ganadora con el dato $I^b(t)$ recibido en la Capa de Supervisión. Si estos valores no coinciden, la predicción es incorrecta dando como consecuencia que el parámetro de vigilancia se actualice automáticamente utilizando el valor del actual nivel de reset para buscar una nueva categoría, cancelando su nivel de activación $T_J = 0$.

10. Si el estado de clasificación del nodo ganador es el mismo que la etiqueta obtenida en la Capa de Supervisión se asume que la categoría activada es la que mejor representa la muestra de aprendizaje actual $I^a(t)$. Por consiguiente, se aplica un proceso de actualización de los pesos W_{ji}, C_{ji} y V_{ji} de la categoría seleccionada utilizando un aprendizaje gradual para aumentar su parecido con los datos de entrada. Esta actualización de los pesos se realiza con las ecuaciones de aprendizaje dinámicas del *S-dFasArt* (4.9). Los parámetros A_W, A_C, A_V, B_W and B_V se pueden representar como las velocidades de aprendizaje asociadas al crecimiento de los pesos mínimos, centrales y máximos respectivamente, de las categorías difusas.

$$\begin{aligned} \frac{dW}{dt} &= -A_W W + B_W min\left(I^a(t), W\right) \\ \frac{dC}{dt} &= A_C\left(I^a(t) - C\right) \\ \frac{dV}{dt} &= -A_V V + B_V max\left(I^a(t), V\right) \end{aligned} \tag{4.9}$$

4.4. Estimación del estado del usuario mediante lógica difusa y redes neuronales

En este apartado se comprueba si es factible utilizar el método neuro-difuso de clasificación propuesto para estimar el estado emocional del usuario a partir de las señales fisiológicas. Para abordar este punto, se ha realizado una experimentación donde se registran señales fisiológicas de diferentes usuarios, sin ningún tipo de problema cognitivo o físico, durante la realización de tres tareas que inducen a tres estados psico-fisiológicos bien diferenciados: relax, excitación media y estrés. Con estas medidas se entrena el método de clasificación neuro-difuso para comprobar sus posibilidades y rendimiento al clasificar este tipo de señales.

4.4.1. Metodología

Los experimentos se han dirigido de la misma manera que en (Badesa et al., 2014c). Durante las pruebas se ha utilizado una configuración hardware basada en un dispositivo robótico, un sistema de adquisición de señales y un sistema RV para realizar las actividades solicitadas y monitorear en tiempo real las señales fisiológicas del usuario (frecuencia de pulso, frecuencia respiratoria, SCL, SCR y la temperatura de la piel). El dispositivo robótico utilizado en estos experimentos ha sido el sistema robótico planar neumático llamado PUPArm, el cual se ha explicado en la Sección 3.8.2 de esta tesis. En la Figura 4.6 se pueden comprobar todos los elementos utilizados para completar el *setup* de la experimentación.

El sistema de RV ejecuta la actividad mostrada en la Figura 4.7 para inducir al usuario a los estados psico-fisiológicos comentados estableciendo diferentes niveles de dificultad. Esta actividad está formada por tres componentes principales: el área de trabajo la actividad delimitada por un marco negro, el puntero representado por un cuadrado verde que mueve el usuario y una serie de rectángulos azules de diferentes tamaños que se mueven de manera aleatoria a través de la pantalla. El objetivo de la actividad es mover libremente el puntero verde evitando la colisión con los rectángulos azules dentro del espacio de trabajo de la pantalla sin dejar el área delimitada por el marco negro. Cada vez que el sujeto toca un rectángulo azul o deja el área de la actividad significa un error, implicando que el puntero del usuario se vuelva rojo junto con la emisión de un sonido estridente. Existen

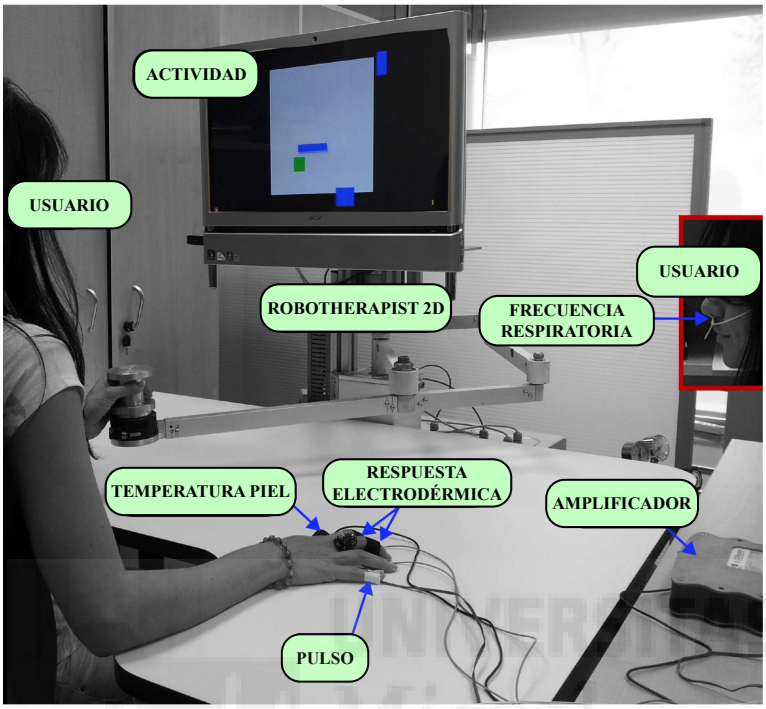


Figura 4.6: Un sujeto durante los experimentos.

tres niveles de dificultad dependiendo del número de rectángulos azules y sus velocidades: nivel de relax (uno con velocidad baja), normal (tres con velocidad media) y estrés (cuatro con velocidad alta).

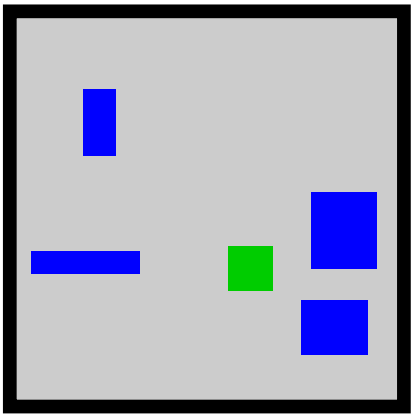


Figura 4.7: Actividad para inducir diferentes estados psicofisiológicos.

Los datos humanos presentados en esta tesis se han adquirido bajo un protocolo experimental aprobado por el Comité de Ética Médica de la Universidad Miguel Hernández de Elche y todos los sujetos dieron su consentimiento informado por escrito. Los siete usuarios estaban sanos, sin déficit cognitivos o físicos, y tenían edades comprendidas entre los 26 y 42 (edad media de 31 años, edad mediana de 29 años, desviación estándar de 6.3 años). Todo el protocolo se muestra en la Figura 4.8. Después de cada actividad, se le presentaba a los usuarios una prueba de Autoevaluación Self-Assessment Manikin (SAM) para medir las respuestas afectivas.

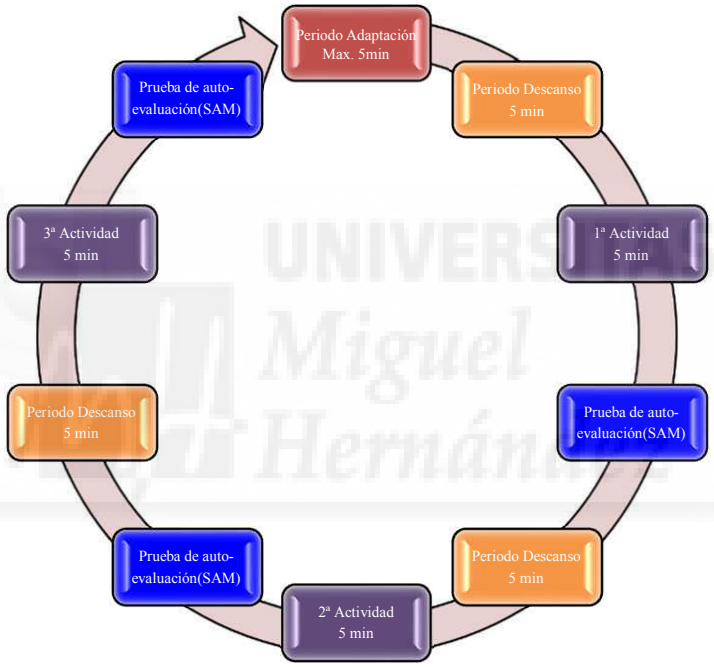


Figura 4.8: Protocolo Experimental: Este protocolo se divide en tres actividades y un periodo de descanso antes y después de las actividades.

Una vez que se adquieren las señales fisiológicas, se aplica una normalización de las características como se ha informado anteriormente en (Badesa et al., 2014c). Después de eso, se ha utilizado un procesamiento basado en un Analisis de Componentes Principales (PCA) con el fin de estudiar la posibilidad de reducir aún más el número de características de entrada para el modelo neuro-difuso propuesto, cuyos resultados se han comparado con diferentes algoritmos de aprendizaje automático utilizados en otros estudios para la clasificación de este tipo de señales .

4.4.2. Evaluación del clasificador neuro-difuso

En este apartado se explica el estudio de clasificación de las señales fisiológicas realizado para comprobar las posibilidades que ofrece el modelo neuro-difuso *S-dFasArt* utilizando la técnica de validación cruzada (Stone, 1974) llamada Leave-one-Out (LOOCV). Con este método de validación se examina la generalización de la red en situaciones no entrenadas y estima el rendimiento del clasificador. Esta técnica de validación es muy conveniente cuando los datos experimentales no contienen demasiadas medidas.

Con el objetivo de obtener un modelo funcional de clasificación se debe aplicar un proceso de ajuste del clasificador *S-dFasArt* presentando un conjunto de muestras de aprendizaje al algoritmo. Los datos de información del aprendizaje se obtienen a partir de la adquisición y el procesamiento de las respuestas fisiológicas de 7 usuarios como se ha comentado anteriormente.

El ajuste del clasificador se puede dividir en tres fases. Primero, se inicializan los parámetros que forman parte de las ecuaciones dinámicas del *S-dFasArt* con valores por defecto. En segundo lugar, se efectúa un aprendizaje de los pesos que representan las categorías difusas. Y por último, se aplica un fase de ajuste de parámetros para calcular los dos parámetros de la red más influyentes durante el proceso de clasificación de los datos y obtener los mejores valores de interpretación. Estos parámetros están relacionados con el carácter difuso σ de las categorías y su velocidad de activación A_T. Sin embargo, el parámetro A_T no se ajusta en esta experimentación debido a la naturaleza del método de validación, donde sólo se comprueba una muestra en cada iteración por lo tanto no es necesario realizar el cálculo continuo de las activaciones de las categorías. Por tanto, A_T mantiene su valor por defecto constante durante todo el aprendizaje.

Antes de empezar el entrenamiento de la red, se debe establecer el parámetro A_R para controlar el número de categorías que se generan dentro de la red neuronal. Con este fin, se realiza un estudio cuantitativo de las categorías creadas por el algoritmo a partir de un rango de valores de A_R que permita una generación de categorías razonable para la cantidad de muestras de aprendizaje presentadas a la red neuronal. Así, se limita la creación de categorías evitando una generación excesiva, ya que un sobreentrenamiento puede provocar el problema de la proliferación de categorías que *S-dFasArt* hereda de la arquitectura ARTMAP dando como consecuencia la anulación de la generalización de las categorías. De esta manera se evita la aparición de una categoría por cada muestra de entrada.

De manera inicial, se define un vector de valores desde 0 hasta 20 (con un intervalo de 0.1 entre datos) del parámetro A_R con el objetivo de calcular las categorías que pueden generarse con la arquitectura S-dFasArt utilizando este tipo de características. El número de categorías comprometidas no se prefija de antemano en los modelos basados en la arquitectura ARTMAP, sino que este valor depende del proceso de aprendizaje. En *S-dFasArt*, las categorías se generan en función del número de muestras de aprendizaje proporcionadas al modelo. Por cada valor de A_R se ha implementado un modelo de clasificación aplicando los valores por defecto de la red y todas las muestras de aprendizaje. Después, se calcula el número de categorías de todos los modelos. En la Figura 4.9 se muestra la generación de categorías difusas en una gráfica con el número de nodos creados en función del valor de velocidad de crecimiento A_R del nivel de reset.

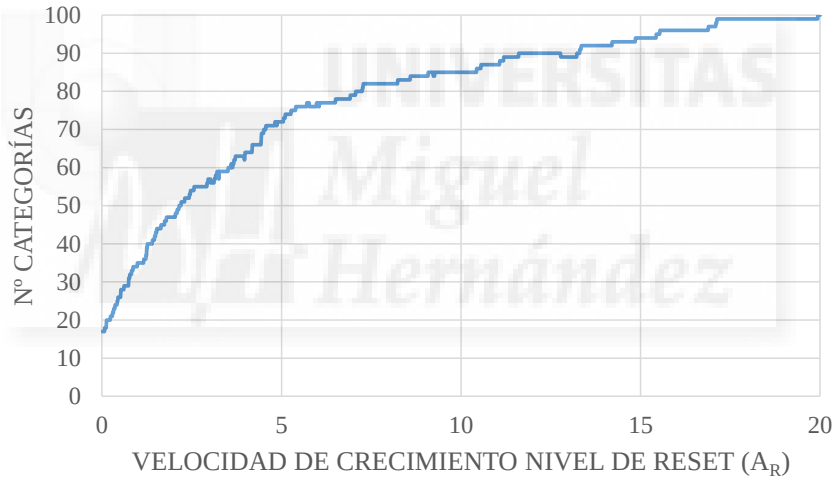


Figura 4.9: Proliferación de categorías.

En la gráfica se puede observar un incremento en la cantidad de categorías debido al crecimiento del valor del parámetro A_R. Por consiguiente, se selecciona un rango de valores de A_R que permite obtener un número de categorías entre 20 y 40 para evitar una generación excesiva de categorías que pueda provocar una incorrecta clasificación de las muestras. Se ha elegido este rango de valores puesto que la base de datos contiene 105 muestras, 15 por usuario. Entonces, el rango de valores de A_R se ha establecido entre 0.05 y 2.15.

Con el método de validación elegido (LOOCV), se generan K-modelos de clasificación dependiendo de la cantidad de medidas de aprendizaje que se han extraído de las señales fisiológicas (en este caso 105 muestras). En cada iteración del método, se crea un modelo de clasificación utilizando todas las muestras de aprendizaje a excepción de una medida que se retira del conjunto de aprendizaje. Esta medida aislada se clasifica durante la fase de test con la red ya entrenada. El conjunto de datos restante se utiliza en la fase de aprendizaje de los pesos durante el proceso de ajuste del clasificador. Este procedimiento se repite una vez para cada medida, entonces se calcula la media aritmética de todo el proceso iterativo para obtener el porcentaje de éxito de clasificación de la red.

El siguiente paso consiste en obtener el conjunto de valores de A_R y σ que ofrece mejores resultados de acierto utilizando el método de ensayo y error. Como el rango de G-valores del parámetro A_R ya se ha configurado anteriormente, para σ se establece un rango de L-valores entre 0.0001 y 1.0, así, este parámetro cubre un amplio rango de tamaños difusos de la función triangular de activación-pertenencia triangular. De esta manera, el método de validación LOOCV se ejecuta $G * L$ veces en función del rango de valores de A_R-σ. Por consiguiente, el resultado con el mayor porcentaje de acierto aporta los mejores valores del conjunto de parámetros A_R-σ.

El valor del resto de los parámetros también se obtienen a partir del método de ensayo y error, sin embargo se mantienen constantes durante todo el proceso de validación debido a que los valores por defecto proporcionan la generación de los modelos con mejores tasas de clasificación. En la Tabla 4.1 se muestran los valores aplicados durante el proceso de ajuste y de validación del clasificador, junto con una breve descripción de los parámetros.

PARÁMETRO	**VALOR**	**DESCRIPCIÓN**
A_R	0.05-2.15	Velocidad de crecimiento del nivel de reset
σ	0.0001-1.0	Carácter difuso de las categorías
A_T	0.01	Velocidad de activación de las categorías difusas
A_W	0.8	Velocidad de crecimiento de los pesos asociados a las categorías
A_C	0.8	
A_V	0.1	
ϵ	0.001	Valor mínimo del carácter difuso
α	1e-30	Valor de activación para generar nuevas categorías
$h(\rho)$	0.1	Parámetro de vigilancia
R_{max}	0.2	Valor máximo de reset para deshabilitar las categorías
B_T, B_R B_W, B_V	1	Ganancias de las ecuaciones diferenciales del S-dFasArt

Tabla 4.1: Descripciones y variables dinámicas del algoritmo *S-dFasArt*.

Una vez procesados todos los modelos de clasificación, y probados todos los conjuntos de valores de los parámetros A_R y σ, se han obtenido 11130 posibles resultados. En la Figura 4.10 se muestra una gráfica 3D que recolecta todos los valores de las tasas de éxito de todos los modelos de clasificación generados. La zona marcada del gráfico son los puntos cuyos valores de tasas de acierto superan el 90 %. Entonces, se puede extraer un plano (Figura 4.11) de la gráfica 3D con los valores de los parámetros A_R y σ con un porcentaje superior al 90 % para una mejor observación.

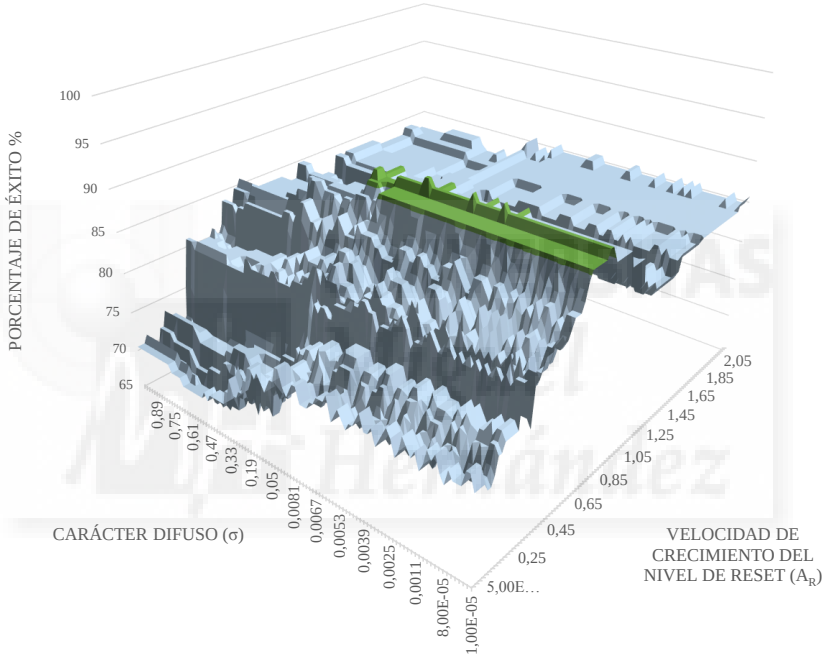


Figura 4.10: Porcentajes de éxito en función de A_R y σ.

Sin embargo, para obtener el mayor porcentaje de éxito se seleccionan los valores situados en los picos máximos de la gráfica. De esta manera, se cumple la tercera fase del proceso de ajuste del clasificador, obteniendo el conjunto de valores de los parámetros A_R y σ que proporcionan mejor tasa de acierto. En la Tabla 4.2 se recogen los resultados finales obtenidos durante el desarrollo de esta experimentación. De los 11130 posibles modelos, se han seleccionado los cinco modelos de clasificación que han obtenido los mejores resultados en términos de tasa de éxito. La primera columna muestra las categorías generadas por las iteraciones del método de validación. Las

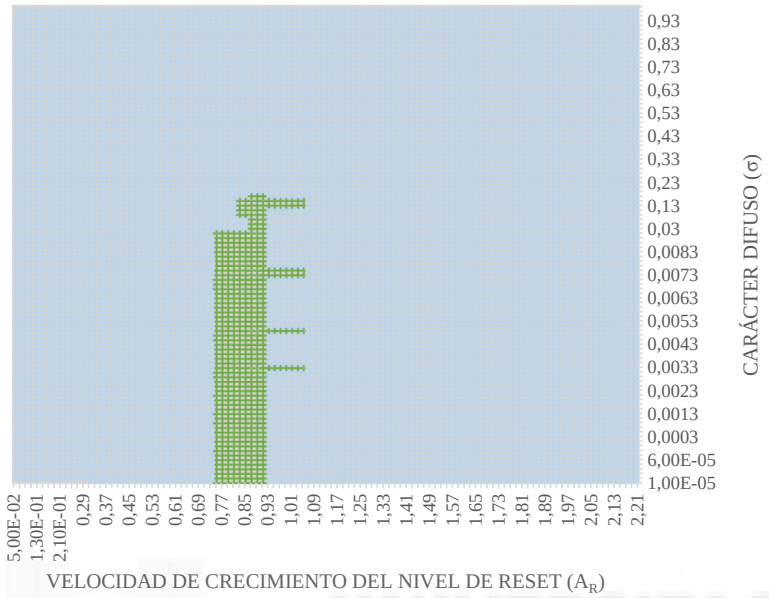


Figura 4.11: Plano con los valores de A_R y σ.

siguientes dos columnas recolectan los valores de los parámetros ajustados durante la fase de ajuste mientras que la tasa de éxito se muestra en la última columna. En esta tabla también se puede observar la variabilidad de las tasas de acierto en función de los parámetros A_R y σ.

Categorías	A_R	σ	Acierto %
33-34	0.87	0.0033	92.38
33-34	0.89	0.0083	91.43
30-31	0.75	0.01	90.48
34-35	0.99	0.17	89.52
28-29	0.69	0.05	87.62

Tabla 4.2: Tabla resumen con los mejores resultados de acierto.

De manera complementaria, se ha realizado una comparación de los rangos de clasificación del modelo neuro-difuso S-dFasArt con algunos clasificadores utilizados en (Badesa et al., 2014c) para interpretar este tipo de señales, creando modelos de nueve técnicas de aprendizaje automático con el mismo conjunto de medidas de entrenamiento y la misma técnica de validación. Los resultados comparativos obtenidos con LOOCV se muestran en la Tabla 4.3. Esta tabla se distribuye en seis columnas para mostrar los valores de clasificación aplicando un análisis PCA a los datos de entrada y

los resultados sin emplear algún procesamiento complementario. En las primeras cinco columnas se presenta el rendimiento de cada una de las nueve técnicas de aprendizaje automático y el modelo neuro-difuso propuesto utilizando un número diferentes de Componentes Principales (PC) como datos de entrada, mientras que en la última columna se muestran los resultados sin aplicar PCA.

ALGORITHM	PCA 1 PC	PCA 2 PC	PCA 3 PC	PCA 4 PC	PCA 5 PC	NO PCA
PLA	56.57	81.9	83.5	82.86	82.1	83.05
LR	61.90	65.71	84.76	83.81	85.71	85.71
LDA	50.48	64.76	74.29	74.29	76.19	76.19
QDA	49.52	63.81	75.24	78.10	78.10	78.10
SVM	60.00	67.62	86.67	85.71	85.71	85.71
SVM with RBF	78.10	80.00	91.43	91.43	91.43	91.43
NB	49.52	61.90	66.67	64.76	60.00	53.33
KNN	64.76	72.38	80.95	80.95	80.95	80.95
RBF	58.10	57.05	56.10	56.10	56.29	56.19
S-dFasArt	69.52	80.95	90.48	90.48	90.48	92.38

Tabla 4.3: Comparación LOOCV de los métodos de clasificación.

4.5. Interfaz auto-adaptativa para terapias virtuales en rehabilitación robótica

El sistema de interacción definido en este trabajo se basa de acuerdo con (Novak et al., 2012) en la adaptación del nivel de dificultad a partir del estado psico-fisiológico del usuario, de una actividad 3D implementada con el motor de tareas desarrollado en el Capítulo 3, así como en la adaptación automática proporcionada por el tipo de clasificador analizado en este capítulo. Como se pudo comprobar en el apartado anterior, el método de clasificación encargado de realizar la estimación del estado tiene un buen rendimiento para su utilización en esta integración debido a que supera el 90 % de acierto, una tasa bastante elevada para obtener un sistema en tiempo real bastante efectivo. La aplicación final está orienta a una terapia auto-adaptativa de neuro-rehabilitación asistida por dispositivos robóticos.

En la Figura 4.12 se muestra el diagrama general con el sistema de interacción auto-adaptativo implementado. Está formado por cuatro bloques funcionales: el sistema de registro y procesado de señales fisiológicas, el método de clasificación que decide el estado actual de las señales registra-

das, el sistema de realidad virtual encargado de ejecutar la tarea virtual con el ajuste de los elementos cuando se cambia de nivel de dificultad y la asistencia proporcionada por el dispositivo robótico. Los sensores se conectan al usuario para registrar sus señales fisiológicas mientras controla el sistema hardware de interactuación. El sistema hardware de interacción está formado por el dispositivo de rehabilitación robótico llamado PUPArm y por el subsistema de realidad virtual. Por consiguiente, el usuario obtiene tres tipos de realimentación: visual, sonora y de fuerzas.

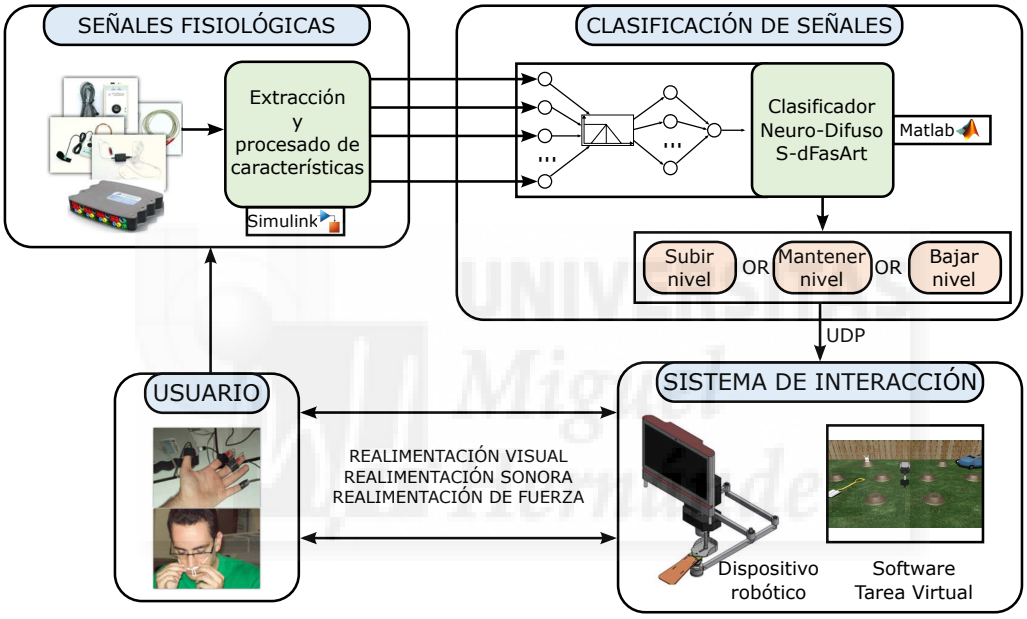


Figura 4.12: Diagrama del sistema auto-adaptativo.

Cuando el usuario controla el sistema robótico, se empiezan a registrar y procesar sus señales fisiológicas para extraer las características más importantes. Estas características se envían al modelo funcional establecido por el clasificador neuro-difuso. El bloque de clasificación es capaz de generar 3 posibles estados a partir de este procesamiento en función de la información de entrada, comprobando el estado psico-fisiológico, y decide el tipo de modificación que debe aplicar en la tarea (subir nivel, bajar nivel o mantener nivel). El estado de clasificación obtenido se envía al sistema de interacción para modificar automáticamente el nivel de dificultad de la tarea virtual descrita en la Sección 3.7.3 llamada "golpear elementos". Estas tareas virtuales reciben el comando generado por el clasificador y automáticamente

actualizan el nivel de dificultad cambiando los tiempos, la velocidad y el número de elementos que forman parte del escenario virtual. En este caso se definieron 6 posibles niveles modificando el número y la intensidad de aparición de los elementos de interactuación de la tarea virtual con el usuario:

- Nivel más relajado: Aparece un conejo y se mantiene en escena durante 10 segundos.
- Aparece un conejo y se mantiene en escena durante 5 segundos.
- Nivel normal: Dos conejos aparecen al mismo tiempo y se mantienen en pantalla durante 10 segundos.
- Aparecen dos conejos de manera asíncrona y permanecen visibles durante 5 segundos.
- Tres conejos aparecen al mismo tiempo en escena durante 10 segundos.
- Nivel más estresante: Aparecen tres conejos de manera asíncrona y cada uno se mantiene en escena durante 5 segundos.

El objetivo de este tipo de sistema es mantener siempre un nivel de dificultad intermedio durante el periodo de ejecución de la terapia virtual para maximizar la motivación y participación del paciente, permitiendo un nivel de atención medio. De esta manera se evita una terapia relajada o un estado de estrés continuo, hecho que da como consecuencia el abandono de dicha terapia por falta de interés o por estrés elevado.

La extracción y procesamiento de las características de las señales se realiza con un esquema de Simulink, mientras que el bloque de clasificación se ha diseñado con Matlab. Para la comunicación entre el bloque de clasificación y el software de realidad virtual de la tarea se ha utilizado un protocolo UDP. Esta comunicación tiene lugar cada 30 segundos para enviar uno de los tres posibles comandos de acción y cambiar el nivel de dificultad de la tarea virtual:

1. Si el usuario tiene un nivel de estrés se reduce un nivel de dificultad.
2. Si el usuario tiene un estado estable, no se realiza ningún cambio de nivel de dificultad.

3. Si el usuario está relajado, se incremente el nivel de dificultad actual.

Antes de utilizar el sistema, se registran las señales durante 5 minutos en un estado de relax para obtener una medición de referencia de los usuarios, los cuales deben completar un periodo de adaptación de unos pocos minutos. Finalmente, los usuarios pueden realizar la tarea virtual en sesiones de 10 minutos o en función de lo que considere el terapeuta rehabilitador, empezando en el primer nivel de dificultad y ajustándose durante los diferentes estados emocionales del usuario.

4.6. Conclusiones

La hipótesis principal presentada en este capítulo en lo que respecta a la estimación del estado fisiológico del paciente a partir de un modelo de clasificación neuro-difuso se basa en el modelo conceptual propuesto por (Pessoa, 2008) sobre cálculos neuronales y emociones. Bajo esta suposición, Feng planteó en CRAA una comparación entre una red cognitiva, una red afectiva y una red de diagnóstico. La red afectiva se desarrolló utilizando una red neuronal basada en modelos ART. Este punto ha sido uno de los pilares que sostienen la hipótesis: "*Las redes neuronales basadas en ART deberían funcionar mejor que los clasificadores implementados en trabajos previos, ya que son ampliamente utilizados para modelar procesos neuronales relacionados con las emociones*".

En este capítulo se han presentado los resultados obtenidos al clasificar reacciones fisiológicas de usuarios durante ejercicios de rehabilitación asistidos por dispositivos robóticos mediante un modelo ART llamado *S-dFasArt*, el cual se basa en redes neuronales combinadas con sistemas de lógica difusa. Además, estos resultados se han comparado con nueve técnicas de aprendizaje automático. Los mejores resultados en términos de precisión (92.38 % con LOOCV) se han obtenido con el enfoque ofrecido por *S-dFasArt* seguidos por el modelo SVM con RBF (91.43 % con LOOCV), como se ha podido comprobar en la Tabla 4.3. A partir de estos resultados se puede decir que la combinación de la naturaleza dinámica del algoritmo *dFasArt* junto con un módulo supervisado produce un clasificador robusto capaz de proporcionar muy buenos resultados a pesar de tener un conjunto reducido de datos de entrada.

El algoritmo propuesto ha sido aplicado en estudios anteriores a problemas relacionados con la clasificación de señales variantes en el tiem-

po con una alta contaminación de ruido, en la clasificación de datos GPS (Toledo-Moreo et al., 2010) para la gestión de vehículos o en señales electroencefalográficas (Cano-Izquierdo et al., 2012). Las señales de este tipo de aplicaciones poseen mucho ruido y sus cambios temporales son unas características relevantes que se necesitan analizar, como las bio-señales utilizadas para modelar el estado emocional a partir de reacciones fisiológicas son muy ruidosas. Por esta razón se ha decidido probar este tipo de arquitectura.

S-dFasArt permite una adaptación entre su complejidad o número de categorías y la base de datos utilizada durante su proceso de aprendizaje, es decir, la complejidad incrementa en función de la variabilidad de los datos de aprendizaje. Este hecho implica que no se asumen premisas iniciales y cuanto más grande es el conjunto de datos el modelo genera más categorías, dando como resultado un sistema más robusto. Además, este algoritmo puede hacer frente a los datos contradictorios e inconsistentes que producen graves problemas serios en los mecanismos de optimización de otros tipos de modelos de clasificación.

En este capítulo también se ha realizado un análisis del rendimiento del algoritmo *S-dFasArt* al aplicar un procesamiento PCA (Tabla 4.3). Estos resultados indican que el proceso de reducción de las características de los datos de entrada no proporciona ninguna mejora a la hora de clasificar. Por lo tanto, este método se debe utilizar sin procesamiento PCA. Por otra parte, parece que el *S-dFasArt* tendría mejores resultados con un conjunto de datos mayor, y sería más robusto utilizando datos de entrada ruidosos que el enfoque SVM con RBF. Si se corrobora este supuesto, el enfoque presentado se puede considerar como el mejor candidato para clasificar reacciones fisiológicas de usuario en terapias de rehabilitación asistidos por robots.

Existen varias razonas para usar este tipo de técnica de clasificación para estimar el estado emocional del usuario: una de estas razones es su rendimiento frente a datos con una alta cantidad de ruido; otra es su capacidad de ajuste para actualizar los modelos de clasificación y los parámetros de las redes en tiempo real. De esta manera, el método de clasificación se puede adaptar automáticamente a las demandas y necesidades específicas de cada paciente. Incluso, es capaz de permitir la adquisición de nuevos ejemplos de aprendizaje sin borrar el conocimiento acumulado.

Una vez evaluado el rendimiento del método de clasificación neuro-difuso con este tipo de señales, obteniendo mejores resultados en términos de precisión que las nueve técnicas de aprendizaje automático (92.38 % en LOOCV), se ha implementando un sistema de interacción con el objetivo de ofrecer

una terapia complementaria de rehabilitación que se adapte automáticamente a las necesidades específicas de cada usuario modificando el nivel de dificultad de las tareas virtuales. La alta tasa de eficiencia en la clasificación permite generar un sistema de actuación en tiempo real capaz de visualizar tareas virtuales con un alto grado de realismo gráfico y físico, hechos que incrementan la sensación de inmersión en el entorno virtual generando una participación más activa por parte del usuario.

UNIVERSITAS
Miguel
Hernández

Capítulo 5

Evaluación de la influencia de la visualización en la función sensoriomotriz durante terapias de rehabilitación

En este capítulo se expone un estudio acerca de la influencia de la visualización de tareas virtuales 2D y 3D en el rendimiento sensorimotor durante terapias de neuro-rehabilitación asistidas por dispositivos robóticos. El propósito de este estudio es evaluar si existen diferencias en los patrones cinemáticos de movimiento cuando pacientes con ACV realizan una tarea de alcance visualizando un juego terapéutico virtual con dos diferentes tipos de representación del entorno. De esta manera se pueden identificar parámetros tales como la velocidad máxima, el tiempo de reacción, la distancia total o la distancia inicial a partir de la adquisición de datos objetivos por parte del dispositivo robótico para evaluar si el tipo de visualización puede

influir en la calidad de percepción de la tarea y comprobar si afecta a la función sensoriomotriz del miembro superior.

En este estudio se ha utilizado como herramienta un sistema robótico y dos tareas de alcance para proporcionar una terapia de neuro-rehabilitación virtual a una serie de pacientes con ACV durante un cierto número de sesiones. Para finalizar, se analizan los datos obtenidos y se muestran los resultados de una encuesta de usabilidad donde se determina el nivel del satisfacción de los pacientes con respecto al sistema global.

5.1. Introducción

La neuro-rehabilitación de ACV basada en terapias virtuales se realiza completando ejercicios repetitivos monitoreados por dispositivos electrónicos visuales, cuyo contenido representa tareas imaginarias o AVD. Actualmente, existen dos maneras diferentes de visualizar estas tareas: entornos virtuales en 3D, cuyo realismo viene determinado por la resolución y la fidelidad de los elementos de la tarea; y los entornos virtuales en 2D que representan tareas con un reducido grado de realismo. Sin embargo, el tipo de visualización puede influenciar en la calidad de la percepción de la tarea afectando al rendimiento sensoriomotriz del paciente. Los recientes desarrollos en la tecnología robótica han ayudado a realizar un análisis más objetivo y fiable de las terapias que se aplican a pacientes con lesiones neurológicas (Badesa et al., 2012; Badesa et al., 2014c; Badesa et al., 2014a). Esto se debe a la capacidad de registrar datos cinemáticos y cinéticos por parte de este tipo de dispositivos para extraer cientos de marcadores que permiten cuantificar el proceso de recuperación motor durante la terapia (Volpe et al., 2009; Einav et al., 2011; Bertomeu-Motos et al., 2015; Papaleo et al., 2015).

Actualmente se está incrementando el uso de sistemas RV más complejos en las terapias de neuro-rehabilitación asistidos por dispositivos robóticos. Por consiguiente, la combinación de sistemas robóticos para una rehabilitación neuro-motora con sistemas RV obtiene las ventajas de ambas técnicas, tales como: aumentar la motivación del paciente, mejorar la variabilidad y adaptabilidad, almacenamiento transparente de los datos proporcionados por el sistema robótico y el sistema RV por separado, registro online de los datos para una verificación remota e incluso tener la posibilidad de replicar cualquier entorno de la vida diaria sin necesidad de tener la representación física del entorno. Con esta metodología, se pueden obtener un tratamiento más eficaz para llevar a cabo una mejor recuperación del paciente (González et al., 2015).

En lo relativo a la RV existen dos cuestiones importantes: una está relacionada con la forma en que el usuario percibe el entorno virtual utilizando diferentes plataformas de visualización, y la otra tiene que ver con el modo de representación del contenido gráfico. En cuanto a la primera cuestión, existen diferentes plataformas de visualización como pueden ser los

monitores de ordenador, Gafas de Realidad Virtual (HMD) o Sistemas de Proyección en Pantalla Grande (SPS). Cada plataforma tiene una manera particular de aplicar las terapias virtuales, teniendo en cuenta los objetivos terapéuticos, y pueden proporcionar diferentes beneficios adecuados para las necesidades del paciente. En (Rand et al., 2005) se comparan los efectos de visualizar el mismo entorno virtual a través de un HMD (plataforma 3D) y un monitor de computador (plataforma 2D) en sujetos jóvenes y mayores. Además, en (Subramanian y Levin, 2011) se evalúa el rendimiento motor con respecto a los movimientos cinemáticos realizados por sujetos sanos y pacientes mientras visualizaban un entorno virtual con gráficos 3D a través de un HMD y un SPS (plataforma 2D). En ambos estudios, se han obtenido mejores resultados al mostrar el entorno virtual en la plataforma de visualización 2D, en un monitor de computador y un SPS respectivamente. Sin embargo, estos estudios se han enfocado en la plataforma de visualización presentando el mismo entorno sin tener en cuenta el modo de representación del contenido gráfico.

Con respecto al segundo tema comentado anteriormente sobre el contenido gráfico, existen estudio basados en sistemas RV con entornos implementados con gráficos 2D y otros con gráficos 3D. Por ejemplo en (García-Betances et al., 2015) se ha realizado una descripción actual de la tecnología RV para aplicaciones destinadas al Alzheimer, y estos sistemas utilizan visualizaciones de gráficos 2D convencionales o gráficos 3D indistintamente. Del mismo modo ocurre con la rehabilitación del daño cerebral en (Rose et al., 2005), y estudios en pacientes que han sufrido un ACV tales como (Merians et al., 2006; Saposnik, 2016; Henderson et al., 2007). Por lo tanto, existe un amplio panorama en la literatura científica con respecto a la rehabilitación virtual. Sin embargo, todavía no se han realizado estudios objetivos que comparen como afecta la visualización de gráficos 2D y entornos virtuales 3D a la percepción del movimiento en sujetos después del ACV. Esto significa, que no existen evidencias que muestren si es mejor o no realizar tareas de rehabilitación virtual implementadas con gráficos 2D o 3D. La utilización de gráficos 3D puede aumentar la percepción visual de los elementos virtuales, de tal manera que las tareas basadas en actividades de la vida diaria son más similares a la realidad. Mientras que los gráficos 2D permiten una representación más simple de las tareas. Estas dos perspectivas se deben probar para evaluar qué tipo de representación visual proporciona una mejor calidad del rendimiento motor en términos de movimientos cinemáticos. Esta evaluación puede llevarse a cabo cuando

el sujeto realiza el mismo movimiento para completar los mismos objetivos utilizando ambos tipos de visualización. Por lo tanto, los dispositivos robóticos se pueden utilizar para restringir este movimiento y extraer objetivamente datos cuantitativos, por lo que las terapias se podrían adaptar a cada paciente (Morales et al., 2014).

En este capítulo, se evalúan los efectos de aplicar juegos terapéuticos con 2D o 3D en terapias virtuales asistidas por dispositivos robóticos y se analizan los resultados. De esta manera, se proporcionan datos cuantitativos que permiten evaluar la influencia de la terapia virtual y valorar qué tipo de entorno virtual se ajusta mejor a cada paciente en términos de facilidad de uso y comodidad. Por lo tanto, el objetivo principal de este estudio es determinar si existen diferencias en los parámetros cinemáticos de movimiento registrados por el dispositivo robótico que permiten evaluar el rendimiento motor del paciente en tareas virtuales 2D y 3D. Para ello, se han diseñado dos tareas virtuales modificando el nivel de inmersión a partir de gráficos bidimensionales y tridimensionales, pero manteniendo el mismo objetivo cinemático.

5.2. Materiales y métodos

5.2.1. Pacientes

El estudio se ha realizado en el hospital de atención a pacientes crónicos y de larga estancia de la Pedrera en Dénia, y han participado nueve pacientes de DCA (edades comprendidas entre 40 y 70 años). El Comité de Ética Médica de la Universidad Miguel Hernández aprobó el protocolo experimental del estudio propuesto. El equipo médico de dicho hospital fue el responsable de incluir pacientes que estaban recibiendo tratamiento de fisioterapia y terapia ocupacional, e informó debidamente a todos los pacientes y ellos dieron su consentimiento por escrito antes de iniciar el estudio indicando que entendían el propósito y los requisitos. Los criterios de inclusión fueron: adultos con hemiparesia/hemiplejía secundaria con ictus en fase subaguda (entre 1 y 6 meses después de la lesión). Los criterios con respecto a la función motora musculares de la extremidad superior fueron:

1. Tono muscular con puntuación por debajo de 2 en la Escala Modificada de Ashwoth (Bohannon y Smith, 1987).

2. Balance equilibrio en la abducción del hombro y flexión del codo con base del Índice Motor (MI) ≥ 2 (Collin y Wade, 1990).

En el proceso de selección, se evitó la inclusión de pacientes con las siguientes lesiones: hombro doloroso, apraxia, sin control de tronco en sedestación, diagnósticos con efecto sobre la mano (como artritis u otras enfermedades reumatológicas), déficits perceptivos graves, ictus de la circulación posterior (sistema vertebrobasilar) y déficits lingüísticos que le impidan una comunicación útil. Las principales características de los participantes en el estudio y sus diagnósticos clínicos se muestran en la Tabla 5.1.

Paciente	Sexo	Edad	Diagnóstico	Tipo	Localización	Lateralidad
1	Mujer	69	Mielitis isquémica	Mielopatía	Medular	Tetraparesia
2	Hombre	40	Hemorragia ganglios basales	Hemorrágico	Ganglios basales	Derecha
3	Hombre	41	Hematoma glioblastoma	Hemorrágico	Ganglios basales	Izquierda
4	Hombre	46	Infarto indeterminado ACM	Isquémico	Parietal	Izquierda
5	Mujer	66	Infarto protuberancial	Isquémico	Troncoencéfalo	Izquierda
6	Mujer	41	Hemorragia parietal	Hemorrágico	Parietal	Derecha
7	Hombre	53	Ictus indeterminado ACM	Isquémico	Parietal	Derecha
8	Mujer	41	Hemorragia Cerebral	Hemorrágico	Frontal	Izquierda
9	Mujer	46	Hemorragia Cerebral	Hemorrágico	Frontal	Izquierda

Tabla 5.1: Características clínicas de los participantes del estudio.

5.2.2. Sistema de neuro-rehabilitación

El estudio propuesto se ha llevado a cabo utilizando el dispositivo robótico presentado en la Sección 3.8.2 para realizar la terapia motora y obtener toda la información objetiva de los movimientos. En este caso, el sistema actúa con movimiento libre para que los usuarios puedan realizar cualquier trayectoria y verificar la existencia de diferencias cinemáticas al visualizar tareas en 2D y tareas en 3D. El sistema es capaz de registrar información sobre el progreso del paciente en la rehabilitación, con base a parámetros como la posición, la velocidad y las fuerzas de interacción.

5.2.3. Tareas virtuales

En este apartado se definen las tareas virtuales planteadas para el estudio. La tarea virtual con el entorno 2D consiste en una ruleta formada por un objetivo central y ocho objetivos periféricos. Estos objetivos son círculos con un radio de 1 cm. Los ocho objetivos periféricos se distribuyen uniformemente en una circunferencia alrededor del objetivo central a 10 cm de

distancia. El propósito principal de esta tarea es alcanzar uno de los ocho objetivos periféricos iniciando el movimiento desde el objetivo central a través del control del efector final del robot fijado a la mano del sujeto mientras el correspondiente objetivo permanece iluminado a modo de señalización. Para completar estos objetivos, la tarea se visualiza en el monitor del sistema de neuro-rehabilitación con un refuerzo visual representado por un círculo blanco de 1 cm de radio para indicar la posición actual del efector final del robot. En la Figura 5.1 se muestra una captura de pantalla con el entorno 2D de la tarea virtual junto con información estructural. Algunos estudios previos han utilizado esta tarea virtual para determinar la influencia de la edad en la función sensorimotora de la extremidad superior (LLinares et al., 2013) o en casos especiales de trastornos neurológicos (Badesa et al., 2014b).

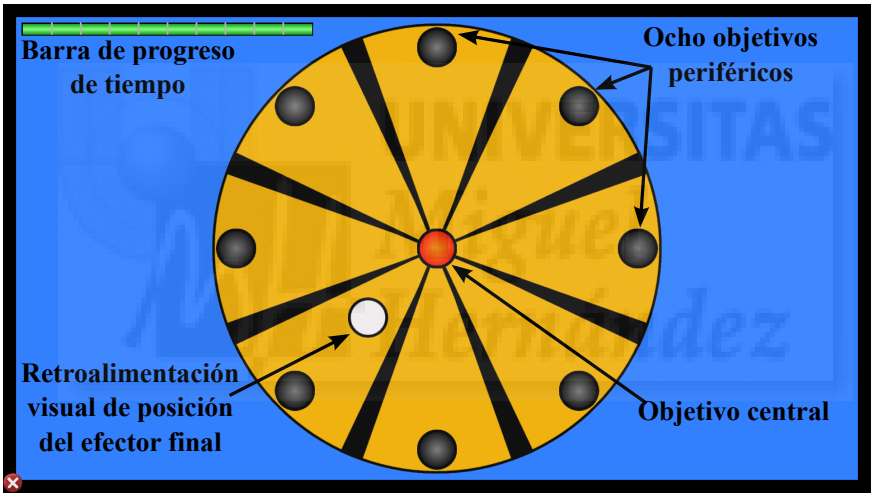


Figura 5.1: Escenario de la tarea 2D.

En cuanto al modo de representación en 3D se ha utilizado la tarea virtual explicada en el apartado 3.7.4. Esta tarea ha seguido los mismos criterios de objetivos utilizados en la Ruleta 2D con el fin de realizar el mismo tipo de movimientos. El escenario gráfico consta de ocho plataformas y un depósito central para representar los ocho objetivos periféricos y el objetivo central de la Ruleta 2D. La Figura 5.2 muestra una captura de pantalla con el entorno 3D de la tarea virtual junto con información estructural.

Básicamente, el propósito de esta tarea es la misma que la tarea 2D, pero con un nivel de visualización diferente. El flujo de trabajo para completar el objetivo principal de la tarea virtual 3D es:

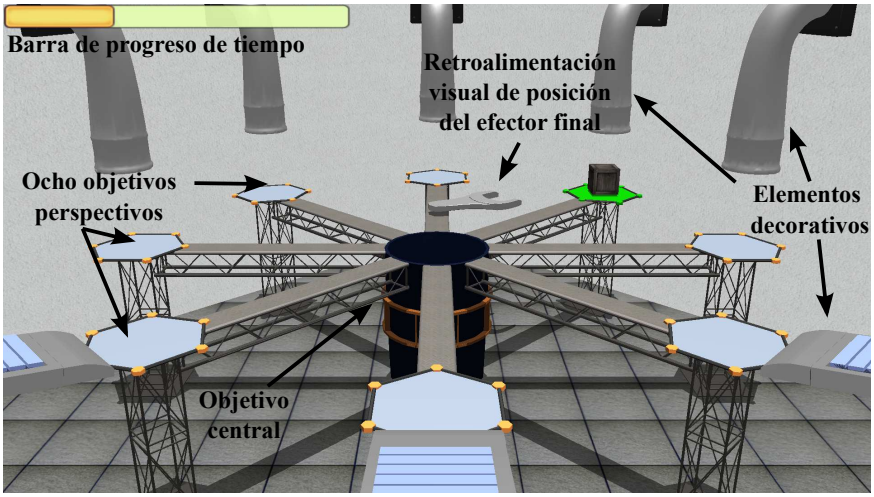


Figura 5.2: Escenario de la tarea 3D.

1. En primer lugar, el usuario debe aproximar la herramienta virtual al depósito central para iniciar la visualización de la posición del siguiente objetivo perspectivo.

2. Entonces, aparece aleatoriamente una caja en cualquiera de las ocho plataformas. La plataforma se ilumina como soporte visual, y la llave inglesa virtual se orienta de forma dinámica al objetivo posicional.

3. El usuario dispone de un tiempo limitado para recoger la caja objetivo. Este tiempo se muestra en una barra de progreso colocado en el lateral superior izquierdo de la pantalla. Si no se alcanza el objetivo, la caja desaparece y se ejecuta el siguiente objetivo.

4. La caja se selecciona cuando la llave virtual está cerca del objetivo. A continuación, el usuario tiene que llevar la herramienta virtual al depósito central para soltar la caja. También se ha incorporado un soporte sonoro para indicar que el objetivo se ha completado.

La funcionalidad y la estructura de ambas tareas virtuales son las mismas. Esto es necesario para una comparación objetiva de los valores de los parámetros obtenidos por el dispositivo robótico, ya que se examina la misma situación pero con diferentes estímulos externos. En la Figura 5.3 se presenta una correlación estructural entre las tareas 2D y 3D. El propósito

de desarrollar un escenario tridimensional es proporcionar una correlación más natural entre el movimiento del robot y la vista del usuario. En tareas 2D, cuando el usuario se acerca o se aleja el efector final con respecto del cuerpo, el elemento controlable en la tarea se mueve hacia arriba o hacia abajo en la pantalla, sin embargo, muchos pacientes tratan de forzar el efector final hacia arriba o hacia abajo para poner el elemento controlable en su lugar correspondiente en la pantalla. Por lo tanto, asociar el movimiento planar horizontal del dispositivo robótico con el movimiento vertical en la tarea 2D que aparece en pantalla, puede causar confusión en algunos pacientes. De esta manera, la tarea 3D replica en la pantalla el mismo tipo de movimiento del robot.

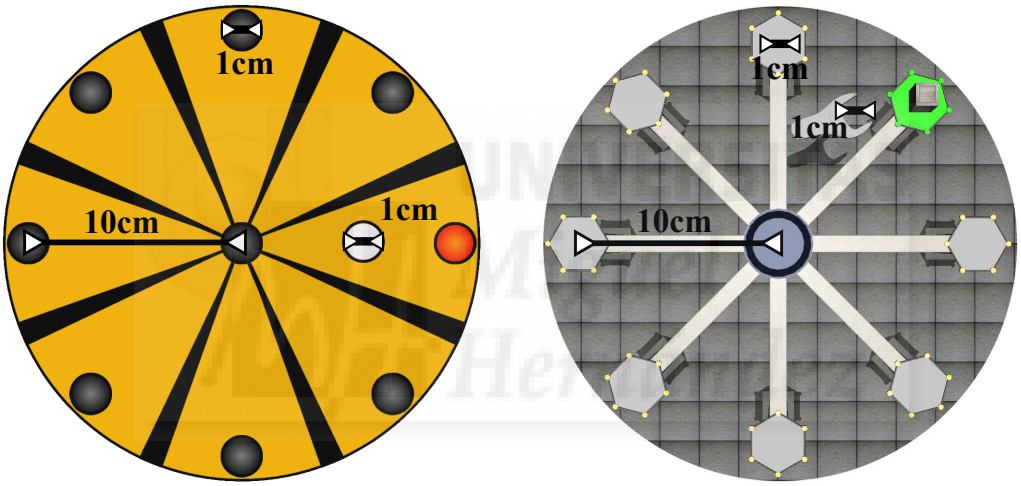


Figura 5.3: Correlación estructural entre las tareas 2D y 3D (vista cenital).

5.2.4. Preparación y protocolo

El grupo de estudio ha recibido el tratamiento de la terapia asistida por el robot PUPArm durante un período de dos meses con cuatro sesiones semanales de diez minutos, 36 sesiones en total. En la primera sesión, se lleva a cabo una evaluación general de la colocación del paciente para obtener parámetros como la altura de la pantalla o de la silla, y su rango de movilidad. De esta manera, se consigue el rango funcional máximo del paciente específico. Antes de iniciar la sesión, el paciente se coloca en frente del dispositivo en una posición cómoda con los parámetros obtenidos en la

primera sesión. El monitor encargado de ofrecer la retroalimentación visual se encuentra a 70 cm del paciente. Cada sesión se estructura en dos bloques de movimientos de entrenamiento dependiendo de la tarea virtual. Entre cada bloque, los pacientes tenían períodos de descanso de tres minutos. El tiempo de la sesión se organiza de la siguiente manera:

- Al comienzo de la sesión se muestra al azar una de las dos tareas para realizar el primer bloque de movimientos. Entonces, el paciente tiene que completar 32 ensayos enfocándose en la tarea 2D o 3D mediante movimientos globales, tanto de hombro como de codo. Aproximadamente, el bloque se realiza en 4-5 minutos.

- Una vez que se han completado los 32 ensayos, el paciente tiene un tiempo de descanso de tres minutos.

- Para finalizar la sesión, se realiza el segundo bloque de movimientos completando 32 ensayos de la otra tarea. Al igual que con el primer bloque, el paciente tiene que realizar los mismos movimientos globales y el tiempo necesario para realizar estos ensayos también es aproximadamente de 4-5 minutos.

En la Figura 5.4 se muestra esquemáticamente el flujo de trabajo para realizar un ensayo. Básicamente, los usuarios tienen que alcanzar uno de los ocho posibles objetivos y volver hacia el objetivo central. Al iniciar el ensayo, los sujetos tienen que mantener el elemento controlable durante dos segundos encima del objetivo central. Entonces, uno de los objetivos alcanzables se iluminaba para indicar la siguiente posición donde el paciente tiene que colocar el efector final. El sistema da un tiempo limitado de tres segundos para completar la trayectoria del movimiento, y cuando el sujeto alcanza el objetivo, tiene que volver al objetivo central sin límite de tiempo. Por lo tanto, esta secuencia de movimientos se lleva a cabo en los 32 ensayos planteados, señalando objetivos de manera aleatoria. Este protocolo de pasos se ha utilizado en las dos tareas virtuales.

La caracterización de la función sensorimotor se ha realizado a través del cálculo de siete parámetros para cada movimiento. Estos se basan en los parámetros utilizados por (Coderre et al., 2010) y son los siguientes:

1. **Velocidad máxima**: Es la máxima velocidad alcanzada por el movimiento de la mano.

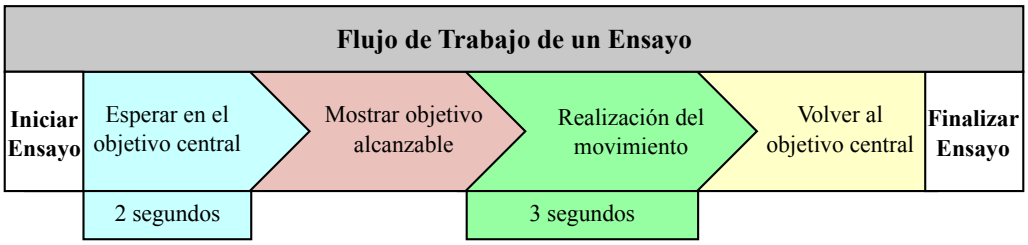


Figura 5.4: Flujo de trabajo completo de un ensayo.

2. **Tiempo de reacción**: Es el tiempo que transcurre desde la señalización del objetivo aleatorio hasta el inicio del movimiento del brazo.

3. **Distancia total**: La distancia total alcanzada para completar un objetivo

4. **Distancia inicial**: La distancia recorrida durante el movimiento inicial cuya trayectoria tiene una desviación con respecto al objetivo alcanzable hasta que se corrige esta desviación.

5. **Razón del movimiento inicial**: Relación entre la distancia inicial y la longitud de la trayectoria.

6. **Error en la dirección del movimiento inicial**: Es la desviación angular entre la trayectoria óptima, establecida por la línea recta desde el objetivo central a los objetivos alcanzables, y el vector desde el punto central hasta la posición de la mano generado por el movimiento inicial

7. **Tiempo**: El tiempo total necesitado para alcanzar un objetivo.

8. **Tasa de éxito**: El porcentaje de ensayos completados con éxito.

Adicionalmente se ha definido un estudio de la usabilidad global del sistema de neuro-rehabilitación con el fin de obtener un indicador cuantitativo de la usabilidad que mide el nivel de cumplimiento frente a las expectativas de los usuarios, el nivel de satisfacción y el rendimiento del sistema. Para ello se ha utilizado una encuesta de tipo Escala de la Usabilidad del Sistema (SUS) (Brooke et al., 1996), la cual se ha entregado a cada uno de los pacientes después de probar el sistema y contiene diez preguntas que dan una visión global de las evaluaciones subjetivas de la usabilidad. Seis

de estas preguntas tienen un carácter positivo y las preguntas restantes son negativas para contrastar las respuestas. Los usuarios tenían que responder en función de su nivel de acuerdo o de desacuerdo con respecto al sistema utilizando una escala *Likert* (de 1 a 5) (Likert, 1967).

5.3. Resultados

En este estudio se han adquirido datos objetivos durante 36 sesiones de tratamiento de terapia virtual asistida por el robot PUPArm y la Tabla 5.2 recoge las principales estadísticas descriptivas de cada paciente. Los resultados se presentan en una tabla de diez columnas para mostrar los valores de los parámetros cinemáticos que evalúan la calidad de la función motora del paciente. Cada parámetro contiene dos valores posibles dependiendo del modo de visualización de la tarea proporcionada al paciente. En este caso, el primer valor se extrae de la Ruleta 2D y el otro de la Ruleta 3D para poder comparar los resultados numéricos. En la última columna se muestra la tasa de éxito a la hora de completar los objetivos alcanzables. La alta tasa de éxito entre el 95,10 % y el 100 % indica que el sistema es fácil de utilizar sin complicaciones durante la realización de las tareas virtuales. A simple vista no existen diferencias significativas en la tasa de éxito entre los dos tipos de visualización sin embargo se ha realizado un análisis estadístico para obtener relaciones entre parámetros.

Paciente	Tarea	Velocidad Máxima(mm/s)	Tiempo de Reacción(s)	Distancia Total(mm)	Distancia Inicial(mm)	Razón de Mov. Inicial	Error Inicial Ángulo(º)	Tiempo	Tasa de Éxito(%)
1	2D	104.13	0.69	104.27	72.45	0.70	1.11	7.11	99.61
	3D	115.56	0.89	114.41	83.01	0.74	1.13	8.18	99.61
2	2D	57.07	0.65	114.55	41.60	0.38	3.21	10.35	100
	3D	58.64	0.71	121.53	42.01	0.36	3.17	11.38	98.64
3	2D	91.81	0.85	116.94	58.38	0.52	2.05	11.81	100
	3D	92.58	1.09	119.06	60.60	0.53	1.87	13.59	100
4	2D	118.30	0.71	123.82	69.13	0.61	1.57	13.45	99.67
	3D	134.66	0.89	149.68	78.62	0.60	1.67	16.26	98.58
5	2D	153.19	0.88	197.90	95.59	0.61	1.71	15.41	98.83
	3D	153.40	1.04	250.27	97.45	0.51	2.53	27.04	95.10
6	2D	45.94	0.40	110.07	33.96	0.32	3.56	12.13	98.83
	3D	46.57	0.48	111.14	32.89	0.31	3.77	13.48	97.01
7	2D	63.24	0.64	120.42	42.98	0.37	3.19	16.07	97.13
	3D	60.49	0.77	121.61	41.68	0.36	3.17	15.99	95.53
8	2D	110.09	0.52	105.96	69.78	0.68	1.34	8.03	98.96
	3D	113.72	0.71	115.34	74.28	0.67	1.37	11.76	98.96
9	2D	112.37	0.73	130.40	75.48	0.64	1.73	12.36	100
	3D	121.37	1.05	157.78	83.21	0.61	1.88	13.33	98.27

Tabla 5.2: Datos adquiridos por el dispositivo robótico.

Con los valores extraídos de las tareas se ha realizado un análisis comparativo con el objetivo de calcular la variación en el rendimiento del paciente de los parámetros 3D con respecto a los parámetros 2D. Se ha utilizado la Ecuación 5.1 para calcular los porcentajes de esta variación. Esta ecuación se aplica a cada uno de los parámetros para todos los usuarios, cuyas variables $dato3D$ y $dato2D$ generalizan el valor numérico de estos parámetros cinemáticos. Como cada parámetro tiene asociados dos posibles valores, la variable $dato3D$ registra el valor de la tarea 3D y $dato2D$ contiene el valor de la tarea 2D. Los valores de variación se reúnen en la Tabla 5.3 incluyendo la media, la desviación estándar, la mediana y el valor máximo-mínimo de cada parámetro para todos los datos. Unos valores positivos indican el incremento de porcentaje, donde el parámetro extraído de la tarea en 3D es mayor que el mismo parámetro en la tarea 2D.

$$\left(\frac{dato3D - dato2D}{dato2D}\right) * 100 \tag{5.1}$$

Paciente	Velocidad Máxima	Tiempo de Reacción	Distancia Total	Distancia Inicial	Razón de Mov. Inicial	Error Inicial de Ángulo	Tiempo	Tasa de Éxito
1	10.97	29.27	9.73	14.57	5.79	1.55	15.13	0
2	2.75	9.35	6.09	0.99	-4.69	-2.14	10	-1.36
3	0.83	27.80	1.80	3.80	2.14	-8.99	15.12	0
4	13.83	25.26	20.89	13.72	-3.07	6.74	20.89	-1.09
5	0.13	18.92	26.46	1.94	-17.28	47.90	75.47	-3.78
6	1.36	20.11	0.97	-3.09	-3.79	5.99	11.12	-1.84
7	-4.35	10.99	0.98	-3.03	-3.06	-0.49	-0.47	-1.64
8	3.29	37.77	8.85	6.45	-1.16	2.18	46.58	0
9	8	42.93	20.99	10.24	-4.77	8.99	7.83	-1.73
MEDIA	4.09	24.71	10.75	5.07	-3.32	6.86	22.41	-1.27
STD	5.42	10.60	9.14	6.27	5.93	15.37	22.42	1.21
MEDIANA	2.75	25.26	8.85	3.80	-3.07	2.18	15.12	-1.36
MAX	13.83	42.93	26.46	14.57	5.79	47.90	75.47	0
MIN	-4.35	9.35	0.97	-3.09	-17.28	-8.99	-0.47	-3.78

Tabla 5.3: Variación de los parámetros 3D con respecto a los parámetros 2D [%].

Estos parámetros se representan como un diagrama de cajas en la Figura 5.5 para proporcionar una visión general de la distribución de los datos. En un diagrama de cajas, las cajas se dividen por un segmento horizontal que indica la posición del valor de la mediana. Por lo tanto, se puede observar la relación entre este valor y los percentiles 25th y 75th, representados por la parte inferior y la parte superior de la caja, con una distribución asimétrica. Las cajas se encuentran en un segmento cuyos extremos indican

los valores mínimo y máximo del parámetro. En este diagrama de cajas, los valores atípicos se han marcado a través del símbolo +.

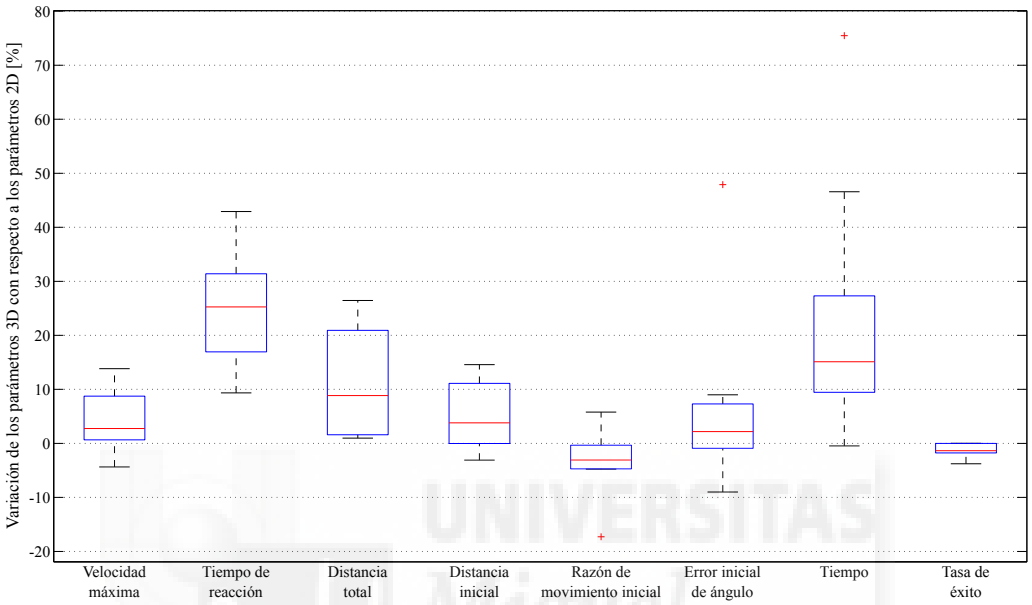


Figura 5.5: Análisis estadístico de los datos adquiridos, representado en un diagrama de cajas.

Para evaluar la función sensomotora después de la terapia de neurorehabilitación asistida por robots se han comparado las trayectorias de movimiento realizadas por los sujetos en la primera y en la última sesión. En la Figura 5.6 se muestran las trayectorias efectuadas por un usuario. El lado izquierdo corresponde a las trayectorias realizadas durante la visualización de la Ruleta 2D, mientras que las del lado derecho se llevaron a cabo con la Ruleta 3D. En ambas tareas, todos los pacientes efectuaron trayectorias más erráticas en la primera sesión cuando intentaban llegar a los objetivos alcanzables. Sin embargo las trayectorias realizadas con el entorno 2D eran mejores en términos de movimiento rectilíneo, mientras que las trayectorias generadas durante la tarea 3D presentaban unas trayectorias más irregulares. Por otra parte, los pacientes llevaron a cabo menos desviaciones de trayectoria cuando tenían que alcanzar el objetivo central en la tarea 2D. Con respecto a la primera sesión, las trayectorias entre el objetivo alcanzable y el objetivo central presentan una mayor longitud en la tarea 3D que en la tarea 2D, y el error en la desviación y el tiempo para alcanzar

los objetivos es mayor (véase la Tabla 5.3 y Figura 5.6). Por lo tanto, los pacientes presentan más dificultades a la hora de alcanzar los objetivos en la tarea 3D al inicio de la terapia. Esto puede significar que el uso de tareas con gráficos en 2D y menor nivel de detalle, es más fácil de percibir y se adapta mejor a los usuarios que no han utilizado nunca este tipo de sistema. En la Figura 5.6 también se puede observar una mejora en el control del sistema por parte el paciente. Este hecho sugiere que el rendimiento sensomotor del paciente se incrementa debido a la repetición de los movimientos del brazo planteando que existe un aprendizaje asociado al uso continuo de estos sistemas.

Otro punto de este estudio ha consistido en la realización de un análisis estadístico basado en una correlación bivariada de los elementos cuantitativos de los parámetros cinéticos y cinemáticos para verificar la relación entre variables así como el nivel y la dirección de la correlación. De esta manera, se puede analizar qué tipo de visualización genera mejores relaciones entre parámetros cinemáticos, y por consiguiente, se puede lograr una mejor evaluación de los resultados. Este análisis se realiza calculando el coeficiente de correlación de *Pearson* y el nivel de significación para indicar si existe relación entre cada par de los parámetros del estudio. La Tabla 5.4 muestra la matriz de correlación con el coeficiente de correlación de *Pearson* y el nivel de significación de cada par de parámetros asignados al registro de los datos de la visualización en 2D. En cuanto a la tarea 3D, la matriz de correlación se registra en la Tabla 5.5.

		Velocidad Máxima	Tiempo de Reacción	Distancia Total	Distancia Inicial	Error Inicial de Ángulo	Tiempo
Velocidad Máxima	R	1	0.645	0.654	0.986**	-0.842**	0.075
	Sig.	-	0.060	0.056	0.000	0.004	0.848
Tiempo de Reacción	R	0.645	1	0.605	0.640	-0.457	0.317
	Sig.	0.060	-	0.084	0.063	0.217	0.406
Distancia Total	R	0.654	0.605	1	0.642	-0.158	0.614
	Sig.	0.056	0.084	-	0.062	0.685	0.078
Distancia Inicial	R	0.986**	0.640	0.642	1	-0.852**	0.003
	Sig.	0.000	0.063	0.062	-	0.004	0.994
Error Inicial de Ángulo	R	-0.842**	-0.457	-0.158	-0.852**	1	0.386
	Sig.	0.004	0.217	0.685	0.004	-	0.305
Tiempo	R	0.075	0.317	0.614	0.003	0.386	1
	Sig.	0.848	0.406	0.078	0.994	0.305	-

Tabla 5.4: Matriz de correlación para los datos obtenidos en la visualización 2D.

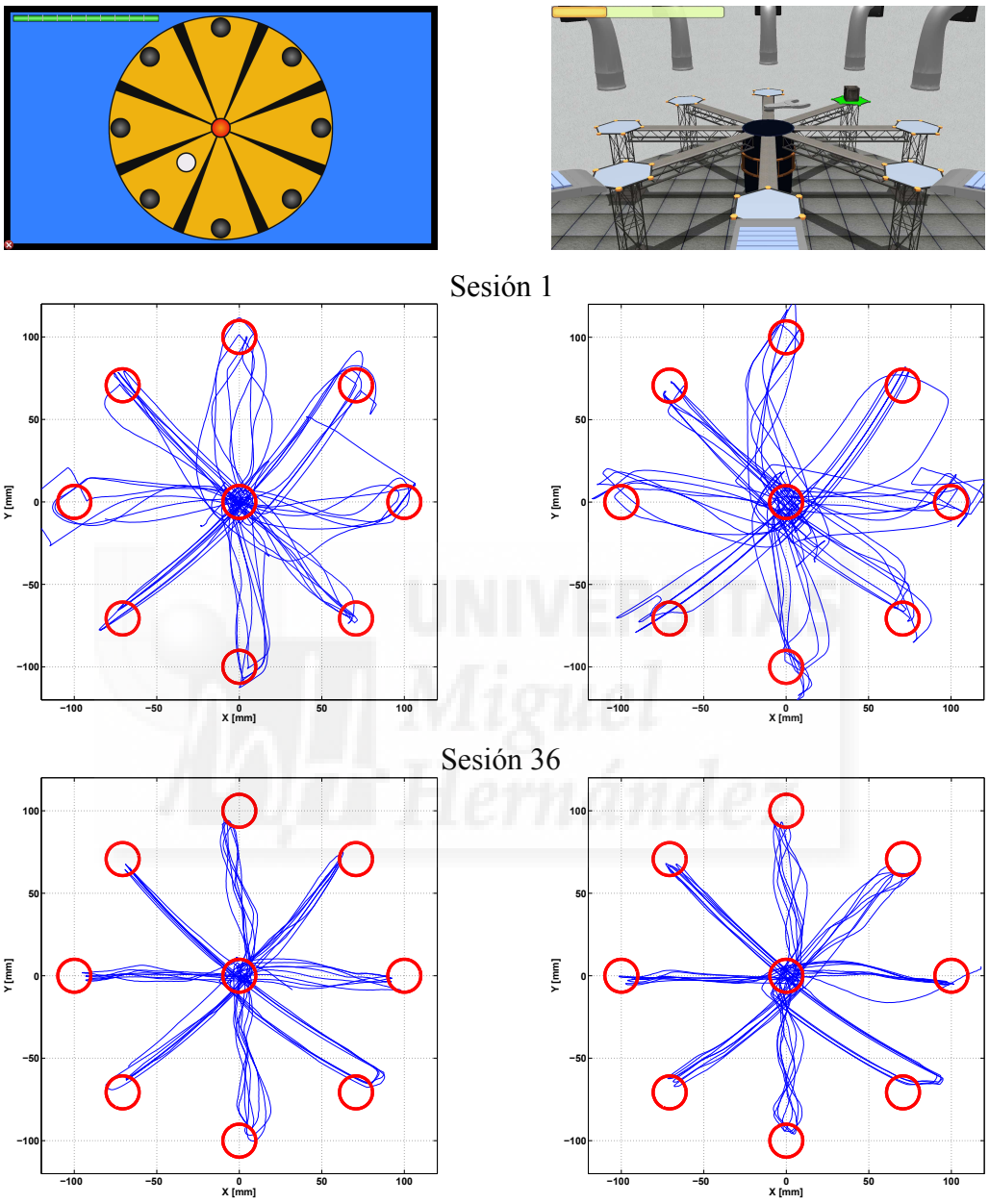


Figura 5.6: Trayectorias realizadas por un usuario para alcanzar los objetivos. Valoración de la función sesomotora en las dos tareas: En la izquierda se muestran las trayectorias realizadas en la tarea 2D durante la primera y la última sesión. En el lado derecho se muestran las trayectorias para la tarea 3D.

		Velocidad Máxima	Tiempo de Reacción	Distancia Total	Distancia Inicial	Error Inicial de Ángulo	Tiempo
Velocidad Máxima	R	1	0.729*	.0.684*	0.982**	-0.713*	0.445
	Sig.	-	0.026	0.042	0.000	0.031	0.230
Tiempo de Reacción	R	0.729*	1	0.530	0.745*	-0.594	0.321
	Sig.	0.026	-	0.142	0.021	0.092	0.400
Distancia Total	R	0.684*	0.530	1	0.649	-0.002	0.899**
	Sig.	0.042	0.142	-	0.059	0.996	0.001
Distancia Inicial	R	0.982**	0.745*	0.649	1	-0.746*	0.360
	Sig.	0.000	0.021	0.059	-	0.021	0.342
Error Inicial de Ángulo	R	-0.713*	-0.594	-0.002	-0.746*	1	0.253
	Sig.	0.031	0.092	0.996	0.021	-	0.512
Tiempo	R	0.445	0.321	0.899**	0.360	0.253	1
	Sig.	0.230	0.400	0.001	0.342	0.512	-

Tabla 5.5: Matriz de correlación para los datos obtenidos en la visualización 3D.

En ambas tablas el coeficiente de correlación viene determinado por el parámetro R, mientras que el nivel de significación está indicado con Sig. Además, se ha marcado las correlaciones significativas en un nivel de 0.05 a través de $*$ y para una correlación de nivel 0.01 con $**$.

Al finalizar la terapia, se le proporcionó una encuesta SUS a cada uno de los pacientes para calcular una medida de usabilidad del sistema de neurorehabilitación. La Tabla 5.6 muestra las cuestiones que cubren una variedad de aspectos de la usabilidad del sistema, tales como la necesidad de apoyo, el entrenamiento y la complejidad.

	Nº Paciente								
Nº Pregunta	1	2	3	4	5	6	7	8	9
1 Creo que me gustaría usar este sistema con frecuencia	5	5	5	4	5	5	5	5	5
2 Encontré el sistema innecesariamente complejo	5	5	1	2	1	3	3	1	1
3 Pensé que el sistema era fácil de usar	5	5	5	4	4	3	3	5	5
4 Creo que voy a necesitar el apoyo técnico para utilizar el sistema	3	3	2	2	1	5	4	3	4
5 Me parece que las funciones del sistema se integran bien	4	4	5	4	5	3	3	4	4
6 Pienso que no hay incongruencias en el sistema	5	5	1	2	5	3	5	5	1
7 Imagino que la gente aprende a utilizar este sistema rápidamente	4	4	5	4	5	3	4	3	2
8 Encontré el sistema complicado de usar	1	1	1	1	1	3	4	1	1
9 Me sentí seguro usando el sistema	5	5	5	4	5	3	3	5	5
10 Tenía que aprender muchas cosas antes de poder utilizar sistema	1	1	1	1	1	3	4	1	1

Tabla 5.6: Preguntas de la encuesta y las respuestas de los pacientes.

Esta tabla con las cuestiones también reúne los resultados proporcionados por cada usuario en una escala de 5 puntos que van desde “muy de

acuerdo" a "muy en desacuerdo". La Figura 5.7 agrupa el promedio de la evaluación de todos los pacientes sobre los aspectos tratados en la encuesta. En este gráfico radial, la línea punteada representa la respuesta óptima para conseguir el nivel subjetivo más alto de usabilidad, mientras que la línea continua es la respuesta promedio de los nueve pacientes. En general, las respuestas de los pacientes fueron ampliamente similares a las respuestas óptimas, a excepción de la pregunta acerca de la necesidad de apoyo técnico.

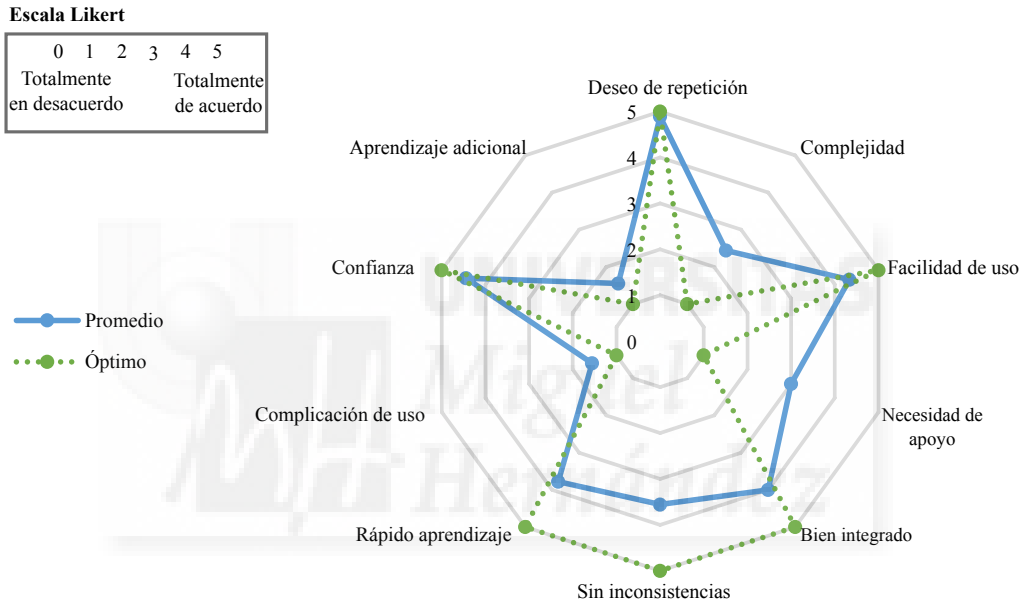


Figura 5.7: Promedio de las respuestas de los pacientes en la encuesta.

Con el objetivo de conseguir una interpretación de la usabilidad representada por un indicador de porcentaje, se añade una contribución numérica a cada punto con un rango de 0 a 4 dependiendo de la pregunta (Figura 5.8). Las respuestas subjetivas de los pacientes que corresponden con la respuestas óptimas obtendrán un valor numérico de 4. Entonces, se suman todas las contribuciones y el resultado se multiplica por 2,5 para obtener la puntuación SUS (Brooke et al., 1996). La puntuación SUS total de la encuesta fue un 76,11 %, el cual es un indicador muy positivo que refleja un alto grado de satisfacción y compromiso de los pacientes evaluados, ya que los porcentajes más altos significan un mejor nivel de usabilidad.

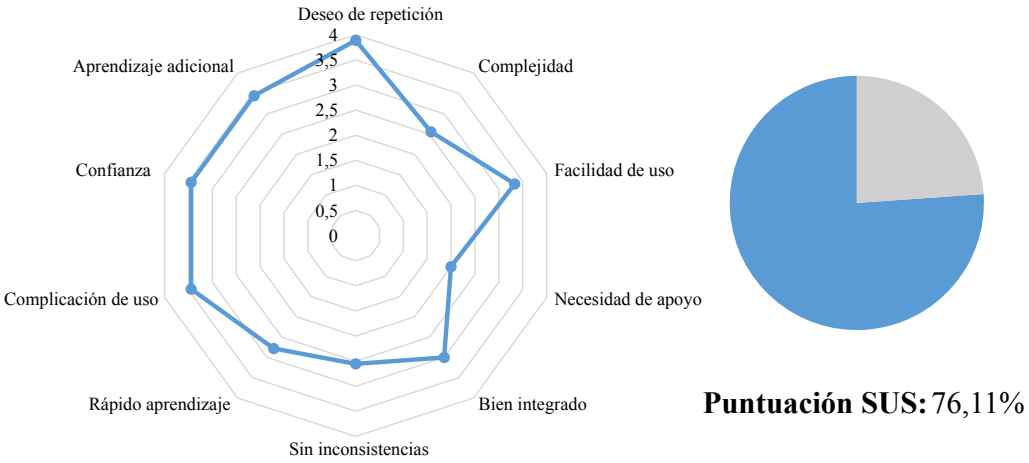


Figura 5.8: Contribuciones numéricas de los aspectos de la encuesta y la puntuación SUS.

5.4. Discusiones

En esta sección se realiza una discusión de los resultados y del análisis estadístico de correlaciones bivariadas sobre la influencia de la aplicación de juegos terapéuticos 2D o 3D en la rehabilitación del miembro superior en pacientes con daño cerebral. En la actualidad, este tipo de estudios no se han abordado todavía en la literatura científica o los temas tratados en este campo son discutidos por evaluaciones subjetivas. Por estas razones, se ha evaluado de forma objetiva una correlación cuantitativa entre la función motora del miembro superior y la visualización de una tarea de alcance a través del cálculo de parámetros cinemáticos proporcionadas por el dispositivo robótico PUPArm. A primera vista, el análisis de los parámetros cinemáticos con los dos tipos de visualización han proporcionado resultados muy similares comparando valores nominales, sin embargo se han encontrado algunas pequeñas diferencias en el rendimiento sensorimotriz dependiendo de la visualización de la tarea. Cada paciente ha logrado resultados similares cuando realizó las tareas con ambos entornos durante todas las sesiones de la terapia. Sin embargo, algunos obtienen mejores resultados que otros, mostrando una variación en las capacidades sensoriomotrices. Estos cambios pueden deberse a la edad, a daños motores o al nivel de lesión cerebral que afecta a la eficiencia de los procesos cognitivos y fisiológicos.

El propósito más importante de las tareas era ofrecer una referencia visual del movimiento del brazo asistido con el dispositivo robótico para

completar diferentes objetivos realizando trayectorias rectilíneas. En las primeras sesiones, prácticamente todos los pacientes alcanzaban gran parte de los objetivos de ambas tareas con trayectorias erráticas y presentaban diferentes desviaciones aleatorias en cada uno de los ensayos. Sin embargo, se lograron mejores movimientos cinemáticos durante la tarea 2D llevando a cabo trayectorias más rectilíneas en los objetivos que necesitaban recorridos diagonales (Figura 5.6). En la tarea 3D se observan trayectorias más desviadas en casi todos los objetivos. Con respecto al alcance del objetivo central, se generaron trayectorias más precisas y con menos desviación en la tarea 2D.

En la Figura 5.9 se muestran los primeros diagramas significativos del estudio estadístico de correlaciones. El gráfico de la izquierda presenta el diagrama de dispersión entre la distancia inicial y el error en la dirección del movimiento inicial. Estos parámetros afectan directamente al rendimiento de las trayectorias y han tenido un nivel de significación bastante alto en el análisis de correlaciones en ambos tipos de visualización. Esta correlación fue negativa y el grado de asociación lineal es más fuerte en la tarea 2D. Esto significa que el error de desviación cuando el paciente empieza el movimiento, se corrige con menos distancia inicial para alcanzar la mejor dirección de la trayectoria hacia el objetivo con entornos 2D. Este hecho es gracias a que la vista cenital permite visualizar mejor el camino recorrido por la trayectoria y facilita la corrección de los errores iniciales de desviación.

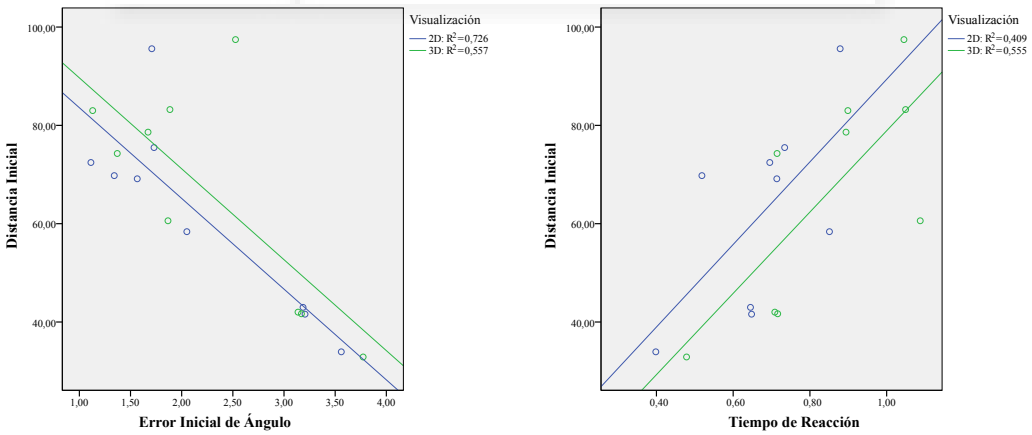


Figura 5.9: Diagrama de dispersión con las variables más significativas que afectan a la desviación inicial de las trayectorias.

Estos datos sugieren que las trayectorias en las tareas 2D tienen un me-

jor rendimiento y que los entornos 2D son escenarios más apropiados para generar tareas que proporcionan un mejor control sensoriomotriz cuando los pacientes comienzan a utilizar el sistema. En la última sesión, las trayectorias de movimiento se corrigieron significativamente hasta el punto de lograr caminos casi sin desviación, mientras que los objetivos se alcanzan satisfactoriamente con trayectorias estables. Comparando los resultados entre la primera y la última sesión se puede observar que las trayectorias mejoran con la experiencia del paciente. Esta mejora indica una valoración positiva en la recuperación de la función sensoriomotriz, de igual manera que ocurrió en (LLinares et al., 2013; Badesa et al., 2014b). En general, las tasas de éxito en todas las sesiones fueron bastante altas, dando como conclusión que las tareas no eran complejas y los objetivos son reconocidos con claridad.

El tiempo de reacción de todos los pacientes en la tarea 3D fue superior que en la tarea 2D, implicando que el aumento del nivel de inmersión en el entorno provoca distracciones innecesarias a los pacientes. Así, el nivel de concentración del paciente aumenta con menos nivel de detalle en el entorno virtual. Sin embargo, la correlación entre la distancia inicial y el tiempo de reacción (gráfico de la derecha en la Figura 5.9) es más significativa en la tarea 3D. Entonces, se puede decir que el tiempo de reacción en la tarea 2D implica un mayor error de desviación inicial junto con la necesidad de realizar una distancia de corrección mayor para compensar dicho error, a pesar de que el valor nominal del tiempo de reacción en la tarea 2D es más pequeño que en la tarea 3D.

Durante la realización de las trayectorias, los pacientes desplazaban el manipulador del robot en menor medida en la tarea 2D, entonces la distancia total recorrida ha sido mayor en la tarea 3D. En consecuencia, el tiempo total para completar todos los ensayos fue superior en la tarea 3D. Sólo un paciente ha necesitado menos tiempo para realizar la tarea 3D. Este hecho puede indicar que los pacientes guían mejor el manipulador robótico cuando están observando un entorno 2D. La profundidad del escenario en la tarea 3D aumenta el nivel de dificultad para completar los objetivos, provocando que los pacientes se adapten mejor a la tarea 2D. Aunque la distancia total fue mayor en la tarea 3D, los tiempos para completar ambas tareas no difieren sustancialmente debido a que los pacientes lograban una velocidad mayor de movimiento en la tarea 3D. La correlación entre la distancia total y el tiempo para alcanzar un objetivo sólo ha tenido un nivel de significación relevante con un valor de relación muy fuerte en la tarea 3D (Figura 5.10).

Todos los ensayos se pueden realizar de una manera óptima siguiendo

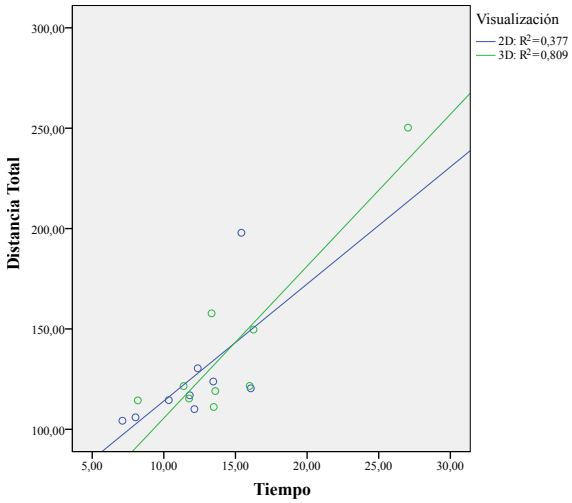


Figura 5.10: Diagrama de dispersión entre los parámetros de distancia total y tiempo.

una trayectoria recta desde el objetivo central y el objetivo alcanzable, y viceversa. Sin embargo, los pacientes tenían una desviación cuando comenzaban el movimiento en casi todos los ensayos. Esta desviación implica que los pacientes realizan una ruta incorrecta antes de que la dirección de trayectoria se corrigiera para alcanzar la ruta óptima. Esta situación se acentúa más en la tarea 3D como se observa a través del análisis de los valores nominales de la distancia inicial y el error inicial del ángulo en la dirección del movimiento. En cuanto a la velocidad máxima, existen fuertes coeficientes de correlación entre éste y los demás parámetros. La Figura 5.11 reúne cuatro diagramas de dispersión donde la velocidad máxima se compara con los parámetros que proporcionan una correlación mayor.

Los dos diagramas situados en la parte izquierda indican que la velocidad máxima tiene una mejor correlación con la distancia inicial y el error inicial del ángulo en la dirección del movimiento en la tarea 2D. En general, se puede decir que la tarea 2D proporciona un mejor control de la velocidad del efector final teniendo en cuenta el comienzo de los movimientos. La correlación entre la velocidad máxima y el tiempo de reacción o la distancia total fue mejor en la tarea 3D. Estas dos correlaciones pueden sugerir que la tarea 3D permite al usuario realizar trayectorias más largas con movimientos más naturales y dinámicos. Entre el resto de los parámetros no se han encontrado correlaciones significativas.

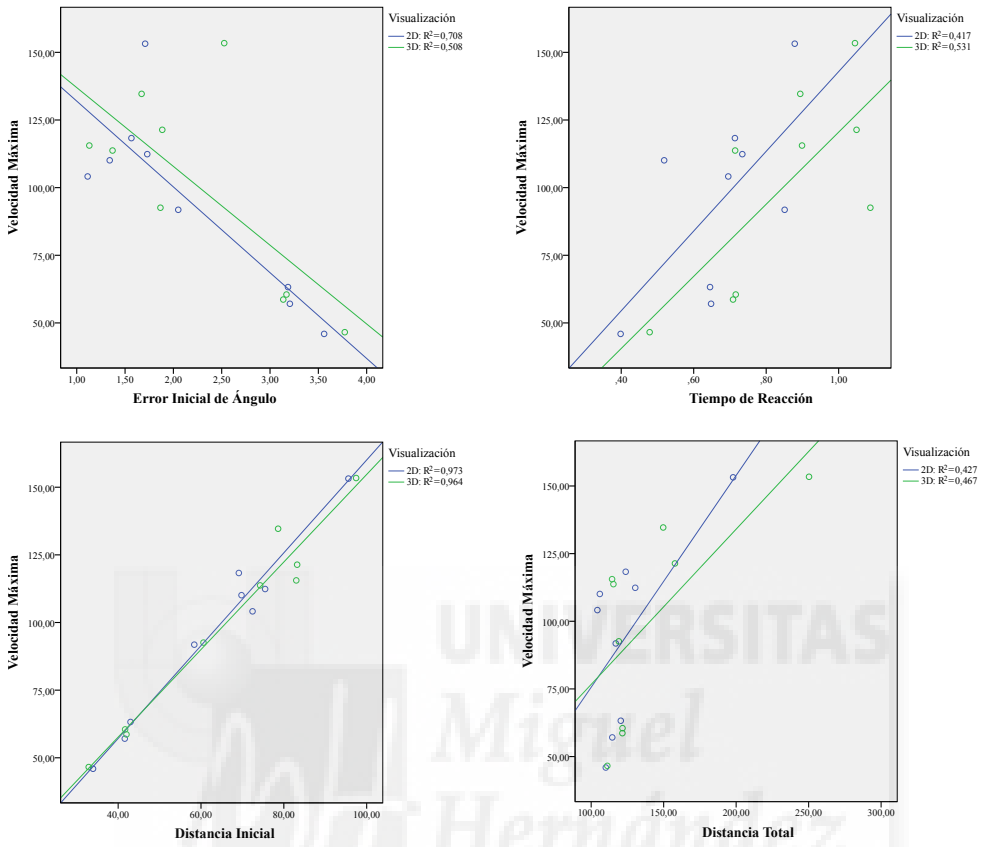


Figura 5.11: Diagramas de dispersión con las correlaciones más significativas de la velocidad máxima.

5.5. Conclusiones

El objetivo principal de este estudio era verificar si habían diferencias en los patrones de los movimientos cinemáticos de pacientes con daño cerebral asistidos por un dispositivo robótico para situaciones donde se visualizaban entornos en dos y tres dimensiones. A pesar de la similitud en los resultados y del análisis de correlaciones entre parámetros, la hipótesis que consiste en mostrar un entorno de visualización más natural aumentando el nivel de inmersión no ha proporcionado muchas mejoras con respecto a un entorno más simple. Por lo tanto, el uso de entornos 2D en la terapia virtual puede ser una manera más adecuada y cómoda para llevar a cabo tareas de rehabilitación del miembro superior en pacientes con daño cerebral, en

términos de precisión de movimiento con el objetivo de efectuar trayectorias cinemáticas óptimas sobretodo al principio de la terapia. Sin embargo, la visualización 3D permite realizar movimientos más naturales debido a la mejor asociación entre el movimiento del efector final y el movimiento del avatar dentro de la tarea virtual.

Tener conocimiento sobre qué entorno virtual es más apropiado para cada usuario permite implementar terapias con mejores instrumentos de evaluación que pueden ser adaptadas a las necesidades y a las limitaciones del paciente (Morales et al., 2014). Entonces, dependiendo del objetivo de evaluación acerca de la función sensoriomotriz como el tiempo de reacción, la velocidad o la estabilidad de movimiento, se puede utilizar un tipo de visualización u otro. Esto proporciona muchos beneficios en el entorno clínico para mejorar el proceso de rehabilitación de la función sensoriomotriz y reducir los tiempos de recuperación.

Capítulo 6

Conclusiones

UNIVERSITAS
Miguel
Hernández

6.1. Conclusiones

La mayor parte de las terapias virtuales asistidas por dispositivos robóticos están basadas en simulaciones de AVD. Cada tarea implica la creación de un nuevo fichero de ejecución que reproduzca el entorno virtual en un dispositivo de visualización. Esta rutina no es la forma más conveniente para desarrollar una biblioteca extensa de múltiples tareas. Una variedad en los ejercicios mantiene la motivación de lo pacientes que realizan este tipo de terapias al evitar la repetición constante de las mismas tareas durante todo el proceso de rehabilitación. Además, los entornos de visualización realistas en 3D y el comportamiento físico de sus elementos buscan el objetivo de conseguir una mayor inmersión, atención y participación por parte del paciente. Por otro lado, existe una gran cantidad de herramientas que ayudan a este proceso de generación de juegos virtuales, sin embargo supone un inconveniente el precio que pueden llegar a tener estas licencias o las limitaciones del software privativo.

Teniendo en cuenta todos estos aspectos, se ha implementado un sistema software con herramientas de libre acceso que permite la generación automática de este tipo de tareas cargando unos *scripts* de texto plano con la organización de la escena y los objetivos a cumplir, junto con los ficheros de recursos para la representación visual del escenario y la parte auditiva. La ventaja que aporta este sistema es la implementación de una herramienta software para obtener de manera óptima, rápida y con bajo coste cualquier tipo de tarea virtual con cualquier tipo de entorno, siempre ajustándose al tipo de pacientes a quiénes van dirigidas y al tipo de dispositivo de interacción que se utiliza, ya sea una simulación de AVD u otro tipo de ejercicios terapéuticos. Además, las tareas generadas con este sistema tienen la posibilidad de adaptar el nivel de dificultad dependiendo de las condiciones externas definidas por el estado emocional del paciente a través de sus señales fisiológicas.

Una vez implementado el sistema que permite obtener juegos terapéuticos se ha definido una interfaz auto-adaptativa de interacción que permita incluir al paciente en el lazo de control de la terapia. Esta interfaz está formada por un sistema de adquisición multimodal de señales fisiológicas que permite incorporar datos del usuario dentro del lazo de control, por un sis-

tema robótico para completar las trayectorias definidas por las tareas de rehabilitación y un ordenador encargado de controlar la coordinación entre todos los elementos. En esta tesis se ha probado un nuevo algoritmo de clasificación basado en lógica difusa y redes neuronales para la estimación del estado psico-fisiológico del usuario a partir de los vectores característicos obtenidos del procesado de las señales fisiológicas. Este algoritmo es la primera vez que se utiliza para la clasificación de este tipo de señales, aportando mejores resultados que las distintas técnicas de aprendizaje automático utilizadas en otros casos de estudio. Además, gracias al módulo de comunicación UDP implementado, las tareas son capaces de recibir esta estimación para modificar el nivel de dificultad y adaptarse a las necesidades específicas de cada usuario.

Para finalizar, se ha realizado un estudio para determinar como influye la visualización de la RV en la realización de trayectorias asistidas por dispositivos roboticos de rehabilitación, comparando los resultados con nueve pacientes con DCA al realizar tareas de alcance mostradas con gráficos en 2D y 3D. Ambas tareas compartían el mismo diseño pero con distinta visualización de elementos y los objetivos eran los mismos para poder realizar una comparación objetiva. Los nueve pacientes realizaron una terapia de neuro-rehabilitación durante dos meses con cuatro sesiones semanales. Como primer resultado se ha demostrado la eficacia de este tipo de terapia para la recuperación motora debido a que los paciente alcanzaron un rango mayor de movimiento en la última sesión que en la primera sin importar el tipo de visualización. Sin embargo, los gráficos en 2D proporcionan una realización de trayectorias más optimizadas para alcanzar los objetivos sin mucha desviación sobre todo al comenzar a utilizar el sistema. Los gráficos en 3D ayudan a realizar trayectorias que requieren de profundidad en la pantalla a través de movimientos más naturales gracias a la asociación directa entre el efector final del dispositivo robótico y el avatar situado dentro de la escena virtual.

6.2. Trabajos Futuros

Durante la elaboración de esta tesis doctoral han quedado algunas líneas de trabajo abiertas para continuar con el estudio de desarrollo de este tipo de sistemas. Estas líneas se pueden resumir en los siguiente puntos:

- Unificar todo el proceso de creación de tareas en un mismo software

para evitar la utilización de programas externos complementarios. A través de la implementación de un sistema con GUI se puede crear un herramienta para modelar los entornos, aplicar comportamientos físicos, asignar objetivos a elementos o decidir cuando se ejecutan los sonidos todo de manera visual y generando los ficheros con su formato apropiado. De esta forma se podría distribuir un paquete software con todo lo necesario para desarrollar ejecutables con tareas virtuales de manera cómoda y sencilla con una licencia de código abierto.

- Establecer comunicación con otro tipo de dispositivos de interacción como cámaras de seguimiento visual, cámaras de profundidad, dispositivos hápticos, gafas virtuales, tablas de equilibrio y muchos más para realizar terapias virtuales de otros tipos de lesiones, como pueden ser las afectan a la vista o al equilibrio.

- Llevar a cabo estudios del sistema de interacción auto-adaptativo implementado en el Capítulo 4 con pacientes con daño cerebral para verificar el funcionamiento en tiempo real de los cambios de dificultad de los ejercicios terapéuticos dentro de una terapia de rehabilitación robótica.

- Comprobar el comportamiento de los usuarios ante dispositivos de inmersión virtual complementarios con los dispositivos de control robóticos para apoyar la terapia con nuevos métodos de asistencia.

- Realizar un estudio más completo con mayor número de pacientes con daño cerebral para determinar si el comportamiento de los movimientos cinemáticos están condicionados por la parte afectada del cerebro y compararlos con los resultados obtenidos con la experimentación del capitulo 5. Además, completar el estudio con un análisis de covarianzas más exhaustivo entre variables cuantitativas y cualitativas.

- Analizar mediante sistemas de electroencefalografía (EEG) la correspondencia entre el estado cerebral del usuario con las señales fisiológicas, constatando la actividad bio-eléctrica cerebral generada al inducir al usuario a los estados de relax, excitación media y estrés. Además, dicha actividad cerebral podría utilizarse como nuevo parámetro característico para estimar el estado psico-fisiológico del usuario.

UNIVERSITAS
Miguel
Hernández

Acrónimos

2D	Dos Dimensiones
3D	Tres Dimensiones
ACV	Accidente Cerebro Vascular
API	Application Programming Interface
ART	Teoría de Resonancia Adaptativa
AVD	Actividades de la Vida Diaria
CEGUI	Crazy Eddie's GUI
CPU	Unidad Central de Procesamiento
CRAA	Cognitive Regulated Affective Architecture
DCA	Daño Cerebral Adquirido
GDL	Grados De Libertad
GLSL	OpenGL Shading Language
GPU	Unidad de Procesamiento Gráfico
GSR	Galvanic Skin Response
GUI	Interfaz Gráfica de Usuario
HLSL	High Level Shader Language
HMD	Gafas de Realidad Virtual

IDE Entorno de Desarrollo Integrado

LGPL Lesser GNU Public License

LOOCV Leave-one-Out

LTM Long Term Memory

MI Índice Motor

OGRE3D Oriented Graphics Rendering Engine 3D

PC Componentes Principales

PCA Analisis de Componentes Principales

PPG Fotopletismografía

RBF Función de Base Radial

RV Realidad Virtual

SAM Self-Assessment Manikin

SCL Nivel de Conductancia de la Piel

SCR Respuesta de Conductancia de la Piel

SDK Kit de Desarrollo de Software

SNC Sistema Nervioso Central

SPS Sistemas de Proyección en Pantalla Grande

STM Short Term Memory

SUS Escala de la Usabilidad del Sistema

SVM Máquinas de Soporte Vectorial

TCE Traumatismo Cráneo Encefálico

UDP User Datagram Protocol

XML Lenguaje de Marcas eXtensible

Bibliografía

Akenine-Möller, T., Haines, E., and Hoffman, N. (2008). *Real-time rendering*. CRC Press.

Alexander, A. L., Brunyé, T., Sidman, J., and Weil, S. A. (2005). From gaming to training: A review of studies on fidelity, immersion, presence, and buy-in and their effects on transfer in pc-based simulations and games. *DARWARS Training Impact Group*, 5:1–14.

Allen, J. (2007). Photoplethysmography and its application in clinical physiological measurement. *Physiological measurement*, 28(3):R1.

Ambrona, A. C. (1999). Las ong's. federación española de daño cerebral (fedace). *Minusval*, (121):67–67.

Amirabdollahian, F., Loureiro, R., Gradwell, E., Collin, C., Harwin, W., and Johnson, G. (2007). Multivariate analysis of the fugl-meyer outcome measures assessing the effectiveness of gentle/s robot-mediated stroke therapy. *Journal of NeuroEngineering and Rehabilitation*, 4(1):1.

Anders, S., Lotze, M., Erb, M., Grodd, W., and Birbaumer, N. (2004). Brain activity underlying emotional valence and arousal: A response-related fmri study. *Human brain mapping*, 23(4):200–209.

Anderson, A. K., Christoff, K., Stappen, I., Panitz, D., Ghahremani, D., Glover, G., Gabrieli, J., and Sobel, N. (2003). Dissociated neural representations of intensity and valence in human olfaction. *Nature neuroscience*, 6(2):196–202.

ARTMAP, F. (1992). A neural network architecture for incremental supervised learning of analog multidimensional maps carpenter. *GA, Grossberg, S., Markuzon, N., Reynolds, JH, & Rosen, DB*, pages 698–713.

Badesa, F. J. (2014). *Interfaz multimodal y control biocooperativo para sistemas de neuro-rehabilitación asistida por robots*. phdthesis, Universidad Miguel Hernández de Elche.

Badesa, F. J., Llinares, A., Morales, R., Garcia-Aracil, N., Sabater, J. M., and Perez-Vidal, C. (2014a). Pneumatic planar rehabilitation robot for post-stroke patients. *Biomedical Engineering: Applications, Basis and Communications*, 26(02):1450025.

Badesa, F. J., Morales, R., Garcia-Aracil, N., Alfaro, A., Bernabeu, A., Fernandez, E., and Sabater, J. (2014b). Robot-assisted rehabilitation treatment of a 65-year old woman with alien hand syndrome. In *Biomedical Robotics and Biomechatronics (2014 5th IEEE RAS & EMBS International Conference on*, pages 398–402. IEEE.

Badesa, F. J., Morales, R., Garcia-Aracil, N., Sabater, J. M., Casals, A., and Zollo, L. (2014c). Auto-adaptive robot-aided therapy using machine learning techniques. *Computer methods and programs in biomedicine*, 116(2):123–130.

Badesa, F. J., Morales, R., Garcia-Aracil, N., Sabater, J. M., Perez-Vidal, C., and Fernandez, E. (2012). Multimodal interfaces to improve therapeutic outcomes in robot-assisted rehabilitation. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1152–1158.

Bertomeu-Motos, A., Lledó, L. D., Díez, J. A., Catalan, J. M., Ezquerro, S., Badesa, F. J., and Garcia-Aracil, N. (2015). Estimation of human arm joints using two wireless sensors in robotic rehabilitation tasks. *Sensors*, 15(12):30571–30583.

Blender (2016). Available online: `https://www.blender.org/` (accessed on 2 september 2016).

Blender2Ogre (2016). Available online: `https://bitbucket.org/iboshkov/blender2ogre/src/` (accessed on 2 september 2016).

Bohannon, R. W. and Smith, M. B. (1987). Interrater reliability of a modified ashworth scale of muscle spasticity. *Physical therapy*, 67(2):206–207.

Boian, R. F., Deutsch, J. E., Lee, C. S., Burdea, G. C., and Lewis, J. (2003). Haptic effects for virtual reality-based post-stroke rehabilitation. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*, pages 247–253. IEEE.

Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.

Burdea, G. (2002). Keynote address: Virtual rehabilitation-benefits and challenges. In *1st International Workshop on Virtual Reality Rehabilitation (Mental Health, Neurological, Physical, Vocational) VRMHR*, pages 1–11. sn.

Burdea, G. C. (1999). Haptic feedback for virtual reality.

Cameirão, M. S., i Badia, S. B., Oller, E. D., and Verschure, P. F. (2010). Neurorehabilitation using the virtual reality based rehabilitation gaming system: methodology, design, psychometrics, usability and validation. *Journal of neuroengineering and rehabilitation*, 7(1):1.

Cano-Izquierdo, J.-M., Almonacid, M., Pinzolas, M., and Ibarrola, J. (2009). dfasart: Dynamic neural processing in fasart model. *Neural Networks*, 22(4):479–487.

Cano-Izquierdo, J.-M., Ibarrola, J., and Almonacid, M. (2012). Improving motor imagery classification with a new bci design using neuro-fuzzy s-dfasart. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 20(1):2–7.

CEED (2016). Available online: `https://http://cegui.org.uk/wiki/CEED/` (accessed on 22 march 2016).

Coderre, A. M., Zeid, A. A., Dukelow, S. P., Demmer, M. J., Moore, K. D., Demers, M. J., Bretzke, H., Herter, T. M., Glasgow, J. I., Norman, K. E., et al. (2010). Assessment of upper-limb sensorimotor function of subacute stroke patients using visually guided reaching. *Neurorehabilitation and neural repair*, 24(6):528–541.

Colibazzi, T., Posner, J., Wang, Z., Gorman, D., Gerber, A., Yu, S., Zhu, H., Kangarlu, A., Duan, Y., Russell, J. A., et al. (2010). Neural systems subserving valence and arousal during the experience of induced emotions. *Emotion*, 10(3):377.

Collin, C. and Wade, D. (1990). Assessing motor impairment after stroke: a pilot reliability study. *Journal of Neurology, Neurosurgery & Psychiatry*, 53(7):576–579.

Crosbie, J., Lennon, S., Basford, J., and McDonough, S. (2007). Virtual reality in stroke rehabilitation: still more virtual than real. *Disability and rehabilitation*, 29(14):1139–1146.

Culmer, P. R., Jackson, A. E., Makower, S., Richardson, R., Cozens, J. A., Levesley, M. C., and Bhakta, B. B. (2010). A control strategy for upper limb

robotic rehabilitation with a dual robot system. *IEEE/ASME Transactions on Mechatronics*, 15(4):575–585.

da Silva Cameirão, M., Bermúdez i Badia, S., Duarte, E., and Verschure, P. F. (2011). Virtual reality based rehabilitation speeds up functional recovery of the upper extremities after stroke: a randomized controlled pilot study in the acute phase of stroke using the rehabilitation gaming system. *Restorative neurology and neuroscience*, 29(5):287–298.

Dario, P., Guglielmelli, E., Carrozza, M., Micera, S., Dipietro, L., and Pisano, F. (2003). Sistemi meccatronici e robotici per la neuro riabilitazione. *Bioingegneria della Postura e del Movimento*.

Deutsch, J. E., Latonio, J., Burdea, G. C., and Boian, R. (2001). Post-stroke rehabilitation with the rutgers ankle system: a case study. *Presence: Teleoperators and Virtual Environments*, 10(4):416–430.

Dickstein, R., Hocherman, S., Pillar, T., and Shaham, R. (1986). Stroke rehabilitation. *Physical Therapy*, 66(8):1233–1238.

Díez, J. A., Badesa, F. J., Lledó, L. D., Sabater, J. M., García-Aracil, N., Beltrán, I., and Bernabeu, Á. (2016). Design and development of a pneumatic robot for neurorehabilitation therapies. In *Robot 2015: Second Iberian Robotics Conference*, pages 315–326. Springer.

Doyon, J. and Benali, H. (2005). Reorganization and plasticity in the adult brain during learning of motor skills. *Current opinion in neurobiology*, 15(2):161–167.

Dukelow, S. P. (2011). Potential of robots as next-generation technology for clinical assessment of neurological disorders and upper-limb therapy. *Journal of rehabilitation research and development*, 48(4):335.

Einav, O., Geva, D., Yoeli, D., Kerzhner, M., and Mauritz, K.-H. (2011). Development and validation of the first robotic scale for the clinical assessment of upper extremity motor impairments in stroke patients. *Topics in stroke rehabilitation*, 18(Supplement-1):587–598.

Fasoli, S. E., Krebs, H. I., Stein, J., Frontera, W. R., and Hogan, N. (2003). Effects of robotic therapy on motor impairment and recovery in chronic stroke. *Archives of physical medicine and rehabilitation*, 84(4):477–482.

FEDACE (2015). Available online: `https://fedace.org/memorias.html` (accessed on 10 april 2017).

Fernandez, D. V., Angelina, C. M., et al. (2011). *Desarrollo de Videojuegos: Arquitectura del Motor de Videojuegos*. Cursos en Español.

Fernando, R. (2004). *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Pearson Higher Education.

Fernández, E., Ruiz, A., and Sánchez, A. (2009). Terapia ocupacional en daño cerebral adquirido. *TOG (A Coruña)*, 6(4):410–464.

Fiolhais, C. and Trindade, J. A. (1999). Virtual water, a virtual reality project for learning physics and chemistry. *Computer Physics Communications*, 121:639.

Fluet, G. G. and Deutsch, J. E. (2013). Virtual reality for sensorimotor rehabilitation post-stroke: the promise and current state of the field. *Current physical medicine and rehabilitation reports*, 1(1):9–20.

Fraile Marinero, J., Pérez Turiel, J., Rodríguez Guerrero, C., and Oliva, P. (2013). Evolución de la plataforma robotizada de neuro-rehabilitación physiobot. In *VII Congreso Iberoamericano de Tecnologías de Apoyo a la Discapacidad (Iberdiscap 2013), Santo Domingo (República Dominicana)*, pages 287–292.

Garcia, N., Sabater-Navarro, J. M., Gugliemeli, E., and Casals, A. (2011). Trends in rehabilitation robotics. *Medical and Biological Engineering and Computing*, 49(10):1089–1091.

García-Betances, R. I., Waldmeyer, M. T. A., Fico, G., and Cabrera-Umpiérrez, M. F. (2015). A succinct overview of virtual reality technology use in alzheimer's disease. *Frontiers in aging neuroscience*, 7.

Gerber, A. J., Posner, J., Gorman, D., Colibazzi, T., Yu, S., Wang, Z., Kangarlu, A., Zhu, H., Russell, J., and Peterson, B. S. (2008). An affective circumplex model of neural systems subserving valence, arousal, and cognitive overlay during the appraisal of emotional faces. *Neuropsychologia*, 46(8):2129–2139.

González, J. C., Pulido, J. C., Fernández, F., and Suárez-Mejías, C. (2015). Planning, execution and monitoring of physical rehabilitation therapies with a robotic architecture. In *Proceedings of the 26th Medical Informatics Europe conference (MIE). Studies in Health Technology and Informatics*, volume 210, pages 339–343.

Gregory, J. (2009). *Game engine architecture*. CRC Press.

Guerrero, C. R., Marinero, J. C. F., Turiel, J. P., and Muñoz, V. (2013). Using "human state awarerobots to enhance physical human–robot interaction in a cooperative scenario. *Computer methods and programs in biomedicine*, 112(2):250–259.

Henderson, A., Korner-Bitensky, N., and Levin, M. (2007). Virtual reality in stroke rehabilitation: a systematic review of its effectiveness for upper limb motor recovery. *Topics in stroke rehabilitation*, 14(2):52–61.

Hiebert, G. (2005). Openal 1.1 specification and reference.

Holden, M. K. (2005). Virtual environments for motor rehabilitation: review. *Cyberpsychology & behavior*, 8(3):187–211.

Initiative, O. S. (2015). Available online: `https://www.blender.org/` (accessed on 1 september 2015).

Itkowitz, B., Handley, J., and Zhu, W. (2005). The openhaptic toolkit: a library for adding 3d touchnavigation and haptics to graphics applications. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, pages 590–591. IEEE.

Izquierdo, J. M. C., Dimitriadis, Y. A., Sánchez, E. G., and Coronado, J. L. (2001). Learning from noisy information in fasart and fasback neuro-fuzzy systems. *Neural Networks*, 14(4):407–425.

Jack, D., Boian, R., Merians, A. S., Tremaine, M., Burdea, G. C., Adamovich, S. V., Recce, M., and Poizner, H. (2001). Virtual reality-enhanced stroke rehabilitation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 9(3):308–318.

Jackson, A., Culmer, P., Makower, S., Levesley, M., Richardson, R., Cozens, A., Williams, M. M., and Bhakta, B. (2007a). Initial patient testing of ipam-a robotic system for stroke rehabilitation. In *2007 IEEE 10th International Conference on Rehabilitation Robotics*, pages 250–256. IEEE.

Jackson, A. E., Holt, R. J., Culmer, P. R., Makower, S. G., Levesley, M. C., Richardson, R., Cozens, J. A., Williams, M. M., and Bhakta, B. B. (2007b). Dual robot system for upper limb rehabilitation after stroke: the design process. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 221(7):845–857.

Jang, S. H., You, S. H., Hallett, M., Cho, Y. W., Park, C.-M., Cho, S.-H., Lee, H.-Y., and Kim, T.-H. (2005). Cortical reorganization and associated functional motor recovery after virtual reality in patients with chronic stroke: an experimenter-blind preliminary study. *Archives of physical medicine and rehabilitation*, 86(11):2218–2223.

Jebara, N., Orriols, E., Zaoui, M., Berthoz, A., and Piolino, P. (2014). Effects of enactment in episodic memory: a pilot virtual reality study with young and elderly adults. *Frontiers in aging neuroscience*, 6:1–16.

Jezernik, S., Colombo, G., Keller, T., Frueh, H., and Morari, M. (2003). Robotic orthosis lokomat: A rehabilitation and research tool. *Neuromodulation: Technology at the neural interface*, 6(2):108–115.

Johnson, D. and Wiles, J. (2003). Effective affective user interface design in games. *Ergonomics*, 46(13-14):1332–1345.

Jørgensen, H., Nakayama, H., Raaschou, H., and Olsen, T. S. (1999). Stroke. neurologic and functional recovery the copenhagen stroke study. *Physical medicine and rehabilitation clinics of North America*, 10(4):887–906.

Kim, B. R., Chun, M. H., Kim, L. S., and Park, J. Y. (2011). Effect of virtual reality on cognition in stroke patients. *Annals of rehabilitation medicine*, 35(4):450–459.

Kim, G. J. et al. (2005). A swot analysis of the field of virtual reality rehabilitation and therapy. *Presence*, 14(2):119–146.

Kleim, J. A. and Jones, T. A. (2008). Principles of experience-dependent neural plasticity: implications for rehabilitation after brain damage. *Journal of speech, language, and hearing research*, 51(1):S225–S239.

Koenig, A., Omlin, X., Novak, D., and Riener, R. (2011). A review on bio-cooperative control in gait rehabilitation. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–6. IEEE.

Koenig, S. T., Crucian, G. P., Dalrymple-Alford, J. C., and Dünser, A. (2009). Virtual reality rehabilitation of spatial abilities after brain damage. *Stud Health Technol Inform*, 144:105–107.

Kovach, P. J. and Richter, J. (1999). *Inside Direct3d with Cdrom*. Microsoft Press.

Krebs, H. I., Ferraro, M., Buerger, S. P., Newbery, M. J., Makiyama, A., Sandmann, M., Lynch, D., Volpe, B. T., and Hogan, N. (2004). Rehabilitation robotics: pilot trial of a spatial extension for mit-manus. *Journal of NeuroEngineering and Rehabilitation*, 1(1):1.

Krebs, H. I., Hogan, N., Aisen, M. L., and Volpe, B. T. (1998). Robot-aided neurorehabilitation. *Rehabilitation Engineering, IEEE Transactions on*, 6(1):75–87.

Krebs, H. I., Volpe, B. T., Williams, D., Celestino, J., Charles, S. K., Lynch, D., and Hogan, N. (2007). Robot-aided neurorehabilitation: a robot for wrist rehabilitation. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, 15(3):327.

Kruchten, P. (2004). *The rational unified process: an introduction*. Addison-Wesley Professional, Wokingham, UK.

Laver, K., George, S., Thomas, S., Deutsch, J. E., and Crotty, M. (2012). Virtual reality for stroke rehabilitation. *Stroke*, 43(2):e20–e21.

Li-xin, W. (2009). On the physics engine to use the physx sdk. *Computer Knowledge and Technology*, 5(20):5561–5562.

Likert, R. (1967). The method of constructing and attitude scale. *Methods and Techniques in Business Research*, page 54.

LLinares, A., Badesa, F. J., Morales, R., Garcia-Aracil, N., Sabater, J., and Fernandez, E. (2013). Robotic assessment of the influence of age on upper-limb sensorimotor function. *Clinical interventions in aging*, 8:879.

Lloréns, R., Gil-Gómez, J.-A., Mesa-Gresa, P., Alcañiz, M., Colomer, C., and Noé, E. (2011). Biotrak: a comprehensive overview. In *2011 International Conference on Virtual Rehabilitation*, pages 1–6. IEEE.

Loureiro, R. C. and Harwin, W. S. (2007). Reach & grasp therapy: design and control of a 9-dof robotic neuro-rehabilitation system. In *2007 IEEE 10th International Conference on Rehabilitation Robotics*, pages 757–763. IEEE.

Loureiro, R. C., Lamperd, B., Collin, C., and Harwin, W. S. (2009). Reach & grasp therapy: Effects of the gentle/g system assessing sub-acute stroke whole-arm rehabilitation. In *2009 IEEE International Conference on Rehabilitation Robotics*, pages 755–760. IEEE.

Lum, P. S., Burgar, C. G., Shor, P. C., Majmundar, M., and Van der Loos, M. (2002). Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. *Archives of physical medicine and rehabilitation*, 83(7):952–959.

Maciejasz, P., Eschweiler, J., Gerlach-Hahn, K., Jansen-Troy, A., and Leonhardt, S. (2014). A survey on robotic devices for upper limb rehabilitation. *Journal of neuroengineering and rehabilitation*, 11(1):1.

Maciel, A., Halic, T., Lu, Z., Nedel, L. P., and De, S. (2009). Using the physx engine for physics-based virtual surgery with force feedback. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5(3):341–353.

Mehrholz, J., Platz, T., Kugler, J., and Pohl, M. (2008). Electromechanical and robot-assisted arm training for improving arm function and activities of daily living after stroke. *Cochrane Database Syst Rev*, 4(4).

Merians, A. S., Poizner, H., Boian, R., Burdea, G., and Adamovich, S. (2006). Sensorimotor training in a virtual reality environment: does it improve functional recovery poststroke? *Neurorehabilitation and neural repair*, 20(2):252–267.

Mihelj, M., Novak, D., Ziherl, J., Olenšek, A., and Munih, M. (2011). Challenges in biocooperative rehabilitation robotics. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–6. IEEE.

Miller, E. L., Murray, L., Richards, L., Zorowitz, R. D., Bakas, T., Clark, P., Billinger, S. A., on Cardiovascular Nursing, A. H. A. C., et al. (2010). Comprehensive overview of nursing and interdisciplinary rehabilitation care of the stroke patient a scientific statement from the american heart association. *Stroke*, 41(10):2402–2448.

Mirelman, A., Patritti, B. L., Bonato, P., and Deutsch, J. E. (2010). Effects of virtual reality training on gait biomechanics of individuals post-stroke. *Gait & posture*, 31(4):433–437.

Morales, R., Badesa, F. J., Garcia-Aracil, N., Perez-Vidal, C., Sabater, J. M., Papaleo, E., Salerno, A., Zollo, L., and Guglielmelli, E. (2014). Patient-tailored assistance. *Robotics and Automation Magazine*.

Munih, M., Riener, R., Colombo, G., Lunenburger, L., Muller, F., Slater, M., and Mihelj, M. (2009). Mimics: Multimodal immersive motion rehabilitation of upper and lower extremities by exploiting biocooperation principles. In *2009*

IEEE International Conference on Rehabilitation Robotics, pages 127–132. IEEE.

Nakayama, H., Jgtgensen, H., Raaschou, H. O., and Olsen, T. S. (1994). Recovery of upper extremity function in stroke patients: the copenhagen stroke study. *Age (SD)*, 74(11):12.

Nielen, M., Heslenfeld, D., Heinen, K., Van Strien, J., Witter, M., Jonker, C., and Veltman, D. (2009). Distinct brain systems underlie the processing of valence and arousal of affective pictures. *Brain and Cognition*, 71(3):387–396.

Norouzi-Gheidari, N., Archambault, P. S., and Fung, J. (2012). Effects of robot-assisted therapy on stroke rehabilitation in upper limbs: systematic review and meta-analysis of the literature. *Journal of rehabilitation research and development*, 49(4):479.

Novak, D., Mihelj, M., and Munih, M. (2012). A survey of methods for data fusion and system adaptation using autonomic nervous system responses in physiological computing. *Interacting with computers*, 24(3):154–172.

Novak, D., Mihelj, M., Ziherl, J., Olensek, A., and Munih, M. (2011). Psychophysiological measurements in a biocooperative feedback loop for upper extremity rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(4):400–410.

OgreXmlConverter (2015). Available online: `http://www.ogre3d.org/tikiwiki/OgreXmlConverter/` (accessed on 12 february 2015).

Papaleo, E., Zollo, L., Garcia-Aracil, N., Badesa, F., Morales, R., Mazzoleni, S., Sterzi, S., and Guglielmelli, E. (2015). Upper-limb kinematic reconstruction during stroke robot-aided therapy. *Medical & biological engineering & computing*, 53(9):815–828.

Pessoa, L. (2008). On the relationship between emotion and cognition. *Nature reviews neuroscience*, 9(2):148–158.

Pollock, A., Farmer, S. E., Brady, M. C., Langhorne, P., Mead, G. E., Mehrholz, J., and van Wijck, F. (2014). Interventions for improving upper limb function after stroke. *Cochrane Database Syst Rev*, 11.

Posner, J., Russell, J. A., Gerber, A., Gorman, D., Colibazzi, T., Yu, S., Wang, Z., Kangarlu, A., Zhu, H., and Peterson, B. S. (2009). The neurophysiological bases of emotion: An fmri study of the affective circumplex using emotion-denoting words. *Human brain mapping*, 30(3):883–895.

Rand, D., Kizony, R., Feintuch, U., Katz, N., Josman, N., Weiss, P. L. T., et al. (2005). Comparison of two vr platforms for rehabilitation: video capture versus hmd. *Presence: Teleoperators and Virtual Environments*, 14(2):147–160.

Riener, R. (2012). Technology of the robotic gait orthosis lokomat. In *Neurorehabilitation Technology*, pages 221–232. Springer.

Rose, F. D., Brooks, B. M., and Rizzo, A. A. (2005). Virtual reality in brain damage rehabilitation: review. *CyberPsychology & Behavior*, 8(3):241–262.

Rumbaugh, J., Jacobson, I., and Booch, G. (2004). *The Unified Modeling Language Reference Manual*. Pearson Higher Education.

Russell, J. A. (1978). Evidence of convergent validity on the dimensions of affect. *Journal of personality and social psychology*, 36(10):1152.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.

Saltares Márquez, D. (2011). Iberogre, wiki de ogre3d en español y sion tower, videojuego de estrategia.

Saposnik, G. (2016). Virtual reality in stroke rehabilitation. In *Ischemic Stroke Therapeutics*, pages 225–233. Springer.

Saposnik, G., Levin, M., Group, S. O. R. C. S. W., et al. (2011). Virtual reality in stroke rehabilitation a meta-analysis and implications for clinicians. *Stroke*, 42(5):1380–1386.

Sensable, P. O. (2015). Available online: `http://www.dentsable.com/haptic-phantom -omni.htm/` (accessed on 8 february 2015).

Shiroma, E. J., Ferguson, P. L., and Pickelsimer, E. E. (2012). Prevalence of traumatic brain injury in an offender population: A meta-analysis. *The Journal of head trauma rehabilitation*, 27(3):E1–E10.

Shu, F. and Tan, A.-H. (2012). A biologically-inspired affective model based on cognitive situational appraisal. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE.

Simonetti, D., Zollo, L., Papaleo, E., Carpino, G., and Guglielmelli, E. (2016). Multimodal adaptive interfaces for 3d robot-mediated upper limb neurorehabilitation: An overview of bio-cooperative systems. *Robotics and Autonomous Systems*, 85:62 – 72.

Small, D. M., Gregory, M. D., Mak, Y. E., Gitelman, D., Mesulam, M. M., and Parrish, T. (2003). Dissociation of neural representation of intensity and affective valuation in human gustation. *Neuron*, 39(4):701–711.

Soyuer, F. and Öztürk, A. (2007). The effect of spasticity, sense and walking aids in falls of people after chronic stroke. *Disability and rehabilitation*, 29(9):679–687.

Staines, W., McIlroy, W., Graham, S., and Black, S. (2001). Bilateral movement enhances ipsilesional cortical activity in acute stroke: a pilot functional mri study. *Neurology*, 56(3):401–404.

Steve, S. (2013). Ogre3d: The object-oriented graphics rendering engine.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society. Series B (Methodological)*, pages 111–147.

Subramanian, S. K. and Levin, M. F. (2011). Viewing medium affects arm motor performance in 3d virtual environments. *Journal of neuroengineering and rehabilitation*, 8(1):1.

Sveistrup, H. (2004). Motor rehabilitation using virtual reality. *Journal of neuroengineering and rehabilitation*, 1(1):1.

Tagliaferri, F., Compagnone, C., Korsic, M., Servadei, F., and Kraus, J. (2006). A systematic review of brain injury epidemiology in europe. *Acta neurochirurgica*, 148(3):255–268.

Tamura, Y., Kageyama, A., Sato, T., Fujiwara, S., and Nakamura, H. (2001). Virtual reality system to visualize and auralize numerical simulation data. *Computer physics communications*, 142(1):227–230.

Toledo-Moreo, R., Pinzolas-Prado, M., and Cano-Izquierdo, J. M. (2010). Maneuver prediction for road vehicles based on a neuro-fuzzy architecture with a low-cost navigation unit. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):498–504.

TURNER, P. (2006). Crazy eddie's gui system.

Turolla, A., Dam, M., Ventura, L., Tonin, P., Agostini, M., Zucconi, C., Kiper, P., Cagnin, A., and Piron, L. (2013). Virtual reality for the rehabilitation of the upper limb motor function after stroke: a prospective controlled trial. *Journal of neuroengineering and rehabilitation*, 10(1):1.

van der Laan, W. J., Green, S., and Sainz, M. (2009). Screen space fluid rendering with curvature flow. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pages 91–98. ACM.

Van der Linde, R. Q., Lammertse, P., Frederiksen, E., and Ruiter, B. (2002). The hapticmaster, a new high-performance haptic interface. In *Proc. Eurohaptics*, pages 1–5.

Vasudevamurt, V. B. and Uskov, A. (2015). Serious game engines: Analysis and applications. In *Proc. IEEE Int. Conf. Electro/Information Technology (EIT)*, pages 440–445.

Volpe, B. T., Huerta, P. T., Zipse, J. L., Rykman, A., Edwards, D., Dipietro, L., Hogan, N., and Krebs, H. I. (2009). Robotic devices as therapeutic and diagnostic tools for stroke recovery. *Archives of neurology*, 66(9):1086–1090.

Weiss, P. L., Sveistrup, H., Rand, D., and Kizony, R. (2009). Video capture virtual reality: a decade of rehabilitation assessment and intervention. *Physical Therapy Reviews*, 14(5):307–321.

Winkler, R. G., Ripoll, M., Mussawisade, K., and Gompper, G. (2005). Simulation of complex fluids by multi-particle-collision dynamics. *Computer physics communications*, 169(1):326–330.

Witmer, B. G. and Singer, M. J. (1998). Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and virtual environments*, 7(3):225–240.

Woo, M., Neider, J., Davis, T., et al. (1997). Opengl programming guide.

Yang, S., Chun, M. H., and Son, Y. R. (2014). Effect of virtual reality on cognitive dysfunction in patients with brain tumor. *Annals of rehabilitation medicine*, 38(6):726–733.

Zollo, L., Gallotta, E., Guglielmelli, E., and Sterzi, S. (2011a). Robotic technologies and rehabilitation: new tools for upper-limb therapy and assessment in chronic stroke. *European journal of physical and rehabilitation medicine*, 47(2):223–236.

Zollo, L., Rossini, L., Bravi, M., Magrone, G., Sterzi, S., and Guglielmelli, E. (2011b). Quantitative evaluation of upper-limb motor control in robot-aided rehabilitation. *Medical & biological engineering & computing*, 49(10):1131–1144.

UNIVERSITAS
Miguel
Hernández

Parte II

Publicaciones Aportadas

UNIVERSITAS
Miguel
Hernández

PLOS ONE

RESEARCH ARTICLE

Supervised and Dynamic Neuro-Fuzzy Systems to Classify Physiological Responses in Robot-Assisted Neurorehabilitation

Luis D. Lledó[1☯]*, Francisco J. Badesa[1☯], Miguel Almonacid[2‡], José M. Cano-Izquierdo[2‡], José M. Sabater-Navarro[1‡], Eduardo Fernández[1‡], Nicolás Garcia-Aracil[1☯]

1 Biomedical Neuroengineering Group, Universidad Miguel Hernández, Elche, Alicante, Spain, **2** Systems Engineering and Automation Department, Universidad Politécnica de Cartagena, Cartagena, Murcia, Spain

☯ These authors contributed equally to this work.
‡These authors also contributed equally to this work.
* lledo@umh.es

Citation: Lledó LD, Badesa FJ, Almonacid M, Cano-Izquierdo JM, Sabater-Navarro JM, Fernández E, et al. (2015) Supervised and Dynamic Neuro-Fuzzy Systems to Classify Physiological Responses in Robot-Assisted Neurorehabilitation. PLoS ONE 10(5): e0127777. doi:10.1371/journal.pone.0127777

Academic Editor: Catalin Buiu, Politehnica University of Bucharest, ROMANIA

Received: January 8, 2015

Accepted: April 19, 2015

Published: May 22, 2015

Data Availability Statement: The authors confirm that all data underlying the findings are fully available without restriction. All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by the European Commission (ICT-22-2014: Multimodal and Natural computer interaction) through the project AIDE: "Adaptive Multimodal Interfaces to Assist Disabled People in Daily Activities" (grant agreement no: 645322).

Abstract

This paper presents the application of an Adaptive Resonance Theory (ART) based on neural networks combined with Fuzzy Logic systems to classify physiological reactions of subjects performing robot-assisted rehabilitation therapies. First, the theoretical background of a neuro-fuzzy classifier called S-dFasArt is presented. Then, the methodology and experimental protocols to perform a robot-assisted neurorehabilitation task are described. Our results show that the combination of the dynamic nature of S-dFasArt classifier with a supervisory module are very robust and suggest that this methodology could be very useful to take into account emotional states in robot-assisted environments and help to enhance and better understand human-robot interactions.

Introduction

Recent developments in robotic technology have shown that robotic devices are able to play important roles in neurorehabilitation [1, 2], however there are still many challenges to be solved. As result, despite the increasing popularity of robots in neurorehabilitation, their effectiveness is still discussed controversially. One important issue in this field is to promote active patient participation in the loop control, which refers to the concept of acting cooperatively to the human instead of treating the human as a source of perturbation. Therefore an ideal robotic assisted device should be able to decide which level of difficulty should be applied in different rehabilitative scenarios taking into account biomechanical information as well as physiological and emotional aspects of the patients underlying robot-assisted therapies.

Emotion is a complex state of feeling which involves psychological and physiological reactions produced by the interactions between human being's and the environment. A widely accepted classification of emotions, called also affective states, describe them as a circumplex with two dimensions: valence and arousal [3]. Valence can take values from displeasure state to pleasure state and on the other hand, arousal can take values from deactivation state (from

Competing Interests: The authors have declared that no competing interests exist.

sleep to drowsiness) to activation state (from various stages of alertness to frenetic excitement). The published studies of neural systems involved using neuroimaging techniques suggest that valence and arousal may be associated with separate neural circuits containing the amygdala, insula, thalamus, dorsal anterior cingulate cortex, and prefrontal regions [4–10]. Most of these studies show that the amygdala is the core of affective region suggesting that this region may belong to both valence and arousal neural systems.

The Network theory can be applied as well to the neural computation of emotion as it is described in Pessoa's conceptual proposal of neural computations and emotion [11]. Based on this assumption, a Cognitive Regulated Affective Architecture (CRAA), which comprises a cognitive network, an affective network and an appraisal layer is proposed by Feng [12]. In addition, the affective network was designed to simulate the functions in amygdala and was built using a neural network based on Adaptive Resonance Theory (ART) models. For this reason, we hypothesized that Neural Networks, specially ART based neural networks, should work better than previous classifiers implemented to estimate the user's emotional state based on physiological reactions in rehabilitation therapies assisted by robotic devices. An exhaustive list comparing the use of different classification algorithms accordingly with number of subjects enrolled in the studies, the classes used, the type of classifier with its accuracy and so on for psychophysiological studies were summarized and classified in Novak et al. [13].

In a previous work [14], we used nine machine learning techniques to estimate different user's states such as bored-relaxed, pleased and excited-aroused. Our results showed that the Support vector machines (SVM) with Radial Basis Functions (RBF) kernels provided the best results in terms of accuracy (91.43%). However for a wider use of these technologies we need methods able to cover patients with a broad variety of physical as well as cognitive impairments and capable to adapt automatically to the patient's specific demands and needs. Therefore, due to the widely use of neural networks in neural process modelling related with emotions, in this paper we have investigated the potential usefulness of neural networks incorporating concepts of fuzzy logic theory to estimate user's emotional state and verify the performance of this technology.

Materials and Methods

Neuro-fuzzy Classifier: S-dFasArt

A neuro-fuzzy method of classification called S-dFasArt [15] (Supervised and Dynamic Fuzzy Adaptive System ART-based) has been used in this work to classify temporal patterns of a physiological signals set acquired during rehabilitation therapies assisted by a robotic device. This method combine the properties of neural networks based on Adaptive Resonance theory (more specifically is based on fuzzy ARTMAP architecture [16]), and the fundamentals of the Fuzzy Sets theory using a supervised-competitive learning and dynamic equations for the processing stages of the algorithm.

Its neuro-fuzzy architecture takes advantage of the learning capacity and adaptation of the neural network, and the robustness-interpretability of fuzzy systems. Moreover, the properties of the learning algorithm, the update mode and speed of convergence of the weights are improved. The proposed neuro-fuzzy architecture satisfies the stability-plasticity criterion since the classifier is able to maintain the accumulated knowledge and acquire new learning patterns and allows a quick learning with a small set of training patterns as well.

Due to competitive learning, all nodes or output categories react to an input value but the classifier only active the neuron with the highest response level. The category associated with the winner node is the classification of the network for the current input pattern.

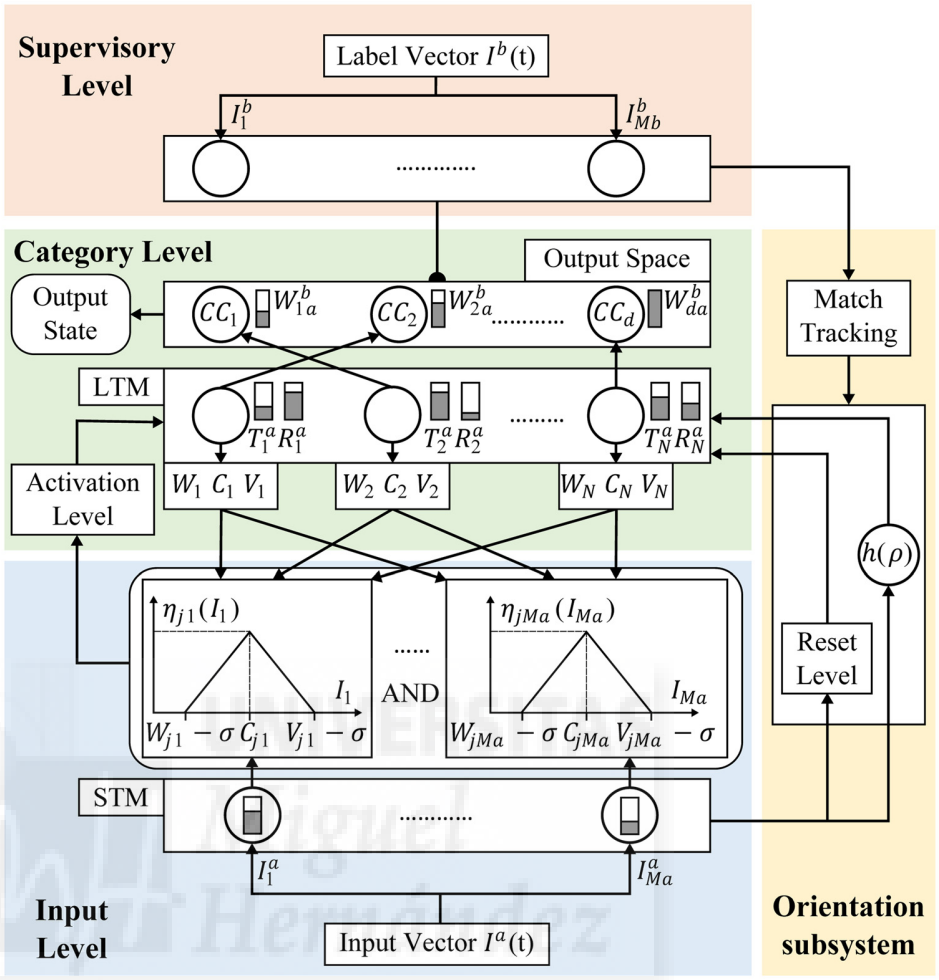


Fig 1. S-dFasArt Architecture.

doi:10.1371/journal.pone.0127777.g001

S-dFasArt Architecture. Fig 1 shows the general architecture of the proposed classifier model, combining neural and fuzzy operations. This model consists of the following elements: an Input Level, a Supervisory Level, an Orientation Subsystem, and a Category Level.

The Input level is formed by M_a nodes. Each node receives one element of the I^a vector to hold temporally the most important input aspects, activating the short-term memory STM. A fuzzy block integrated by a triangular fuzzy activation-membership function η_{ji} is associated to each of the input nodes. These blocks measure the membership degree of each input characteristics i regarding each fuzzy category j. The size of the membership function can be determined by the design parameter σ. So, this parameter modifies the diffuse character of the output categories [17].

The Supervisory Level has M_b nodes to present the pattern with the correct classification state I^b associated to an input vector. This level configures the output space by creating a number d of classes with each of possible classifications that the network can encode. The input of the supervision vectors is only provided to the network during the learning stage.

The Category Level is formed by a set of N nodes representing all the categories that have been created during the learning process, resulting in a set of fuzzy units. Each node or category has two main associated values: T_j indicating the degree of activacion and R_j indicating the ability to learn from that input. These categories are activated with a determined level T_j when an input pattern is presented to the classifier. Three kinds of weights, minimun W_{ji}, central C_{ji}, and maximun V_{ji}, are associated to each output unit. These weights store the long-term memory of the network, whose values are linked with the fuzzy blocks to update the membership functions that handle the acceptability level of the input pattern depending on the category that is reacting. The activation of output fuzzy categories is calculated using an AND operation of all the fuzzy degrees of membership of the input vector characteristics. Each fuzzy category can only encode one output state CC of the d possibilities that have been generated in the network during the learning phase from the supervision values. So different categories may point to the same classification node in the output space. Therefore, this level is responsible for linking the sequence pairs of input vectors with the supervision vectors.

There is also an Orientation Subsystem to detect the similarity of the input vector with the categories learned by the network. This similarity percentage can be compared with a vigilance parameter $h(\rho)$ to control the number of diffuse categories that must be created in the Level Category. The vigilance parameter determines how strict the network must be in the classification process of the input measures, generalizing the results. The match tracking manages the network for the vigilance parameter is automatically set, indicating whether the entry was correctly classified in the node j or if the model has to create another output category. If the similarity calculated by the Orientation Subsystem is not sufficiently similar to a category, a reset level Rj is produced to disable current category and choose another category following the maximum similarity criterion.

Learning Algorithm. In this section, the training algorithm implemented by the classifier during its learning stage is presented (Fig 2). A competitive learning is applied to generate new fuzzy categories, introducing the input data to the network only once, in the temporal order that these data are obtained and processed. With this process an update of the category weights is achieved. A more comprehensive description of the learning algorithm, its dynamics equations and the meaning of the network parameters, can be found in [18] and [19].

Nodes of the Category Level compete among themselves, but the learning phase only occurs in the neuron with the highest activation level. The S-dFasArt algorithm can be described with the following steps:

1. Firstly, the classifier is initialized, defining the values of the parameters of the dynamic equations, such as the activation rate, the growth rate of the level reset, time constants, linked gains or the vigilance parameter. These parameters affect the learning behavior directly. Initially, no input pattern has been presented to the network, so this network does not have any node associated with weights in the Level Category.
2. The training measures $I^a(t)$ and its corresponding supervision labels $I^b(t)$ assigned to this data are computed, getting K number of samples.
3. One input pattern h is presented to the network. Each node of the Input Level receives a feature of the input vector, activating the short-term memory to indicate the presence degree of each signal attributes. Also, the supervision vector is presented to the Supervisory Level, activating its nodes.
4. When the network gets an input pattern and a supervision label, checks whether there are categories to calculate their levels of activation and reset. If the network has no category, this step is not performed.

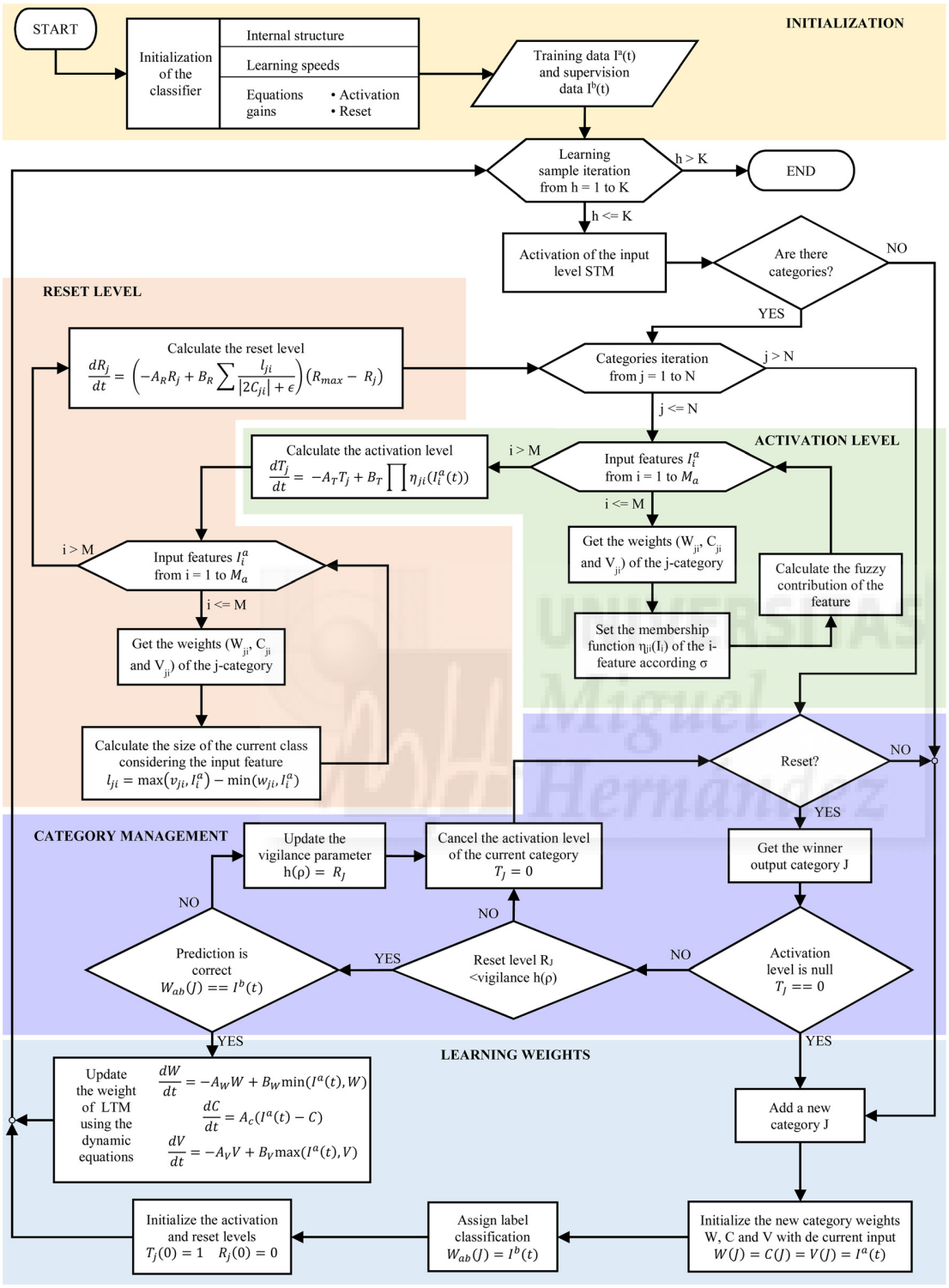


Fig 2. S-dFasArt Learning Algorithm.

doi:10.1371/journal.pone.0127777.g002

5. If there are categories, the following step sequence is executed for each of the existing j-categories:

   a. The current category j sends the weights W_{ji}, C_{ji} and V_{ji} to the fuzzy logic block associated with each of the i-nodes in the Input Level and updates its activation-membership function, considering the σ parameter to control the size and the diffuse character of the categories.

   b. Then, the value of fuzzy membership associated with each input neuron that determine the activation level of each feature of the input vector is computed. The level activation T_j of the processed category can be calculated from these values and the AND operation, using the dynamic equation responsible of the activation process of the S-dFasArt algorithm (Eq 1). The parameter A_T is the activation speed of the categories, and determines the sensibility of the network to respond to input changes. $\eta_{ji}(I_i^a)$ represents the fuzzy contribution of each feature of the input sample to compute the activation level, applying the activation-membership function. While the parameter B_T specifies the dynamic gain of the total fuzzy contribution.

$$\frac{dT_j}{dt} = -A_T T_j + B_T \prod_{i=1}^{M} \eta_{ji}(I_i^a(t)) \tag{1}$$

   c. Next, the size of the fuzzy category is calculated considering the input data as a pattern attached to this category, i.e. the similarity degree that indicates if the input pattern is a subset of the weights of the associated category. Therefore, each i-node of input provides a magnitude (Eq 2) to calculate the total size of the category as the sum of all these values (Eq 3).

$$l_{ji} = max(V_{ji}, I_i^a) - min(W_{ji}, I_i^a) \tag{2}$$

$$dreset = \sum_{i=1}^{M} \frac{l_{ji}}{|2C_{ji}| + \epsilon} \tag{3}$$

   d. This size is used to calculate the level reset of the current category from the dynamic equation of S-dFasArt reset (Eq 4). R_j can be considered as a similarity threshold required for an input vector can be associated with the category j. A_R involves the growing speed of the Reset Level that allows the system to be able to respond to the dynamic changes of the input data. This implies the aptitude of the category to learn from the input that is activating it. The B_R parameter determines the gain linked to the total size of the category. R_{max} denotes the maximum Reset value for the category.

$$\frac{dR_j}{dt} = (-A_R R_j + B_R dreset)(R_{max} - R_j) \tag{4}$$

6. At this point, the levels of activation and reset of the N categories that currently exist in the network have been calculated. Then, the winner category J is determined selecting the

category with the highest activation value (Eq 5).

$$T_J = max\{T_j; \quad j = 1...N\} \tag{5}$$

7. If the maximum activation level is null ($T_j = 0$), an uncommitted category is added. To establish the weights of this uncommitted node as a prototype of the input pattern, a fast commit is applied (Eq 6). Then, the label received in the Supervisory Level is assigned as classification result of the current category (Eq 7) in the output space. Also, the levels of the activation and reset at the first instant of time are initialized (Eq 8).

$$W = C = V = I^a(t) \tag{6}$$

$$W_{ab} = I^b(t) \tag{7}$$

$$T_j(0) = 1 \qquad R_j(0) = 0 \tag{8}$$

8. However, if the winner category have a no null activation level, its reset level is compared with the vigilance parameter. If the reset value exceeds the vigilance threshold, a reset state occurs because the winner neuron does not properly represent the category in which the current input pattern belongs. Then, the Orientation Subsystem temporarily disables the node J ($T_J = 0$) and selects the category whose level activation is the next highest.
9. If the reset level is less than the vigilance parameter, the classification label of the winner category is compared with the data $I^b(t)$ received in the Supervisory Level. If these values do not match, the prediction is incorrect, consequently the vigilance parameter is automatically set using the value of the current reset level to search a new category, canceling its activation level $T_J = 0$.
10. If the classification state of the winner neuron is the same as the label of the Supervisory Level, the activated node is the category most appropriate for the current learning sample $I^a(t)$. Thus, an updating process of the weights W_{ji}, C_{ji} and V_{ji} is applied in the selected category using a slow recode to increase its resemblance to the input data. This update of the weights is performed from the dynamic equations of S-dFasArt learning (Eq 9). The parameters A_W, A_C, A_V, B_W and B_V can be interpreted as learning speeds associated to the growing of the minimun, central and maximum weights respectively of the fuzzy categories.

$$\begin{aligned} \frac{dW}{dt} &= -A_W W + B_W min(I^a(t), W) \\ \frac{dC}{dt} &= A_C(I^a(t) - C) \\ \frac{dV}{dt} &= -A_V V + B_V max(I^a(t), V) \end{aligned} \tag{9}$$

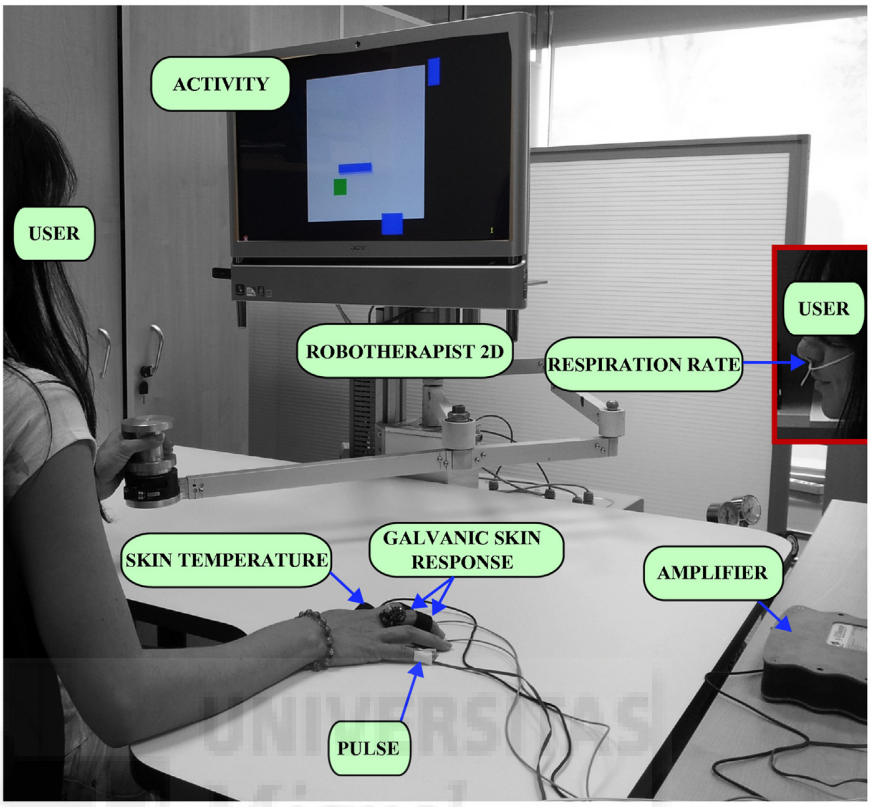


Fig 3. One subject during the experiments.

doi:10.1371/journal.pone.0127777.g003

Methodology and Experiments

The experiments were conducted as we have previously reported [14]. Briefly, a hardware configuration composed by a robotic device, a signal acquisition system and a virtual reality system were used to perform the requested activities and to monitor in real time the user's physiological signals (pulse rate, respiration rate, skin conductance level (SCL), skin conductance response (SCR) and skin temperature). The robotic device used for these experiments was the PUPArm robot system, a commercial platform designed for upper-limb assisted therapy which is now commercialised by Instead Technologies Inc with the trading name of "RoboTherapist 2D" (see Fig 3).

A specific activity to induce different user's psychophysiological states was designed. The activity consisted of three main components: the area of activity, bounded by a black frame, the pointer that moves the user represented by a green square and a series of blue rectangles of different sizes moving randomly across the screen. The goal of the activity was to move freely around the screen avoiding the collision with the blue rectangles without leaving the space delimited by the black frame. Every time the subject touches a blue box or leaves the area of activity means a mistake what implies that the user square turns red and sounds a shrill sound. Three levels of difficulty: relax, medium and stress level were defined depending on the number of blue rectangles and their speed.

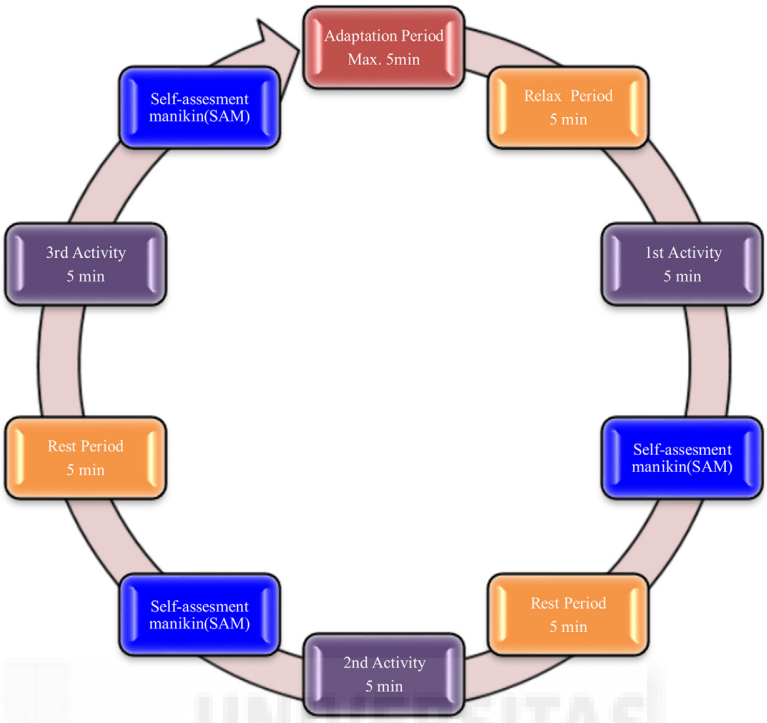


Fig 4. Experiment protocol. This protocol is composed of three activities and rest periods before and after activities.

doi:10.1371/journal.pone.0127777.g004

Human data presented in this article have been acquired under an experimental protocol approved by the Medical Ethics Committee of the Universidad Miguel Hernandez of Spain and all subjects gave written informed consent. Seven volunteers participated in the experiments. All were healthy, without cognitive or physical deficits. They were aged between 26 and 42 (mean age 31 years, median age 29 years, standard deviation 6.3 years). The whole experimental protocol is shown in Fig 4. After each activity, a self-assessment manikin (SAM) was presented to the subjects to measure their affective responses. The dataset used in this experimentation is provided in S1 Dataset.

Once the physiological signals were acquired, a normalization of the features was completed as we have previously reported [14]. After that, a Principal Components Analysis (PCA) was used in order to study the possibility of further reducing the number of input features for the machine learning algorithms and the proposed neuro-fuzzy model based on the S-dFasArt architecture.

Results

A test classification was performed to check the possibilities of the proposed neuro-fuzzy model based on the S-dFasArt architecture, using the cross-validation technique called Leave-one-Out (LOOCV). The network generalization in situations where the network was not trained was examined with this validation method, estimating the performance of the classifier. This technique is suitable when the experimental data do not contain sufficient measurements.

An adjustment process of the S-dFasArt classifier should be applied to obtain a model of functional classification, presenting a set of learning samples to the algorithm explained in Materials and Methods section. The learning information data has been computed by acquiring and processing the physiological responses of seven subjects.

The adjustment of the classifier basically can be divided into three phases. First, the parameters linked to the dynamic equations of S-dFasArt are initialized with default values. Secondly, a learning of the weights that represent the fuzzy categories is effected. Finally, a phase of parameter adjustment is applied to calculate the two network parameters that have more influence in the classification data, and get the better interpretation values. These parameters are related to diffuse character σ of the categories and their activation speed A_T. However, the parameter A_T is not adjusted in this work and its default value is maintained due to the nature of the validation method, where one sample is only tested in each iteration, therefore a continuous calculation of the categories activations is not required.

Before starting the training of the network, the parameter A_R is set. This parameter controls the number of categories that the network generates. First, a quantitative study of the categories that are generated based on A_R is performed to establish a range of values which allow a reasonable generation of categories for the amount of learning samples presented to the network. Thus, the creation of categories is limited, avoiding an excessive generation due to an overtraining caused by the categories proliferation problem that S-dFasArt inherits from the ARTMAP architecture and providing a generalization of the categories.

Then, the values of the parameter A_R is established between a range from 0 to 20 in order to compute the categories that the S-dFasArt architecture can generate using this data type. The number of committed categories (nodes) is not prefixed beforehand in the models based in ARTMAP. This value depends of the learning processing. In S-dFasArt, the categories are generated depending on the number of learning samples which are supplied to the model. One classification model is implemented for each value of A_R applying default values of network and all the learning samples. After that, the number of categories for all models is computed. Fig 5 shows the generation in the fuzzy categories depending on the growing speed value A_R of the reset level.

An increase in the amount of categories was observed in the graph, due to the growth of the A_R value. Therefore, A_R values that achieve a number of categories between a range from 20 to 40 were selected to avoid the generation of too many categories which could cause the network to incorrectly classifies the samples. The range of generated categories is establised with these values because the dataset had 105 samples, 15 per user. Then, the A_R value was set between 0.05 and 2.15. Finally the Leave-one-out technique was used to validate the S-dFasArt classifier.

With this validation technique, K-classification models are generated according to the amount of learning measures that have been extracted from the physiological signals. In each iteration, a classification model is created using all the learning samples with the exception of one measure that is removed from the set. These samples are classified with the trained network during the test phase. The remaining data set is used in the weights learning phase during the adjustment process of the classifier. This procedure is repeated one time for each measure, then the classification success percentage of the network is computed using the arithmetic mean of the iterative process.

Now, the value set of parameters A_R and σ that offers better results is calculated using the trial and error method. Since the range of G-values of the parameter A_R is already configured, σ is established with a range of L-values between 0.0001 and 1.0, so this parameter has values that cover a wide range of sizes of the triangular fuzzy activation-membership function. In this way, the validation method Leave-one-out is performed G*L times, depending on the range

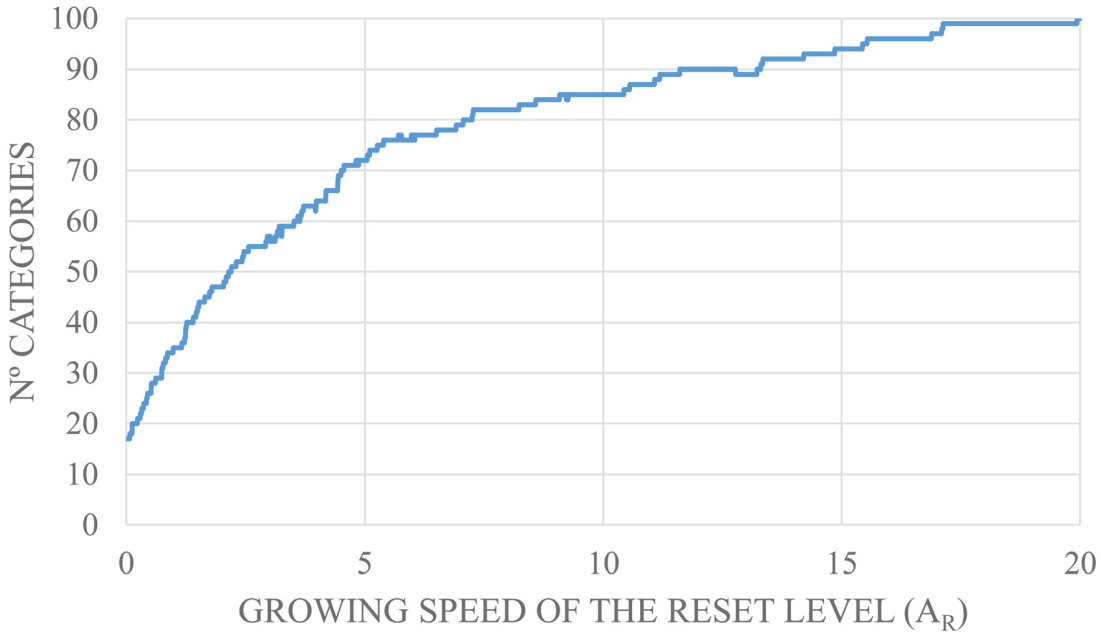


Fig 5. Proliferation of categories.

doi:10.1371/journal.pone.0127777.g005

value of the A_R-σ data set. To conclude, the result with the highest success rate provides the best values for A_R-σ data set.

The remaining parameters are kept constants. The reason for this is that its default values provide the generation of models with the best classification rates. These values are also obtained through trial and error. Table 1 shows the values applied during the process of adjustment and validation of the classifier, together with a brief explanation for each parameter.

Table 1. S-dFasArt network parameters.

PARAMETER	VALUE	DESCRIPTION
A_R	0.05–2.15	Growing speed of the reset level
σ	0.0001–1.0	Diffuse character of the fuzzy categories
A_T	0.01	Activation speed of the fuzzy categories
A_W	0.8	Growing speed of the weights associated to the fuzzy categories
A_C	0.8	
A_V	0.1	
ε	0.001	The minimum value of the diffuse character
α	1e-30	Activation value to generate new categories
$h(\rho)$	0.1	Vigilance parameter
R_{max}	0.2	The maximum reset value to disable the fuzzy categories
B_T, B_R	1	Gains of the S-dFasArt differential equations
B_W, B_V		

Values and descriptions of the S-dFasArt network parameters.

doi:10.1371/journal.pone.0127777.t001

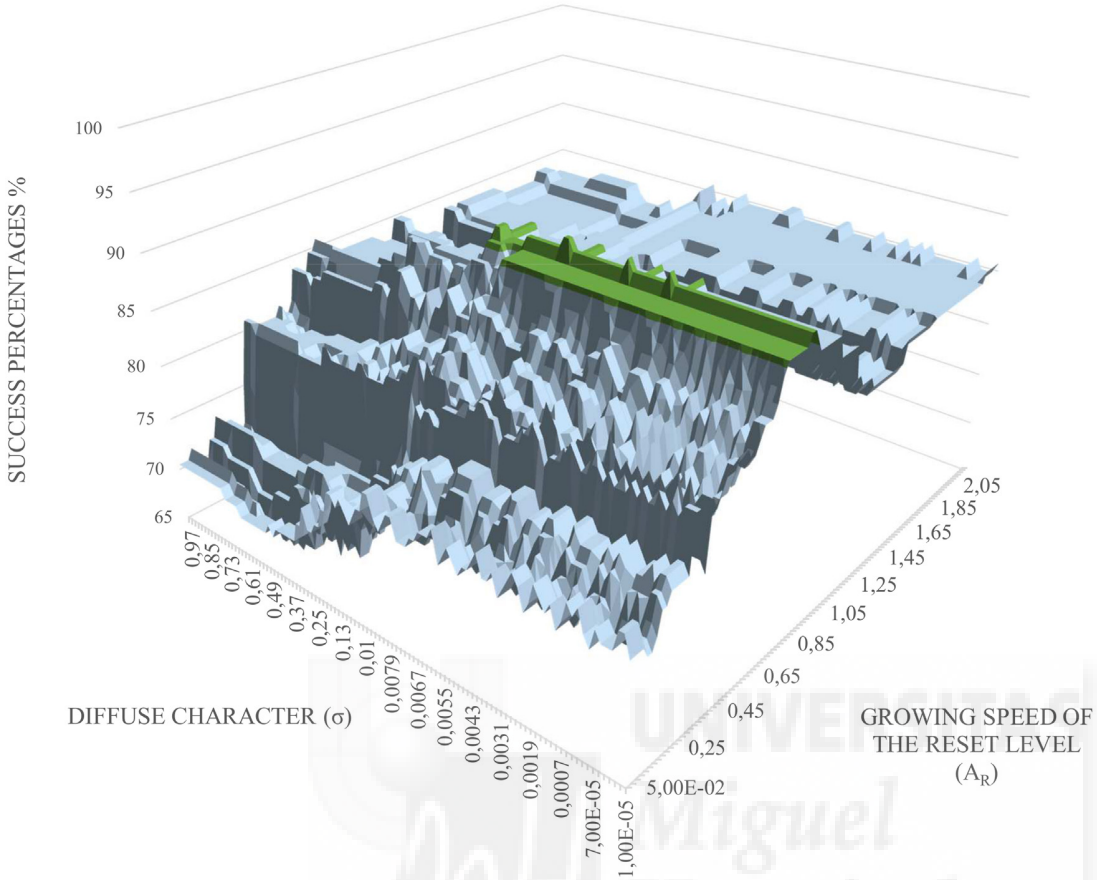


Fig 6. Success percentages depending on A_R and σ.

doi:10.1371/journal.pone.0127777.g006

Once all classification models are processed, and the sets of A_R and σ parameter values are tested, 11130 possible results have been computed. Fig 6 shows a 3D graph that collected the values of the success rates of all generated classification models. The highlighted zone of the graph are the points whose success percentages are greater than 90%. Then, Fig 7 shows a 2D plot of the A_R and σ values for a success percentage greater than 90%.

However, to get the highest success percentage, the selected values are the ones in the maximum peak of the graph. At this point, the third phase of the adjustment process of the classifier is accomplished, getting the set of the parameters A_R and σ, whose values provide the better success rate. Final results during the development of this experiment are summarized in Table 2. From the 11130 possible computed models, five classification models that obtained the best results in terms of success rate, have been selected. The first column shows the categories generated by the iterations of the validation method. The next two columns recollects the values of the adjusted parameters obtained in the adjustment phase. The A_R parameter affects directly in the value of the generated categories. The success rates have been placed in the last column. The variability of the success rates related to A_R and σ can be observed in this Table.

The classification ranges of the S-dFasArt model compared with some of the classifiers [14] used to interpret this signal type have been collected in Table 3. The LOOCV results is

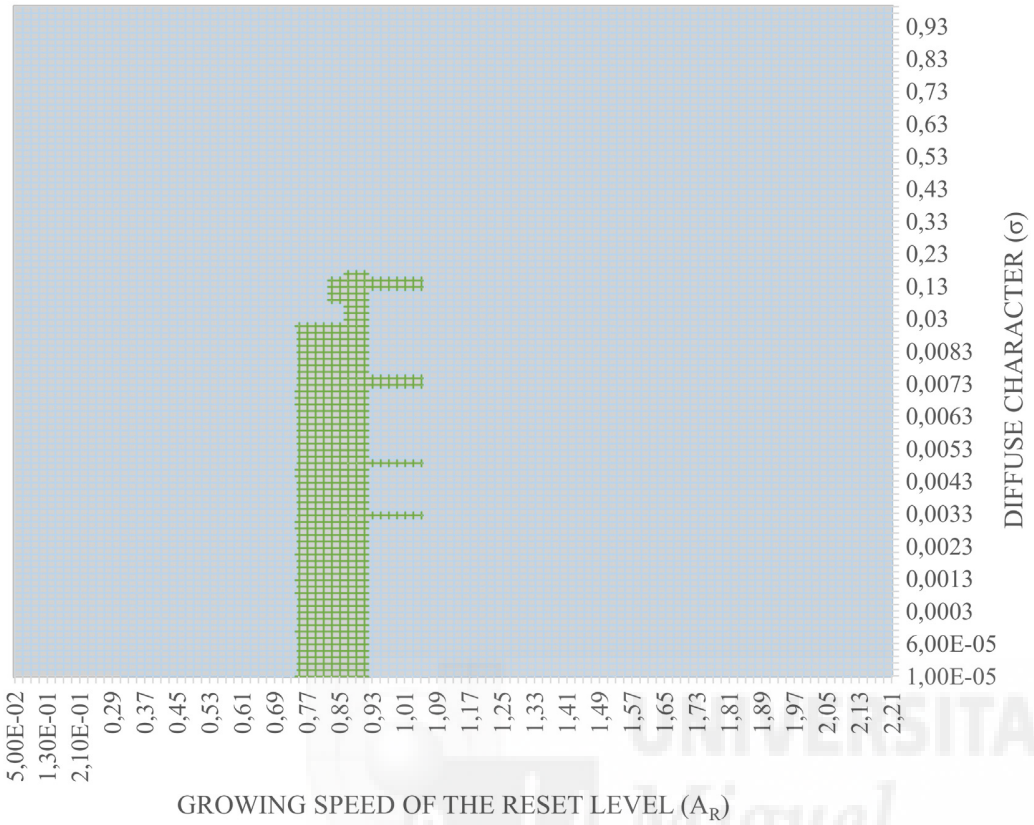


Fig 7. A_R and σ values plane.

doi:10.1371/journal.pone.0127777.g007

presented in six column table to show the classification values applying Principal Components Analysis to the input data and the results without employ any complementary processing. The performance of each of the nine machine learning and the proposed neuro-fuzzy model using different number of principal components (PC) as input data, are presented in the first five columns, while the data without PCA computation can be shown in the last column.

Discussion

The hypothesis of the present work is based on the Pessoa's conceptual proposal of neural computations and emotion. Based on this assumption, a CRAA, which comprises a cognitive

Table 2. Summary results table.

Categories	A_R	σ	Success %
33–34	0.87	0.0033	92.38
33–34	0.89	0.0083	91.43
30–31	0.75	0.01	90.48
34–35	0.99	0.17	89.52
28–29	0.69	0.05	87.62

The best success results.

doi:10.1371/journal.pone.0127777.t002

Table 3. Comparison of classification methods.

ALGORITHM	PCA 1 PC	PCA 2 PC	PCA 3 PC	PCA 4 PC	PCA 5 PC	NO PCA
PLA	56.57	81.9	83.5	82.86	82.1	83.05
LR	61.90	65.71	84.76	83.81	85.71	85.71
LDA	50.48	64.76	74.29	74.29	76.19	76.19
QDA	49.52	63.81	75.24	78.10	78.10	78.10
SVM	60.00	67.62	86.67	85.71	85.71	85.71
SVM with RBF	78.10	80.00	91.43	91.43	91.43	91.43
NB	49.52	61.90	66.67	64.76	60.00	53.33
KNN	64.76	72.38	80.95	80.95	80.95	80.95
RBF	58.10	57.05	56.10	56.10	56.29	56.19
S-dFasArt	69.52	80.95	90.48	90.48	90.48	92.38

Results of Leave-one-out cross-validation (LOOCV).

doi:10.1371/journal.pone.0127777.t003

network, an affective network and an appraisal layer, was proposed by Feng. Moreover, that affective network of CRAA was developed using a neural network based on ART models. This point was one of the pillars that support our hypothesis: "*ART based neural networks should work better than classifiers implemented on previous works since they are widely used to model neural process related with emotions*".

The results of the application of an Adaptive Resonance Theory (ART) based neural network combined with Fuzzy Logic systems, which is known as S-dFasArt, in order to classify user physiological reactions performing robot-assisted rehabilitation therapies are presented and compared with the results of the application of nine machine learning techniques. The S-dFasArt approach obtained better results in terms of accuracy (92.38% in LOOCV) than the SVM with RBF kernel model with 91.43% in LOOCV (See Table 3). These results show that the combination of the dynamic nature of dFasArt with a supervisory module produces a robust classifier capable to provide very good results despite of a small set of input data.

The proposed algorithm has been applied to problems associated with the classification of time-varying signals with high noise contamination or the classification of vehicle handling using GPS [18] data and electroencephalogram signals [15]. The signals of these type of application are very noisy and their temporary arrangements are a relevant feature to be studied. Since bio-signals used in user's emotional state based on physiological reactions are very noisy, this architecture has been tested.

S-dFasArt allows an adaptation between its complexity and the dataset used in its learning step. The complexity (categories number) increases depending of the variability of the learning data. This fact produces that initial premises have not to be assumed. The larger the dataset is, more categories are generated by the model, resulting in a more robust system. Furthermore, this algorithm is robust against the contradictory and inconsistent data, which produce serious problems in the optimization mechanisms of other types of classification models.

Also, in Table 3, the performance of the S-dFasArt when applying PCA analysis can be seen. These results indicate that the feature reduction processing of the input data did not provide any improvement. Therefore, this method have to be used without PCA processing. Moreover, it seems that the S-dFasArt would have better results with a large input data set, and it will be more robust to noisy input data than SVM with RBF approach. If this assumption is corroborated, the presented approach can be considered as the best candidate to classify user physiological reactions in robot-assisted rehabilitation therapies.

There are several reasons to use this type of classification technique to estimate the user's emotional state. One of this reasons is its performance against noisy data. Another one is its capacity of automatic adjustment to update the classification models and the networks parameters in real time. This way, the classification method can be adapted automatically to the patient's specific demands and needs. This architecture can take advantage to classify this type of signals because it is able to allow the acquisition of new learning examples without deleting the accumulated knowledge.

Supporting Information

S1 Dataset. Dataset of physiological signals. The dataset contains temporal patterns of physiological signals acquired during robot-aided rehabilitation therapies. The first column reports the number of user. The main features of these physiological signals are registered in columns 2–6. The vector produced by these five values, represents the input pattern that indicate the presence degree of each signal attributes. The input pattern is ordered as follows: Pulse Rate, Skin Conductance Level (SCL), Skin Conductance Response (SCR), Respiration Rate and Skin Temperature. The last two columns include the supervision label of each input pattern and its corresponding difficulty level.
(CSV)

Acknowledgments

This work has been supported by the European Commission (ICT-22-2014: Multimodal and Natural computer interaction) through the project AIDE: "Adaptive Multimodal Interfaces to Assist Disabled People in Daily Activities" (Grant agreement no: 645322) and The Biomedical Research Networking center (CIBER). CIBER gives support to our research group inside the line of Bioengineering and Medical imaging. Specifically, our research group, Neuroprosthesis and Neuroengineering Research Group, has been financially supported for the development of new research lines, such as, multimodal rehabilitation devices, among others.

Author Contributions

Conceived and designed the experiments: FJB NGA JMSN EF. Performed the experiments: LDL. Analyzed the data: LDL FJB NGA. Contributed reagents/materials/analysis tools: LDL JMCI MA. Wrote the paper: LDL FJB NGA JMSN EF.

References

1. Miller EL, Murray L, Richards L, Zorowitz RD, Bakas T, Clark P, et al. On behalf of the American Heart Association Council on Cardiovascular Nursing, and the Stroke Council. Comprehensive overview of nursing and interdisciplinary rehabilitation care of the stroke patient: A scientific statement from the american heart association. Stroke. 2010; 41(10): 2402–2448. PMID: 20813995
2. The Management of Stroke Rehabilitation Working Group on behalf of the Department of Veterans Affairs, Department of Defense, and The American Heart Association/American Stroke Association, VA/DoD Clinical Practice Guideline for the Management of Stroke Rehabilitation. Available: www.healthquality.va.gov. Accessed 25 Feb 2014.
3. Russell JA. A circumplex model of affect. Journal of Personality and Social Psychology. 1980; 39(6): 1161–1178. doi: 10.1037/h0077714
4. Anderson AK, Christoff K, Stappen I, Panitz D, Chahremani DG, Glover G, et al. Dissociated neural representations of intensity and valence in human olfaction. Nature Neuroscience. 2003; 6(2): 196–202. doi: 10.1038/nn1001 PMID: 12536208
5. Small DM, Gregory MD, Mak YE, Gitelman D, Mesulam MM, Parrish T. Dissociation of neural representation of intensity and affective valuation in human gustation. Neuron. 2003; 39(4): 701–711. doi: 10.1016/S0896-6273(03)00467-7 PMID: 12925283

6. Anders S, Lotze M, Erb M, Grodd W, Birbaumer N. Brain activity underlying emotional valence and arousal: a response-related fMRI study. Human Brain Mapping. 2004; 23(4): 200–209. doi: 10.1002/hbm.20048 PMID: 15449355
7. Nielen MM, Heslenfeld DJ, Heinen K, Van Strien JW, Witter MP, Jonker C, et al. Distinct brain systems underlie the processing of valence and arousal of affective pictures. Brain and Cognition. 2009; 71(3): 387–396. doi: 10.1016/j.bandc.2009.05.007 PMID: 19665830
8. Posner J, Russell J, Gerber A, Gorman D, Colibazzi T, Yu S, et al. The neurophysiological bases of emotion: an fMRI study of the affective circumplex using emotion-denoting words. Human Brain Mapping. 2009; 30(3): 883–895. doi: 10.1002/hbm.20553 PMID: 18344175
9. Gerber A, Posner J, Gorman D, Colibazzi T, Yu S, Wang Z, et al. An affective circumplex model of neural systems subserving valence, arousal, and cognitive overlay during the appraisal of emotional faces. Neuropsychologia. 2008; 46(8): 2129–2139. doi: 10.1016/j.neuropsychologia.2008.02.032 PMID: 18440572
10. Colibazzi T, Posner J, Wang Z, Gorman D, Gerber A, Yu S, et al. Neural systems subserving valence and arousal during the experience of induced emotions. Emotion. 2010; 10(3): 377–389. doi: 10.1037/a0018484 PMID: 20515226
11. Pessoa L. On the relationship between emotion and cognition. Nature Reviews Neuroscience. 2008 9 (2): 148–158. doi: 10.1038/nrn2317 PMID: 18209732
12. Shu F, Tan AH. A biologically-inspired affective model based on cognitive situational appraisal. Neural Networks (IJCNN), The 2012 International Joint Conference. 2012 Jun 10–15.
13. Novak D, Mihelj M, Munih M. A survey of methods for data fusion and system adaptation using autonomic nervous system responses in physiological computing. Interacting with Computers. 2012; 24(3): 154–172. doi: 10.1016/j.intcom.2012.04.003
14. Badesa FJ, Morales R, Garcia-Aracil N, Sabater JM, Casals A, Zollo L. Auto-adaptive robot-aided therapy using machine learning techniques. Computer Methods and Programs in Biomedicine. 2014; 116 (2): 123–130. doi: 10.1016/j.cmpb.2013.09.011 PMID: 24199656
15. Cano-Izquierdo JM, Ibarrola J, Almonacid M. Improving Motor Imagery Classification with a new BCI Design using Neuro-Fuzzy SdFasArt. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2012; 10(1): 2–7. doi: 10.1109/TNSRE.2011.2169991
16. Carpenter G, Grossberg S, Markuzon N, Reynolds N, Rosen D. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on Neural Networks. 1992; 3(4): 698–713. doi: 10.1109/72.159059 PMID: 18276469
17. Cano-Izquierdo JM, Dimitriadis YA, Gómez-Sánchez E, López-Coronado J. Learning from noisy information in FasArt and FasBack neuro-fuzzy systems. Neural Networks. 2001; 14(4–5): 407–425. doi: 10.1016/S0893-6080(01)00031-4 PMID: 11411629
18. Toledo-Moreo R, Pinzolas-Prado M, Cano-Izquierdo JM. Maneuver prediction for road vehicles based on a neuro-fuzzy architecture with a lowcost navigation unit. IEEE Transactions on Intelligent Transportation Systems. 2010; 11(2): 498–504. doi: 10.1109/TITS.2009.2039011
19. Cano-Izquierdo JM, Almonacid M, Pinzolas M, Ibarrola J. dFasArt: Dynamic neural processing in FasArt model. Neural Networks. 2009; 22(4): 479–487. doi: 10.1016/j.neunet.2008.09.018 PMID: 19128936

frontiers in Aging Neuroscience

ORIGINAL RESEARCH
published: 26 August 2016
doi: 10.3389/fnagi.2016.00205

A Comparative Analysis of 2D and 3D Tasks for Virtual Reality Therapies Based on Robotic-Assisted Neurorehabilitation for Post-stroke Patients

*Luis D. Lledó *, Jorge A. Díez, Arturo Bertomeu-Motos, Santiago Ezquerro, Francisco J. Badesa, José M. Sabater-Navarro and Nicolás García-Aracil*

Biomedical Neuroengineering Group, Miguel Hernández University of Elche, Elche, Spain

Post-stroke neurorehabilitation based on virtual therapies are performed completing repetitive exercises shown in visual electronic devices, whose content represents imaginary or daily life tasks. Currently, there are two ways of visualization of these task. 3D virtual environments are used to get a three dimensional space that represents the real world with a high level of detail, whose realism is determinated by the resolucion and fidelity of the objects of the task. Furthermore, 2D virtual environments are used to represent the tasks with a low degree of realism using techniques of bidimensional graphics. However, the type of visualization can influence the quality of perception of the task, affecting the patient's sensorimotor performance. The purpose of this paper was to evaluate if there were differences in patterns of kinematic movements when post-stroke patients performed a reach task viewing a virtual therapeutic game with two different type of visualization of virtual environment: 2D and 3D. Nine post-stroke patients have participated in the study receiving a virtual therapy assisted by PUPArm rehabilitation robot. Horizontal movements of the upper limb were performed to complete the aim of the tasks, which consist in reaching peripheral or perspective targets depending on the virtual environment shown. Various parameter types such as the maximum speed, reaction time, path length, or initial movement are analyzed from the data acquired objectively by the robotic device to evaluate the influence of the task visualization. At the end of the study, a usability survey was provided to each patient to analysis his/her satisfaction level. For all patients, the movement trajectories were enhanced when they completed the therapy. This fact suggests that patient's motor recovery was increased. Despite of the similarity in majority of the kinematic parameters, differences in reaction time and path length were higher using the 3D task. Regarding the success rates were very similar. In conclusion, the using of 2D environments in virtual therapy may be a more appropriate and comfortable way to perform tasks for upper limb rehabilitation of post-stroke patients, in terms of accuracy in order to effectuate optimal kinematic trajectories.

Keywords: virtual reality, rehabilitation robotics, post-stroke, sensorimotor function, upper extremity

OPEN ACCESS

Edited by:
Rodrigo Orlando Kuljiš, University of Miami School of Medicine, USA

Reviewed by:
Adrian Burlacu, Gheorghe Asachi Technical University of Iaşi, Romania
Francesca Morganti, University of Bergamo, Italy

****Correspondence:***
Luis D. Lledó
lledo@umh.es

Received: *23 February 2016*
Accepted: *11 August 2016*
Published: *26 August 2016*

Citation:
Lledó LD, Díez JA, Bertomeu-Motos A, Ezquerro S, Badesa FJ, Sabater-Navarro JM and García-Aracil N (2016) A Comparative Analysis of 2D and 3D Tasks for Virtual Reality Therapies Based on Robotic-Assisted Neurorehabilitation for Post-stroke Patients. Front. Aging Neurosci. 8:205. doi: 10.3389/fnagi.2016.00205

1. INTRODUCTION

Virtual Reality (VR) is a technology platform that allows developing computer generated environments which the subjects can explore and interact with any type of object or events to perform perspectives and motor tasks. VR gives an accurate way to control all the elements of a scene and the objectives, adjusting each task to a specific user. The main feature that the VR provides is the possibility of repeating the same task in any moment, modifying factors such as level of complexity, time and intensity of the practice. In this way, the virtual therapy may be used to promote motor learning and rehabilitation due to the VR can be adjusted to generate environment, scenario, or activity that allows for the user practice motor skills to improve the experience-dependent neural plasticity (Doyon and Benali, 2005). The possibility of modifying factors such as the repetition, intensity, time, and specificity of the activities of the virtual therapies is beneficial for this type of neural recovery (Kleim and Jones, 2008). In recent years, some scientific and clinical trials have demonstrated the effectiveness of VR as an intervention tool for the rehabilitation of different injuries with specific neurological conditions (Burdea, 2002; Crosbie et al., 2007). However, a control device to interact with virtual activities is required, depending of the limb affected by the disease. There is a wide panorama on rehabilitation systems for upper limb that use robotic technology including virtual reality visualization (Maciejasz et al., 2014). In some studies, repetitive movements guiaded by robotic devices and directed by virtual reality improve the motor control in patients with upper limb injuries (Merians et al., 2006). Beside this, there are some clinical studies about the development of VR systems to deliver rehabilitation therapies for motor recovery of hand function (Jack et al., 2001) or to improve the performance of activities of daily living in post-stroke patients (Laver et al., 2012; Turolla et al., 2013). Furthermore, a navigation environment in three dimensions (3D) has been implemented to explore the influence on aging in the episodic memory (Jebara et al., 2014). In Fluet and Deutsch (2013), an overview of virtual reality studies for sensorimotor rehabilitation post-stroke has been performed to evaluate a comparative efficacy between VR and standard of care and/or differences in VR delivery methods, using different categories.

Several studies suggest that the robotic technology can be used to improve the quality and the evaluation in the neurological rehabilitation (Garcia et al., 2011), enhancing the productivity and reducing costs in that field. Recent developments in robotic technology can help to perform a most objective and reliable analysis of the therapies that are applied to the patients with neurological injuries (Badesa et al., 2012, 2014a,c). That is because this type of devices are able to record kinematic and kinetic data. From this data, useful markers can be extracted to quantify the motor recovery process during the therapy (Volpe et al., 2009; Einav et al., 2011; Bertomeu-Motos et al., 2015; Papaleo et al., 2015). Recently in Norouzi-Gheidari et al. (2012), it is shown that the rehabilitation sessions performed with the robotic device get better recovery outcomes than the conventional therapy during the rehabilitation of the upper limb of stroke patients. For these reasons, the rehabilitation with robotic devices can provide an enhancement in the quality of patient's life, giving them most independence in the daily life activities (Pollock et al., 2014).

The use of more complex and realistic VR systems in the neurorehabilitation therapies assisted by robotic device is increasing. The combination of robotic systems for neuromotor rehabilitation and virtual reality takes advantages of both techniques such as: to increase the patient's motivation; to enhance the variability and adaptability; transparent storage of the data provided by the robotic system and the VR system separately; online recording of the data for remote verification; possibility to replicate any environment of the daily life without having the physical. With this methodology, a more effective therapeutic treatment and a better recovery of the patient is accomplished (González et al., 2015).

There are a two important issues concerning the virtual reality: one is related to how the virtual environment may be perceived by the user using different visualization platforms, and the other one is related to graphic content. Regarding the first appointment, different visualization platforms exist such as computer monitors, head-mounted-displays (HMDs) or large screen-projection-systems (SPS). Each platform has a particular way to apply the virtual therapies taking into account therapeutic goals and may provide different benefits that are suitable for the patient's needs. In Rand et al. (2005), the effects of viewing the same virtual environment through a HMD (3D platform) and a computer monitor (2D platform) have been compared in young and older subjects. Conversely, a 3D virtual enviroment shown through a HMD and a SPS (2D platform) have been analyzed by Subramanian and Levin (2011), evaluating the motor performance with respect to the kinematic movements in healthy and post-stroke subjects. In both studies, better outcomes have been obtained when the virtual environment was shown in the 2D platform visualization, in a computer monitor and a SPS respectively. However, this studies have focused in the visualization platform and the same environments have been presented respectively in the experiments without taking account the type of graphic content that are shown (2D or 3D graphics).

Regarding the second issue appointed above about the graphic content, there are studies about VR systems with environments based on 2D graphics and others in 3D graphics. In García-Betances et al. (2015) an overview of recent VR technology for Alzheimer's disease applications has performed, and these systems use conventional 2D graphics display or 3D graphics indistinctly. Similarly occurs with the brain damage rehabilitation in Rose et al. (2005), post-stroke studies such as Merians et al. (2006), Saposnik (2016), Henderson et al. (2007), Mottura et al. (2015). Therefore, there is a wide panorama on virtual rehabilitation in the scientific literature. However, an objective comparison about how affects the visualization of 2D graphics display and 3D virtual environment to the motion perception in post-stroke subjects have not been addressed yet. That means, there is no evidence that shows if it is better or not to perform virtual rehabilitation tasks produced by 2D or 3D graphics. The visual perception of the virtual objects can be incremented using 3D graphics, in such a way that tasks based in the daily life designs are more similar to the reality. While a

2D graphics allow a more simple representation of the tasks. The two perspectives must be tested to evaluate what kind of visual representation provides better quality of motor performance in terms of movement kinematics. This evaluation can be carried out when the subject performs the same movement to complete the targets in both types of visualization. Therefore, the robotic devices can be used to restrict this movement and extract objectively quantitative data. This way, the neuro-rehabilitation therapies can be adapted to each patient (Morales et al., 2014; Lledó et al., 2015a).

In this study, the effects of applying therapeutic games in two or three dimensions in the virtual therapies assisted by a robotic device are evaluated and their outcomes are compared. In this way, quantitative data is provided to evaluate the influence of the virtual therapy and to asses what kind of virtual environment is adjusted better to each patient in terms of usability, confidence, and comfort. Therefore, the main objective of this study was to determine if there are differences in the movement kinematics parameters recorded by the robotic device that assess the patient's motor performance in 2D and 3D virtual tasks. To do this, two visual tasks have been designed modifying the immersion level using graphics in two and three dimensions, but the kinematic target of the two visual tasks was remained.

2. MATERIALS AND METHODS

2.1. Patients

The study has been performed in a hospital of attention to chronic patients and long-stay. The experiment protocol of the proposal study was approved by the Medical Ethics Committee. The medical team has been responsible for including patients who are receiving physiotherapy and occupational therapy treatment. All patients have been informed properly by the medical staff and they gave written consent before starting the study, indicating that they understood the purpose and requirements of the study.

The inclusion criteria were: adults with hemiparesis/hemiplegia secondary to stroke in subacute phase (between 1 and 6 months after the injury). The criteria with respect of the muscular conditions of the upper-limb were (i)muscular tone with punctuation below 2 in the Modified Ashwoth Scale (Bohannon and Smith, 1987), (ii)muscular balance in shoulder abduction and elbow flexion on the basis of the Motor Index ≥ 2 (Collin and Wade, 1990). In the selection process, the inclusion of patients with the following injures was avoided: painful shoulder, apraxia, uncontrolled trunk in seating system, diagnostics with hand effects (as arthritis or other rheumatologic diseases), severe perceptual deficits, stroke of posterior circulation (vertebrobasilar system), linguistic deficits that prevent useful communication. The patients have to be oriented to the three spheres (social, temporal, and spatial), with capacity of collaboration and understanding the tasks instructions and all relevant information from the study. After selection process, nine post-stroke patients (age 40–70 years of both genders) have taken part in the study. The pathologies diagnosed are varied such as pontine and artery cerebral middle infarction, ischemic myelitis, glioblastoma hematoma, parietal and basal ganglia hemorrhage, with a type of diagnosys ischemic, hemorrhagic or myelopathy and left or right laterality. The main characteristics of the study participants and their clinical diagnostic are shown in **Table 1**.

2.2. Neurorehabilitation System

The neurorehabilitation system used to perform the motor therapy and get all the objective information about the proposed study is formed by a PUPArm robot system (Badesa et al., 2014c) and a visualization subsystem. This system was designed and developed by Biomedical Neuroengineering Group at Miguel Hernandez University of Elche as a rehabilitation robot for patients with stroke or other neurological disorders. The neurorehabilitation system is shown in **Figure 1**.

The robotic mechanism consists of four metallic bars, similar to the MIT-MANUS rehabilitation robot (Krebs et al., 1998). These bars are connected as a parallelogram and are driven by pneumatic swivel modules. This structure provides a planar two-dimensional manipulator with two degrees of freedom. The manipulator remains fixed to a table. Consequently, the horizontal movement to upper limb of the subjects is permitted by the system, involving flexion and extension of the elbow and shoulder, and horizontal abduction and adduction. On the other hand, the visualization subsystem is composed of a monitor computer with a custom developed software called REVIRE which is used as VR simulation system to display activities in coordination with the robot's movements. A computer is responsible to coordinate in time-real the pneumatic actuators, the targets of the tasks and the feedback to the user. The system is capable to record information about the patient's progress in rehabilitation, based in parameters such as position, velocity and forces. All data is registered by robot sensors. These data are processed to provide an objective assessment to the therapist.

2.3. Virtual Tasks

The virtual task with the 2D environment consists on a roulette formed by a central target and eight peripheral targets. These targets were circles with a 1 cm of radius. The eight peripheral targets were distributed uniformly on the circumference of the circle and placed 10 cm from the center target. The main purpose of this task was to reach one of the eight peripheral targets from central target by controlling the robot end-effector attached to the subject's hand. The next target remain illuminated. To do that, the therapy task is displayed on the monitor of the neurorehabilitation system with a visual-guided reinforcement represented by a white circle of 1cm of radius which indicate the current position of the robot end-effector. A screenshot with the 2D environment of the virtual task and its structural information is shown in **Figure 2**. This virtual task and the neurorehabilitation system have been used in our previous studies about the age influence in the sensorimotor function of the upper limb (LLinares et al., 2013) or special cases of neurological disorder (Badesa et al., 2014b).

To complete this comparative study, a 3D virtual task has been designed following the same target criteria used in the 2D roulette in order to perform the same type of movements. The 3D virtual task has been developed using the implementation

TABLE 1 | Clinical characteristics of the study participants.

Patient	Sex	Age (years)	Diagnosis	Diagnostic type	Location	Laterality
1	Female	69	Ischemic myelitis	Myelopathy	Medullary	Tetraparesia
2	Male	40	GGBB hemorrhage	Hemorrhagic	Basal ganglia	Right
3	Male	41	GB Hematoma	Hemorrhagic	Basal ganglia	Left
4	Male	46	Undetermined ACM infarct	Ischemic	Parietal	Left
5	Female	66	Pontine infarction	Ischemic	Brainstem	Left
6	Female	41	Parietal hemorrhage	Hemorrhagic	Parietal	Right
7	Male	53	Undetermined ACM stroke	Ischemic	Parietal	Right
8	Female	41	Cerebral hemorrhage	Hemorrhagic	Frontal	Left
9	Female	46	Cerebral hemorrhage	Hemorrhagic	Frontal	Left

ACM, Artery Cerebral Middle; GB, Glioblastoma; GGBB, Basal Ganglia.

FIGURE 1 | Neurorehabilitation system based in PUPArm robot.

pattern of virtual simulators explained in Lledó et al. (2015b). Ogre3D (Steve, 2013) is used as engine of graphical rendering and the physic engine NVIDIA PhysX Nvidia (2011) applies an extra degree of realism to the collision between elements of the 3D environment, whose graphical meshes are designed with the modeling tool called Blender. The environment of the 3D virtual task simulates a box factory with perspective viewing where the scene converges to the central point of the screen. The graphical scenario consists of eight platforms and a central deposit, that are equivalent to the eight peripheral targets and the center target of the 2D roulette. The eight platforms are placed uniformly around the central deposit. To indicate the user the next target position, a box with dynamic behavior is placed in a random platform. In this case, the user controls the robotic end-effector to manage a virtual wrench with kinematic behavior in order to pick the target boxes and drop this box in the central deposit. **Figure 3** shows a screenshot with the 3D environment of the virtual task with structural information. Basically, the purpose of this task is the same than the 2D task, but with a different visualization level. The workflow to comply the virtual 3D task is:

1. First, the user should approximate the virtual tool to the central deposit to initiate the visualization of perspective target positions.
2. A box appears randomly in any of the eight platforms. The platform is illuminated as visual support, and the virtual wrench is dynamically oriented to the positional target.
3. The user has a limited time to pick the target box. This limited time is shown in a progress bar placed in the up left side of the screen. If the target has not selected, the box disappears and the next target is executed.
4. When the virtual wrench is near to the target, the box is caught. Then the user have to bring the tool to the central deposit to release the box. There is a sound support to indicate that the target is completed.

The functionality and structure of the both virtual tasks are the same. This is necessary to an objective comparison of the parameter values obtained by the robotic device. A structural correlation between 2D and 3D tasks is presented in **Figure 4**. This approach compares the same situation with different external stimulus. The purpose of develop a scenario in three dimensions is provide a correlation more natural between the movement of the robot and the view of the user. In 2D tasks, when the user approaches or moves away the end-effector, the controllable element in the task is moved up or down of the screen. Therefore, many patients tried to force the efector upwards or downwards to put the controllable element in its corresponding place in screen. Associating the horizontal planar movement of the robotic device with the vertical movement in the 2D task displayed in screen, can cause confusion in some patients. In this way, the 3D task replicate in the screen the same type of movement of the robot.

2.4. Setup and Protocol

Over a 2 months period, the study group has received the therapy treatment assisted by the PUPArm robot with four weekly sessions of 10 min, 36 sessions in total. In the first session, a general evaluation of patient collocation is carried out to get parameters such as the height of the screen or chair, and their mobility range. After, these values are used during the tasks. In this way, the maximum functional range of the specific patient is achieved. Before starting the session, the patient is placed in front of the robotic device in a comfortable position using the parameters obtained in the first evaluation sesion. The monitor

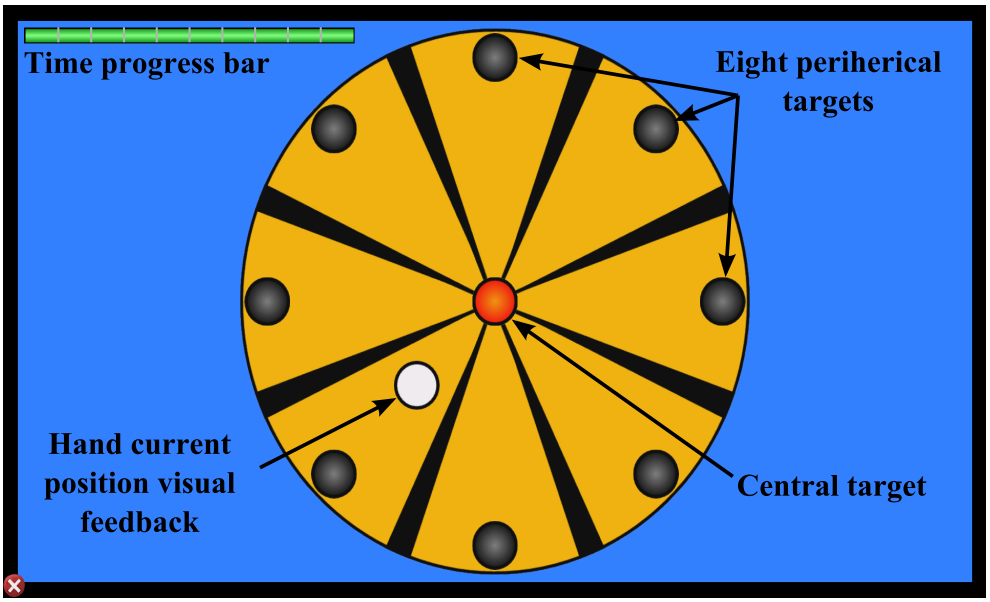


FIGURE 2 | Targets and visual feedback of the 2D task.

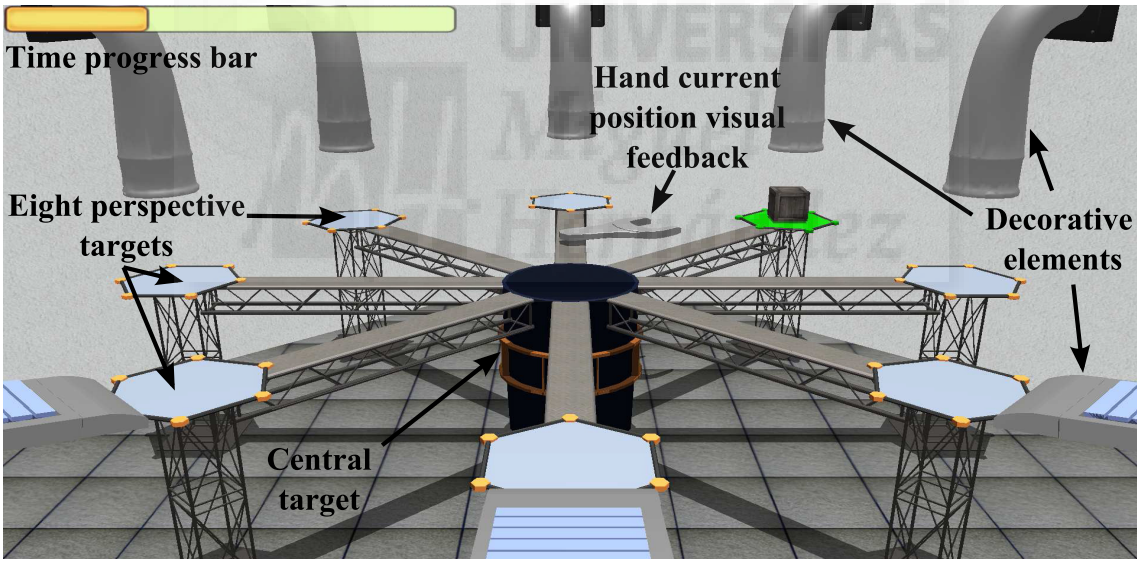


FIGURE 3 | Targets and visual feedback of the 3D task.

that offers the visual feedback is located to 70 cm from the patient. Each session is structured in two blocks of movement training, depending on the virtual task. Between each block, the patients had 3-min rest periods. The session time is organized as follow:

- In the first block of movements, one of the roulette task is choosen randomly. Then, the patient has to perform 32 trials focusing on this selected 3D or 2D task with global movements, both shoulder and elbow. Approximately, this block is completed in 4–5 min.
- Once the 32 trials of the same task have been completed, the patient have a 3-min rest period.
- To finish the session, the second block of movements is performed completing 32 trials with the other roulette task. As with in the first block, the patient have to carry out the same global movements and the time taken to complete these trials are approximately 4–5 min too.

The subjects had to reach one of the eight possible targets and return toward the central target in order to complete one trial. **Figure 5** shows schematically the workflow to perform one

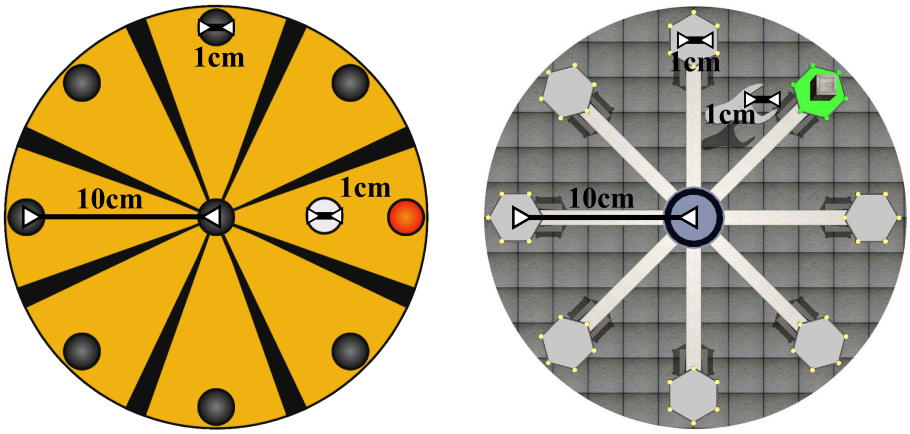


FIGURE 4 | Structural correlation between 2D Roulette and 3D Roulette (zenithal view).

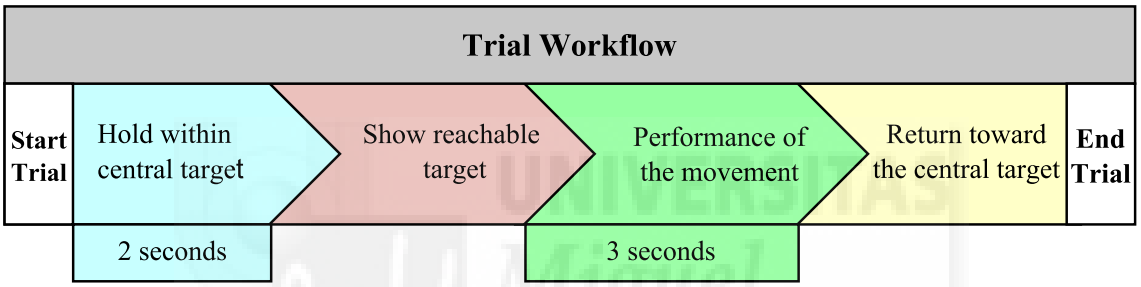


FIGURE 5 | Workflow to complete one trial.

trial. To begin a trial, the subjects had to hold the controllable element through of robotic end-effector for 2 s within the central target. Then, one reachable target was illuminated to indicate the next position where the patient had to place the end-effector. To complete the movement, a limited time of 3 s was given. When the subject reached the target, had to return to the central target without time limit. Therefore, this movement sequence is performed in the 32 planned trials, pointing targets randomly. This protocol was the same to the two virtual tasks, but the elements inside the tasks were different. The following parameters have been calculated from the data recorded by the robotic device. A brief explanation of these parameters (a complete explanation of these parameter can be found in LLinares et al., 2013) are described:

1. Maximum speed: The maximum speed reached by the arm movement.
2. Reaction time: The time elapsed from the indication of the random target and the onset of the arm movement.
3. Path length: The total distance traveled to reach one target.
4. Initial movement: The distance traveled during the initial movement whose trajectory has a deviation with respect the reach target, until this deviation is corrected.
5. Initial movement ratio: Relation between the initial movement and the path length.
6. Initial movement direction error: The angular deviation in degrees, between the optimal path established by the line from the central target to the reach target, and the vector generated by the initial movement.
7. Time: The total time needed to reach one target.
8. Succes rate: The percentage of the trials that have been completed correctly.

Additionally, a study of the system usability has been defined in this work in order to get a quantitative indicator of usability that measures the compliance level front the user expectations, satisfaction level and the performance of the neurorehabilitation system. A type System Usability Scale (SUS) survey Brooke (1996) has been used. A survey has been delivered to each patient after testing the system. This survey had ten questions giving a global view of subjective assessments of usability. Six of these questions had positive character and the remaining questions were negatives to contrast answers. The users had to reply their agree or disagree level regarding the system using a Likert scale (from 1 to 5; Likert, 1974).

3. RESULTS

The objective data have been acquired during 36 sessions of the virtual therapy treatment assisted by the PUPArm robot and the

main descriptive statistics for each patient have been collected in **Table 2**. The outcomes are presented in a ten column table to show the kinematics parameter values that assess the quality of patient's motor performance. Each parameter contains two possible values depending on the visualization level of the task provided to the patient. In this case, one value is extracted from 2D Roulette and the other from 3D Roulette to compare the numerical outcomes. The success rate to complete the reaching target can be observed in the last column. The high success rate between 95.10 and 100% indicated that the system was ease of use without complication to perform the virtual tasks. There were no significant differences in the success rate between the two types of visualization.

A comparative analysis of these parameters has been assessed with both values extracted from 2D and 3D tasks in order to get the patient's performance variation of 3D parameters with respect to 2D parameters. Equation (1) has been used to calculate the percentages of this variation. This equation is applied to each kinematic parameters for all the patients. Therefore, data3D and data2D generalize the numeric value of these kinematic parameters. As each parameter has got associated 2 possible values, the data3D variable records the value of the 3D task, while data2D variable contains the value of the 2D task. The percentages values are gathered in **Table 3** including the mean, standard deviation, median and the maximum-minimum value of each parameter of all the data. The positive values indicate the increased percentage that the parameter extracted from the 3D task is higher than the same parameter in 2D task.

$$\left(\frac{data3D - data2D}{data2D}\right) * 100 \quad (1)$$

These parameters are plotted as box plots in **Figure 6** to provide a general vision of the data distribution. On a box plot, the boxes are divided by a horizontal segment that indicates the position of the median value. Therefore, the relation between this value and the 25th and 75th percentiles, represented by the bottom and top of the box, can be observed. The boxes are located on a segment whose extremes the minimum and maximum values of the parameter. In this box plots, the outliers values have been marked with the "+" symbol and an asymmetric distribution appears in the boxes.

To evaluate the sensorimotor function after the virtual therapy based in robotic-assisted neurorehabilitation, the movement trajectories performed by one subject in the first and the last session are shown in **Figure 7**. The left side of this figure corresponds to trajectories during the visualization of the 2D Roulette, while the right side was carried out with the 3D Roulette. In both tasks, they were more erratic trajectories in the first session when the subject attempted to reach the

TABLE 2 | Data acquired by the robotic device.

Patient	Task	Maximum speed	Reaction time	Path length	Initial movement	Initial mov. ratio	Initial mov. direction error	Time	Success
1	2D	104.13	0.69	104.27	72.45	0.70	1.11	7.11	99.61
	3D	115.56	0.89	114.41	83.01	0.74	1.13	8.18	99.61
2	2D	57.07	0.65	114.55	41.60	0.38	3.21	10.35	100
	3D	58.64	0.71	121.53	42.01	0.36	3.17	11.38	98.64
3	2D	91.81	0.85	116.94	58.38	0.52	2.05	11.81	100
	3D	92.58	1.09	119.06	60.60	0.53	1.87	13.59	100
4	2D	118.30	0.71	123.82	69.13	0.61	1.57	13.45	99.67
	3D	134.66	0.89	149.68	78.62	0.60	1.67	16.26	98.58
5	2D	153.19	0.88	197.90	95.59	0.61	1.71	15.41	98.83
	3D	153.40	1.04	250.27	97.45	0.51	2.53	27.04	95.10
6	2D	45.94	0.40	110.07	33.96	0.32	3.56	12.13	98.83
	3D	46.57	0.48	111.14	32.89	0.31	3.77	13.48	97.01
7	2D	63.24	0.64	120.42	42.98	0.37	3.19	16.07	97.13
	3D	60.49	0.77	121.61	41.68	0.36	3.17	15.99	95.53
8	2D	110.09	0.52	105.96	69.78	0.68	1.34	8.03	98.96
	3D	113.72	0.71	115.34	74.28	0.67	1.37	11.76	98.96
9	2D	112.37	0.73	130.40	75.48	0.64	1.73	12.36	100
	3D	121.37	1.05	157.78	83.21	0.61	1.88	13.33	98.27

Measurement units: Maximum Speed, mm/second; Reaction Time, seconds; Path Length, mm; Initial Movement, mm; Initial Movement Ratio, dimensionless; Initial Movement Direction Error, degrees; Time, seconds; Success, %.

TABLE 3 | Variation 3D parameters with respect to 2D parameters in %.

Patient	Maximum speed	Reaction time	Path length	Initial movement	Initial mov. ratio	Initial mov. direction error	Time	Success
1	10.97	29.27	9.73	14.57	5.79	1.55	15.13	0
2	2.75	9.35	6.09	0.99	−4.69	−2.14	10	−1.36
3	0.83	27.80	1.80	3.80	2.14	−8.99	15.12	0
4	13.83	25.26	20.89	13.72	−3.07	6.74	20.89	−1.09
5	0.13	18.92	26.46	1.94	−17.28	47.90	75.47	−3.78
6	1.36	20.11	0.97	−3.09	−3.79	5.99	11.12	−1.84
7	−4.35	10.99	0.98	−3.03	−3.06	−0.49	−0.47	−1.64
8	3.29	37.77	8.85	6.45	−1.16	2.18	46.58	0
9	8	42.93	20.99	10.24	−4.77	8.99	7.83	−1.73
MEAN	4.09	24.71	10.75	5.07	−3.32	6.86	22.41	−1.27
STD	5.42	10.60	9.14	6.27	5.93	15.37	22.42	1.21
MEDIAN	2.75	25.26	8.85	3.80	−3.07	2.18	15.12	−1.36
MAX	13.83	42.93	26.46	14.57	5.79	47.90	75.47	0
MIN	−4.35	9.35	0.97	−3.09	−17.28	−8.99	−0.47	−3.78

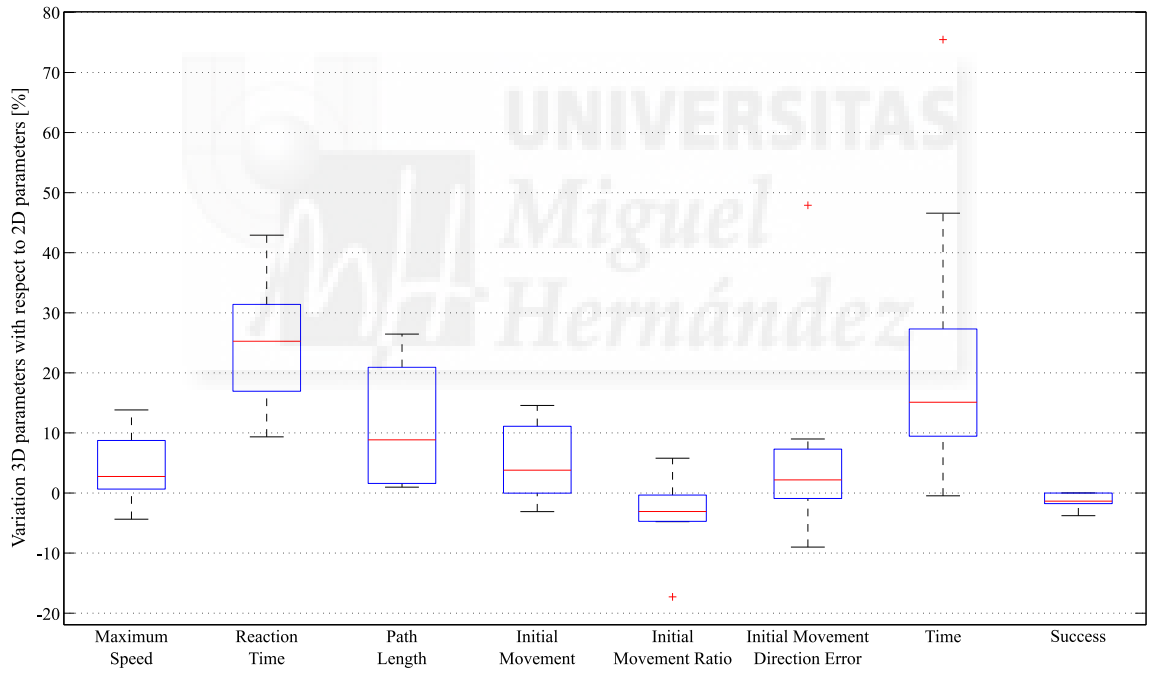


FIGURE 6 | Statistical analysis of the data acquired by the robotic device, represented in box plots.

reachable targets. However, the patients performed more correct trajectories when they used the task with 2D environment. The trajectories presented a more irregular paths in 3D task. Furthermore, less deviations of trajectory are performed when the patients had to reach the central target in the 2D task. Regarding the first session, in the 3D Roulette the trajectories between the reached target and the central target presented a more length than the 2D task and the deviation error and the time to reach the targets is higher (see **Table 3** and **Figure 7**). Therefore, the patients present more difficults to achieve the targets in the 3D task when they started the therapy, using the proposal system. This may mean that the usage of tasks displayed with 2D graphics, hence with less level of detail, is easier to perceive and is better adapted to users who have not used this type of system. In **Figure 7** an improvement in the control of the system by the patient can be observed. This fact suggests that the sensorimotor performance of the patient is increased due to the repetition

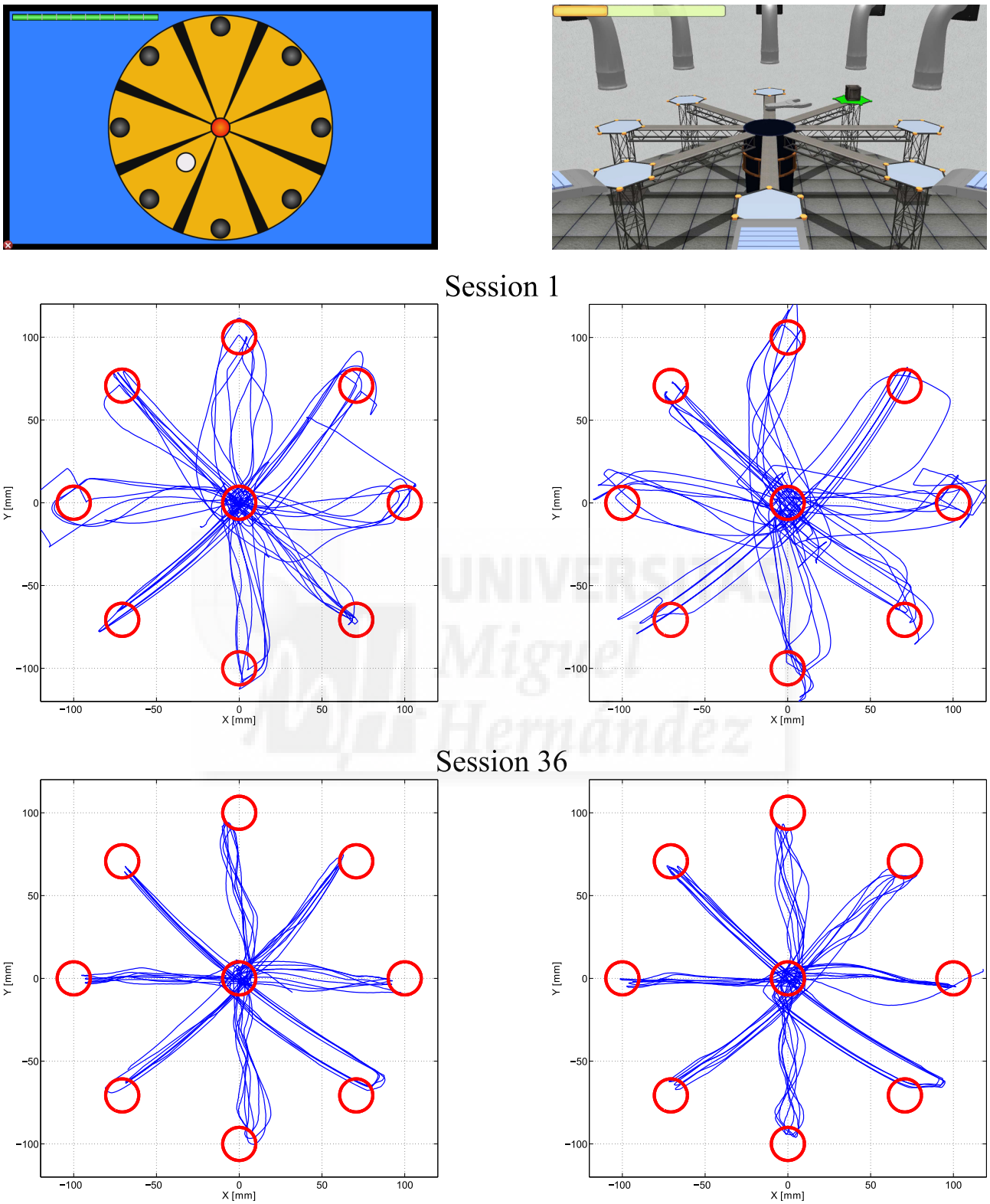


FIGURE 7 | Movement trajectories to reach the targets by one patient. Sensorimotor function assessment for two tasks. In the left are shown trajectories performed in 2D task during the first and the last session. In the right side are shown the trajectories performed in 3D task.

of the movements with the arm limb for practice motor skills.

A statistical analysis of these quantitative elements has been performed to verify the bivariate correlation of the kinematic and kinetic parameters between them, in order to find out the level and direction of the correlation. In this way, the type of visualization that generate better relations between kinematic parameters can be analyzed, and consequently, a better assessment of the results can be achieved. This analysis is performed from calculating the Pearson correlation coefficient and the level of significance to indicate if there is a correlation between each pair of the study parameters. **Table 4** shows the correlation matrix with the Pearson correlation coefficient and the level of significance of each pair of parameters assigned to record the data obtained in the 2D visualization. Meanwhile, **Table 5** records the correlation matrix with the data generated by the 3D task.

A usability measure of the neurorehabilitation system has been calculated when all patients have completed their corresponding SUS surveys. **Table 6** shows the statements that cover a variety of aspects of system usability, such as need for support, training and complexity. Also, the table gather the outcomes provided by each user on a 5 point scale ranging from "strongly agree" to "strongly disagree." The average of the patient assessment of the aspects dealt in the survey are collected in **Figure 8**. In this radial graph, the pointed line represents the optimal response for getting the highest subjective level of usability. The continued line is the response average of the nine patients. Generally, the response of the patients was widely similar to the optimal responses, except for the question about the needed for technical support. This mean a high usability rate.

For getting an usability interpretation or punctuation represented by a percentage indicator, the numerical contribution of each point are added depending on the question. Each item's score contribution has a range from 0 to 4, as can be observed in **Figure 9**. Then, the result of the sum are multiplied by 2.5 to get the SUS scoring (Brooke, 1996). The total SUS score of the survey was 76.11%. It is a very positive indicator that reflects a high satisfaction and engagement of the assessed patients. A percentage more high means a better usability level.

4. DISCUSSION

The influence of applying 2D or 3D therapeutic games in the performance of upper limb rehabilitation in post-stroke patients has been presented in this study. Currently, this type of study have not been addressed yet in the scientific literature or the issues dealt in this field are discussed by subjective assessments. For these reason, a quantified correlation between upper-limb motor function and the visualization of the reaching task was assessed in an objective way computing kinematic parameters provided by PUPArm robotic device. At first glance, the analysis of the kinematic parameters with two types of visualization has provided very similar results comparing nominal values. However, some little differences in sensorimotor performance have been found depending upon the visualization of the task based on peripheral or perspective targets, viewing these values and performing an analysis of the correlation between kinematic parameters. Each patient achieved similar outcomes when performing both task with 2D and 3D environments in all sessions during the therapy. However, some patients obtained better results than others showing a variation in the sensorimotor abilities. These changes may be due to age, motor damage or level of cerebral injury that affect to the efficiency of cognitive and physiological processes.

Perform straight trajectories to reach the targets was an important purpose of the tasks. In the first sessions, the most

TABLE 4 | Correlation matrix of each pair of parameters assigned for the data obtained in the 2D visualization.

		Maximum speed	Reaction time	Path length	Initial movement	Initial mov. direction error	Time
Maximum	R	1	0.645	0.654	0.986**	−0.842**	0.075
speed	Sig.	-	0.060	0.056	0.000	0.004	0.848
Reaction	R	0.645	1	0.605	0.640	−0.457	0.317
time	Sig.	0.060	-	0.084	0.063	0.217	0.406
Path	R	0.654	0.605	1	0.642	−0.158	0.614
length	Sig.	0.056	0.084	-	0.062	0.685	0.078
Initial	R	0.986**	0.640	0.642	1	−0.852**	0.003
movement	Sig.	0.000	0.063	0.062	-	0.004	0.994
Initial Mov.	R	−0.842**	−0.457	−0.158	−0.852**	1	0.386
direction error	Sig.	0.004	0.217	0.685	0.004	-	0.305
Time	R	0.075	0.317	0.614	0.003	0.386	1
	Sig.	0.848	0.406	0.078	0.994	0.305	-

*R, Pearson correlation; Sig, Level of significance; **The correlation is significative in the level 0.01.*

TABLE 5 | Correlation matrix of each pair of parameters assigned for the data obtained in the 3D visualization.

		Maximum speed	Reaction time	Path length	Initial movement	Initial mov. direction error	Time
Maximum speed	R	1	0.729*	0.684*	0.982**	−0.713*	0.445
	Sig.	-	0.026	0.042	0.000	0.031	0.230
Reaction time	R	0.729*	1	0.530	0.745*	−0.594	0.321
	Sig.	0.026	-	0.142	0.021	0.092	0.400
Path length	R	0.684*	0.530	1	0.649	−0.002	0.899**
	Sig.	0.042	0.142	-	0.059	0.996	0.001
Initial movement	R	0.982**	0.745*	0.649	1	−0.746*	0.360
	Sig.	0.000	0.021	0.059	-	0.021	0.342
Initial mov. direction error	R	−0.713*	−0.594	−0.002	−0.746*	1	0.253
	Sig.	0.031	0.092	0.996	0.021	-	0.512
Time	R	0.445	0.321	0.899**	0.360	0.253	1
	Sig.	0.230	0.400	0.001	0.342	0.512	-

*R, Pearson correlation; Sig, Level of significance. *The correlation is significative in the level 0.05. **The correlation is significative in the level 0.01.*

TABLE 6 | Questions of the survey and the patient's answers.

		N Patient								
N	**Question**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	I think that I would like to use this system frequently	5	5	5	4	5	5	5	5	5
2	I found the system unnecessarily complex	5	5	1	2	1	3	3	1	1
3	I thought the system was easy to use	5	5	5	4	4	3	3	5	5
4	I think that I would need the support of a technical person to be able to use this system	3	3	2	2	1	5	4	3	4
5	I found the various functions in this system were well integrated	4	4	5	4	5	3	3	4	4
6	I thought there was not any inconsistency in this system	5	5	1	2	5	3	5	5	1
7	I would imagine that most people would learn to use this system very quickly	4	4	5	4	5	3	4	3	2
8	I found the system very cumbersome to use	1	1	1	1	1	3	4	1	1
9	I felt very confident using the system	5	5	5	4	5	3	3	5	5
10	I needed to learn a lot of things before I could get going with this system	1	1	1	1	1	3	4	1	1

of the targets in the both tasks were achieved with erratic trajectories. For same peripherals and perspective targets, the trajectories presented different deviations in each trial. This fact happened for all patients. However, as is shown in **Figure 7**, the reachable targets that required diagonal trajectories were reached with better kinematic movements in the 2D task. More rectilinear trajectories were performed. In the 3D tasks, deviant trajectories are observed in all targets. Regarding the central target, the trajectories were more precise with less deviation in the 2D task. The left graph in the **Figure 10** shows the dispersion diagram between the Initial Movement and the Initial Movement Direction Error. These two parameters directly affect the correct performing of the trajectories and were highly significative in the correlation analysis in both visualization types. This correlation was negative and the lineal asociation degree was stronger in the 2D task. This means that the deviation error when the patient start the arm movement is corrected with less initial distance to reach the better direction trajectory to the target with 2D environments. This fact, is because the zenithal view allows to visualize better the path traveled, facilitating to carry out more straight trajectories.

Consequently, the best trajectories performance in the 2D tasks may suggest that the 2D environments are more appropiate scenarios to generate tasks that provide a better sensorimotor control when the patients start to use the system. For these reason: the trajectories were straighter and had less deviations. In the last session, the movement trajectories were corrected significantly to the point to achieve paths almost without deviation. While the targets are reached satisfactorily with stable trajectories. Comparing the outcomes of the first and the last session can be observed that the trajectories improve with the patient's experience. This enhancement indicates a positive assessment in the recovery of the sensorimotor function of the patients, as is happened in LLinares et al. (2013), Badesa et al.

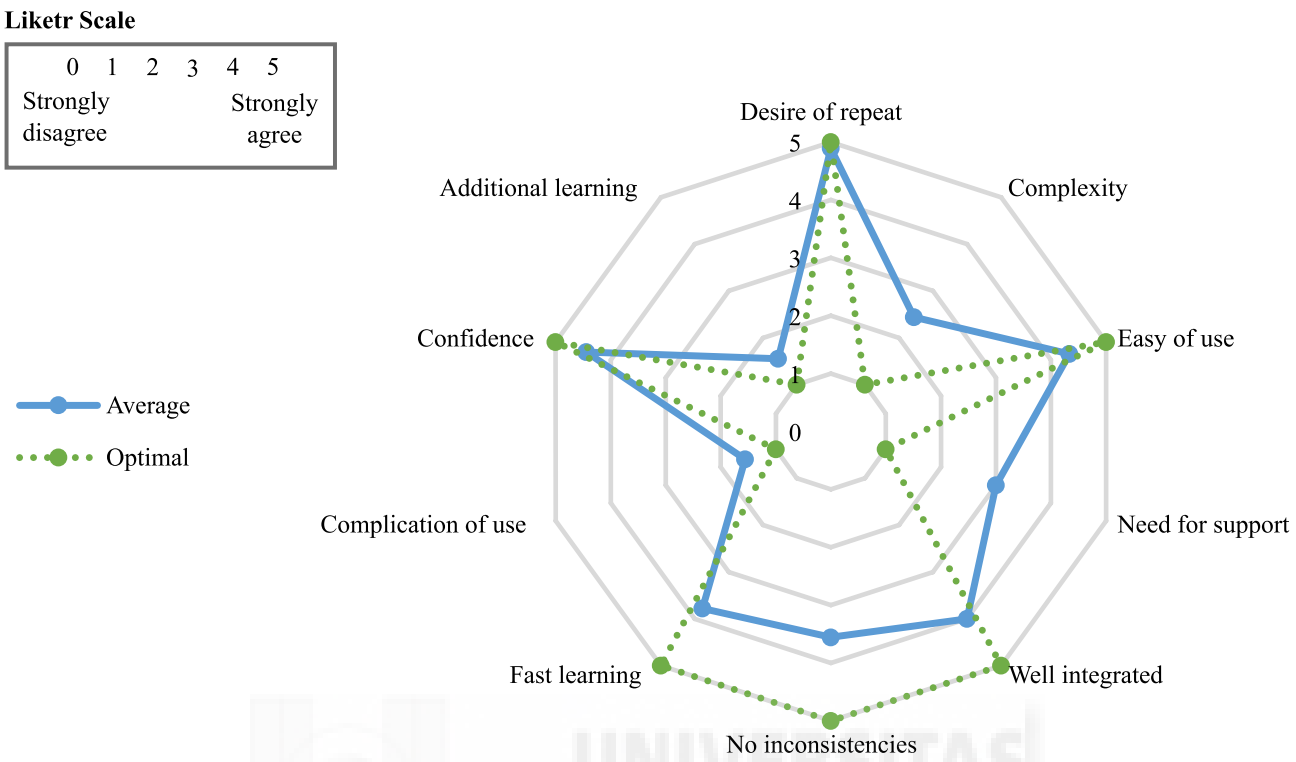


FIGURE 8 | Average of the survey patient responses.

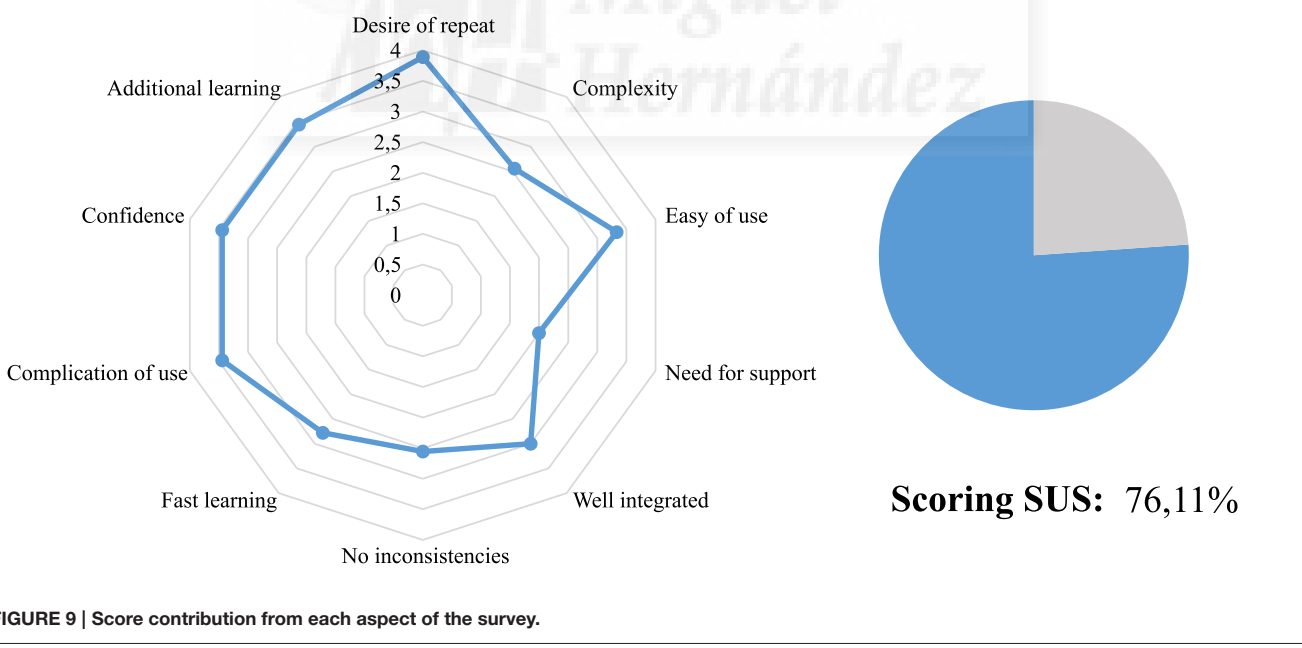


FIGURE 9 | Score contribution from each aspect of the survey.

(2014b). Generally, the success rates in all session during both tasks were quite high. The targets were achieved practically in all trials by the patients. The high values of the success rate insinuate that the tasks were not complex and the targets were recognized clearly.

On the other hand, differences between tasks in terms of kinematic and kinetic parameters can be found in **Tables 2**, **3** and **Figure 6**. For all patients, the time reaction in 3D task was higher than 2D task, which implies that the increasing of the immersion level in the environment provokes unnecessary distractions to

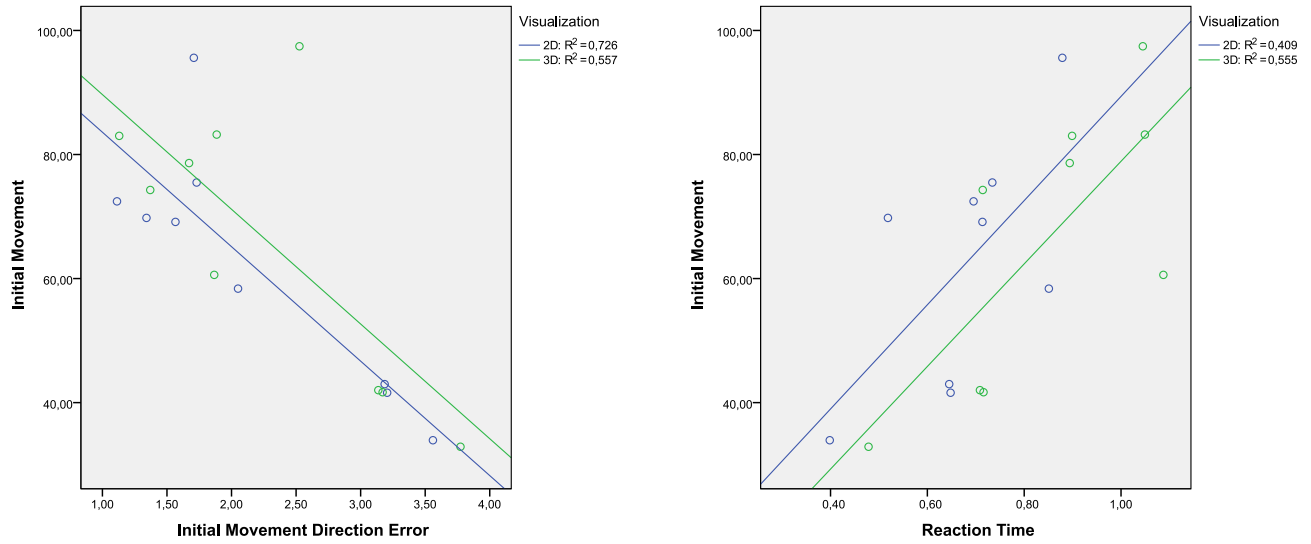


FIGURE 10 | Dispersion diagram with the most significant variables that affect the initial deviation of the trajectories.

the patients. Thus, the patient's concentration level is augmented with less detail level in the virtual environment. Nevertheless, the correlation between the Initial Movement and the Reaction Time (right graph in **Figure 10**) is more significative in the 3D task, indicating that the reaction time in 2D task produces the need to perform a longer correction distance to compensate the deviation error, although the nominal value of reaction time in 2D tasks is smaller than in 3D tasks. This, also affects in the trajectory performing.

While conducting trajectories to reach the targets, the robot manipulator was displaced to a lesser extent in the 2D task. Therefore, the path length was higher in the 3D task for all the patients. Consequently, the total time to complete all the trials was upper in the 3D task. Only one patient has required less time to perform the 3D task. It may be that the patients guide the robotic manipulator better when they are observing a 2D environment. In the 3D task, the depth of the scenario increases the difficulty level to complete the targets. The patients have adapted better to the 2D task. Although the path length was higher in the 3D task, the times to complete both tasks do not differ substantially due to the patients achieved a maximum speed of movement in the 3D task. The correlation between the Length Path and the total time to reach one target, only the 3D task has a relevant level of significance with a very strong relation value (**Figure 11**). The 3D visualization provides a linear behavior between these two parameters and consequently allow to perform more natural movements, due to a better association between the movement of the end-efector and the avatar movement in the virtual task.

All trials can be performed in an optimal way following a straight trajectory from the central target to the reachable target and vice versa. However, a deviation has been produced when the patients started the movement to complete all trials. This deviation entails that the patients accomplished an incorrect path before the trajectory direction was corrected to reach the optimal

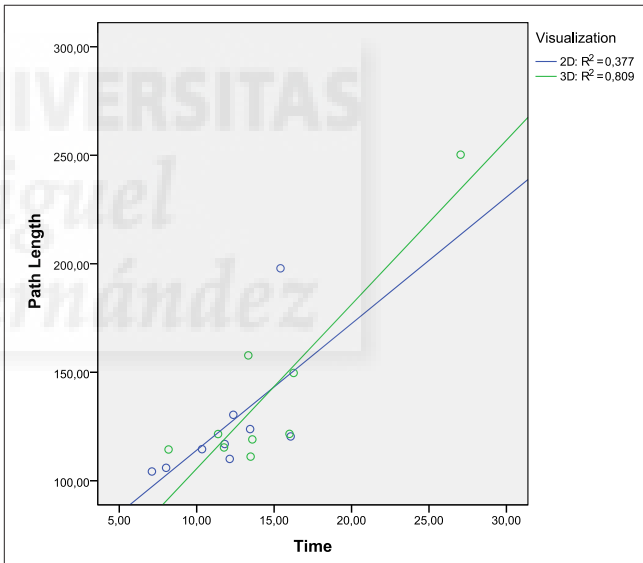


FIGURE 11 | Dispersion diagram between Path Length and Time parameters.

path. This situation is more accentuated in the 3D task in almost all patients, as it be observed through the nominal analysis of the Initial Movement and Initial Movement Direction Error, and the dispersion diagram shown in **Figure 10**.

Regarding the Maximum Speed, there are strong coefficients of correlation between it and the others parameters. **Figure 12** collects 4 dispersion diagrams where the Maximum Speed is analyzed with the parameters that provide a higher correlation. The two graphs placed in the left side indicate that the Maximum Speed has better correlation with the Initial Movement and the Initial Movement Direction Error in the 2D task. In general it can be said that the 2D task provides a better control of the

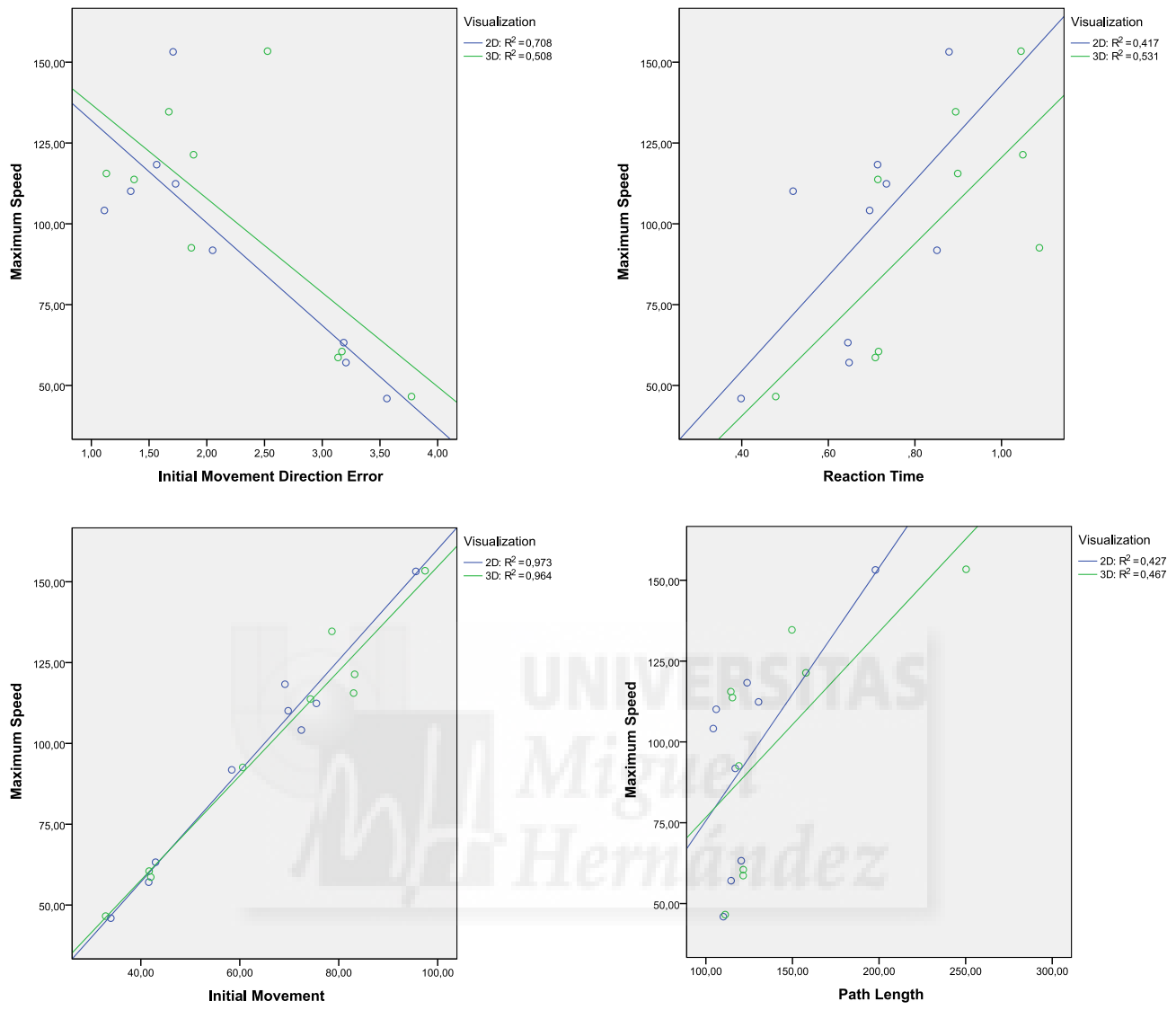


FIGURE 12 | Dispersion diagrams with the significative correlation of speed maximum.

end-effector velocity, taking into account the start of movement. The correlation between the Maximum Speed and Reaction Time or Path Length was better in the 3D task. These two correlations may suggest that the 3D task allows the user to perform longer trajectories but with more natural and dynamic movements. With the rest of parameters, correlation significative was not found.

In conclusion, the main purpose of this study was to verify if there were differences in patterns of kinematic movements of post-stroke patients assisted by a robotic device when environments were visualized in two and three dimensions. Despite the similarity in the results and the correlation analysis, the hypothesis that consists of showing a visualization environment more natural increasing the immersion level did not provide many improvements with regard to an environment simpler. Therefore, the using of 2D environments in virtual therapy may be a more appropriate and comfortable way to perform tasks for upper limb rehabilitation of post-stroke patients, in terms of accuracy in order to effectuate optimal kinematic trajectories. Knowing what virtual environment is more appropriate to each user, therapies with better assessment tools can be implemented and adapted to the patient's needs and limitations (Morales et al., 2014). Depending of the assessment objective about the sensorimotor function such as time reaction, speed or stability of movement, one type of visualization or other can be used. This is highly advantageous in the clinical environment to enhance the course of the rehabilitation of sensorimotor function and reduce the recuperation times.

AUTHOR CONTRIBUTIONS

Conceived and designed the experiments: FB, NG, JS. Performed the experiments: LL, SE, JD. Collected the data and processing

them: LL, AB, SE. Analyzed the data and interpreted the results: LL, FB, JS, NG. Wrote the paper: LL, JD, FB, NG. All authors read and approved the final manuscript.

ACKNOWLEDGMENTS

This work has been supported by the European Commission (ICT-22-2014: Multimodal and Natural computer interaction) through the project AIDE: "Adaptive Multimodal Interfaces to Assist Disabled People in Daily Activities" (Grant agreement no: 645322) and through the project HOMEREHAB: "Development of Development of Robotic Technology for Post-Stroke Home Tele-Rehabilitation—Echord++"(Grant agreement no: 601116), by the Ministry of Economy and Competitiveness through the project DPI2015-70415-C2-2-R and The Biomedical Research Networking Center (CIBER). We are very grateful to Hospital de la Pedrera for its collaboration in this study.

REFERENCES

Badesa, F. J., Llinares, A., Morales, R., Garcia-Aracil, N., Sabater, J. M., and Perez-Vidal, C. (2014a). Pneumatic planar rehabilitation robot for post-stroke patients. *Biomed. Eng. Appl. Basis Commun.* 26:1450025. doi: 10.4015/S1016237214500252

Badesa, F. J., Morales, R., Garcia-Aracil, N., Alfaro, A., Bernabeu, A., Fernandez, E., et al. (2014b). "Robot-assisted rehabilitation treatment of a 65-year old woman with alien hand syndrome," in *2014 5th IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics* (São Paulo: IEEE), 398–402.

Badesa, F. J., Morales, R., Garcia-Aracil, N., Sabater, J., Casals, A., and Zollo, L. (2014c). Auto-adaptive robot-aided therapy using machine learning techniques. *Comput. Methods Programs Biomed.* 116, 123–130. doi: 10.1016/j.cmpb.2013.09.011

Badesa, F. J., Morales, R., Garcia-Aracil, N., Sabater, J. M., Perez-Vidal, C., and Fernandez, E. (2012). Multimodal interfaces to improve therapeutic outcomes in robot-assisted rehabilitation. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* 42, 1152–1158. doi: 10.1109/TSMCC.2012.2201938

Bertomeu-Motos, A., Lledó, L. D., Díez, J. A., Catalan, J. M., Ezquerro, S., Badesa, F. J., et al. (2015). Estimation of human arm joints using two wireless sensors in robotic rehabilitation tasks. *Sensors* 15, 30571–30583. doi: 10.3390/s151229818

Bohannon, R. W., and Smith, M. B. (1987). Interrater reliability of a modified ashworth scale of muscle spasticity. *Phys. Ther.* 67, 206–207.

Brooke, J. (1996). "SUS: A "quick and dirty" usability scale," in *Usability Evaluation in Industry*, eds P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClealland (London: Taylor and Francis), 189–194.

Burdea, G. (2002). "Keynote address: virtual rehabilitation-benefits and challenges," in *1st International Workshop on Virtual Reality Rehabilitation (Mental Health, Neurological, Physical, Vocational) VRMHR* (Lausanne), 1–11.

Collin, C., and Wade, D. (1990). Assessing motor impairment after stroke: a pilot reliability study. *J. Neurol. Neurosurg. Psychiatry* 53, 576–579.

Crosbie, J., Lennon, S., Basford, J., and McDonough, S. (2007). Virtual reality in stroke rehabilitation: still more virtual than real. *Disabil. Rehabil.* 29, 1139–1146. doi: 10.1080/096382806009609090

Doyon, J., and Benali, H. (2005). Reorganization and plasticity in the adult brain during learning of motor skills. *Curr. Opin. Neurobiol.* 15, 161–167.

Einav, O., Geva, D., Yoeli, D., Kerzhner, M., and Mauritz, K.-H. (2011). Development and validation of the first robotic scale for the clinical assessment of upper extremity motor impairments in stroke patients. *Top. Stroke Rehabil.* 18(Suppl. 1), 587–598. doi: 10.1310/tsr18s01-587

Fluet, G. G., and Deutsch, J. E. (2013). Virtual reality for sensorimotor rehabilitation post-stroke: the promise and current state of the field. *Curr. Phys. Med. Rehabil. Rep.* 1, 9–20. doi: 10.1007/s40141-013-0005-2

Garcia, N., Sabater-Navarro, J. M., Gugliemeli, E., and Casals, A. (2011). Trends in rehabilitation robotics. *Med. Biol. Eng. Comput.* 49, 1089–1091. doi: 10.1007/s11517-011-0836-x

García-Betances, R. I., Waldmeyer, M. T. A., Fico, G., and Cabrera-Umpiérrez, M. F. (2015). A succinct overview of virtual reality technology use in alzheimer's disease. *Front. Aging Neurosci.* 7:80. doi: 10.3389/fnagi.2015.00080

González, J. C., Pulido, J. C., Fernández, F., and Suárez-Mejías, C. (2015). Planning, execution and monitoring of physical rehabilitation therapies with a robotic architecture. *Stud. Health Technol. Inform.* 210, 339–343. doi: 10.3233/978-1-61499-512-8-339

Henderson, A., Korner-Bitensky, N., and Levin, M. (2007). Virtual reality in stroke rehabilitation: a systematic review of its effectiveness for upper limb motor recovery. *Top. Stroke Rehabil.* 14, 52–61. doi: 10.1310/tsr1402-52

Jack, D., Boian, R., Merians, A. S., Tremaine, M., Burdea, G. C., Adamovich, S. V., et al. (2001). Virtual reality-enhanced stroke rehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng.* 9, 308–318. doi: 10.1109/7333.948460

Jebara, N., Orriols, E., Zaoui, M., Berthoz, A., and Piolino, P. (2014). Effects of enactment in episodic memory: a pilot virtual reality study with young and elderly adults. *Front. Aging Neurosci.* 6:338. doi: 10.3389/fnagi.2014.00338

Kleim, J. A., and Jones, T. A. (2008). Principles of experience-dependent neural plasticity: implications for rehabilitation after brain damage. *J. Speech Lang. Hear. Res.* 51, S225–S239. doi: 10.1044/1092-4388(2008/018)

Krebs, H. I., Hogan, N., Aisen, M. L., and Volpe, B. T. (1998). Robot-aided neurorehabilitation. *IEEE Trans. Rehabil. Eng.* 6, 75–87.

Laver, K. E., George, S., Thomas, S., Deutsch, J. E., and Crotty, M. (2012). Virtual reality for stroke rehabilitation. *Stroke* 43, e20–e21. doi: 10.1161/STROKEAHA.111.642439

Likert, R. (1974). "A method of constructing an attitude scale," in *Scaling: A Sourcebook for Behavioral Scientists*, ed G. M. Maranell (Chicago, IL: Aldine Publishing), 233–243.

Lledó, L. D., Badesa, F. J., Almonacid, M., Cano-Izquierdo, J. M., Sabater-Navarro, J. M., Fernández, E., et al. (2015a). Supervised and dynamic neuro-fuzzy systems to classify physiological responses in robot-assisted neurorehabilitation. *PLoS ONE* 10:e0127777. doi: 10.1371/journal.pone.0127777

Lledó, L. D., Bertomeu, A., Díez, J., Badesa, F., Morales, R., Sabater, J., et al. (2015b). Auto-adaptative robot-aided therapy based in 3d virtual tasks controlled by a supervised and dynamic neuro-fuzzy system. *Int. J. Artif. Intell. Interact. Multimed.* 3, 63–68. doi: 10.9781/ijimai.2015.328

LLinares, A., Badesa, F. J., Morales, R., Garcia-Aracil, N., Sabater, J., and Fernandez, E. (2013). Robotic assessment of the influence of age on upper-limb sensorimotor function. *Clin. Interv. Aging* 8:879. doi: 10.2147/CIA.S45900

Maciejasz, P., Eschweiler, J., Gerlach-Hahn, K., Jansen-Troy, A., and Leonhardt, S. (2014). A survey on robotic devices for upper limb rehabilitation. *J. Neuroeng. Rehabil.* 11, 3. doi: 10.1186/1743-0003-11-3

Merians, A. S., Poizner, H., Boian, R., Burdea, G., and Adamovich, S. (2006). Sensorimotor training in a virtual reality environment: does it improve functional recovery poststroke? *Neurorehabil. Neural Repair* 20, 252–267. doi: 10.1177/1545968306286914

Morales, R., Badesa, F. J., Garcia-Aracil, N., Perez-Vidal, C., Sabater, J. M., Papaleo, E., et al. (2014). Patient-tailored assistance: a new concept of assistive robotic device that adapts to individual users. *IEEE Rob. Autom. Mag.* 21, 123–133. doi: 10.1109/MRA.2014.2304051

Mottura, S., Fontana, L., Arlati, S., Zangiacomi, A., Redaelli, C., and Sacco, M. (2015). A virtual reality system for strengthening awareness and participation in rehabilitation for post-stroke patients. *J. Multimodal User Interfaces* 9, 341–351. doi: 10.1007/s12193-015-0184-5

Norouzi-Gheidari, N., Archambault, P. S., and Fung, J. (2012). Effects of robot-assisted therapy on stroke rehabilitation in upper limbs: systematic review and meta-analysis of the literature. *J. Rehabil. Res. Dev.* 49, 479–496. doi: 10.1682/JRRD.2010.10.0210

Nvidia, D. Z. (2011). *Physx sdk.* Available online at: http://developer.nvidia.com/physx-downloads

Papaleo, E., Zollo, L., Garcia-Aracil, N., Badesa, F. J., Morales, R., Mazzoleni, S., et al. (2015). Upper-limb kinematic reconstruction during stroke robot-aided therapy. *Med. Biol. Eng. Comput.* 53, 815–828. doi: 10.1007/s11517-015-1276-9

Pollock, A., Farmer, S. E., Brady, M. C., Langhorne, P., Mead, G. E., Mehrholz, J., et al. (2014). Interventions for improving upper limb function after stroke. *Cochrane Database Syst. Rev.* 11:CD010820. doi: 10.1002/14651858.CD010820.pub2

Rand, D., Kizony, R., Feintuch, U., Katz, N., Josman, N., Weiss, P. L. T., et al. (2005). Comparison of two VR platforms for rehabilitation: video capture versus HMD. *Presence Teleoperat. Virtual Environ.* 14, 147–160. doi: 10.1162/1054746053967012

Rose, F. D., Brooks, B. M., and Rizzo, A. A. (2005). Virtual reality in brain damage rehabilitation: review. *CyberPsychol. Behav.* 8, 241–262. doi: 10.1089/cpb.2005.8.241

Saposnik, G. (2016). "Virtual reality in stroke rehabilitation," in *Ischemic Stroke Therapeutics*, ed B. Ovbiagele (Cham: Springer), 225–233.

Steve, S. (2013). *Ogre3d: The Object-Oriented Graphics Rendering Engine*. Available online at: http://ogre3d.org

Subramanian, S. K., and Levin, M. F. (2011). Viewing medium affects arm motor performance in 3d virtual environments. *J. Neuroeng. Rehabil.* 8:36. doi: 10.1186/1743-0003-8-36

Turolla, A., Dam, M., Ventura, L., Tonin, P., Agostini, M., Zucconi, C., et al. (2013). Virtual reality for the rehabilitation of the upper limb motor function after stroke: a prospective controlled trial. *J. Neuroeng. Rehabil.* 10:85. doi: 10.1186/1743-0003-10-85

Volpe, B. T., Huerta, P. T., Zipse, J. L., Rykman, A., Edwards, D., Dipietro, L., et al. (2009). Robotic devices as therapeutic and diagnostic tools for stroke recovery. *Arch. Neurol.* 66, 1086–1090. doi: 10.1001/archneurol.2009.182

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

SOFTWARE—PRACTICE AND EXPERIENCE
Softw. Pract. Exper. 2017; **00**:1–17
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/spe

A software platform for virtual tasks implementation with adaptable difficulty level in robot-assisted neurorehabilitation therapies

Luis D. Lledó*, Arturo Bertomeu-Motos, José M. Catalán, Jorge A. Díez, Franciso J. Badesa, José M. Sabater-Navarro and Nicolás García-Aracil

Neuro-Bioengineering Research Group, Miguel Hernández University, Avda. de la Universidad S/N, 03202 Elche, Spain

SUMMARY

The rehabilitation assisted by robotic devices for post-stroke patients has been increased in recent years. This rehabilitation provides a visualization system for motivational purposes and movements guidance. However, an implementation of a wide range of activities is necessary in order to remove the boredom feeling increasing the attention and motivation of the patients. The aim of this work is to develop a software tool to allow the generation of virtual tasks intended for upper-limb therapies in robot-assisted neurorehabilitation. This paper describes the architectural aspects of the virtual reality-based software system followed in the implementation of rehabilitation tasks assisted by end-effector configuration robots. These tasks can be configured by the therapist according to the specific needs of each patient. One of the most important features is the adaptation capacity of the difficulty level regarding the patient's psycho-physiological state. Therefore, it is possible to maintain a constant attention condition by patients. Several virtual tasks have been implemented to verify the versatility and the potentiality in the development of any type of virtual environment. The system has proved that it is an effective method to online run the virtual tasks together with the robot communication and the physiological signals acquisition system. The only requirements in the implementation of one virtual task are to establish the organization of visual elements, to define the task targets and to indicate the properties of the difficulty level. Four post-stroke patients have participated in a preliminary validation study where they have received a robot-assisted neurorehabilitation therapy using this software platform. Copyright © 2017 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Neurorehabilitation; Software architecture; Biocooperative system; Virtual reality; Visual rendering; Optimization process

1. INTRODUCTION

Stroke has been one of the main cause for chronic deterioration in neurological field during the last decades. It is a cerebral injury that affects abruptly in the vital development of people [1] and can cause cognitive, motor, organic, sensorial or emotional disorders. Physical problems [2] such as spasticity, muscle weakness or coordination and movements limitation reduce the quality of the patients life having a negative impact in their autonomy and independence with respect to the immediate environment relation. Therefore, this injury harms directly in the realization of the activities of daily living (ADL) [3] and, consequently, a damage in the social relations appears. The aftermath of a cerebral injury can be partially recovered in most cases. For this purpose, it is essential the implementation of a rehabilitation program [4] where the patients have to perform physical exercises based in voluntary and specific movements [5]. This type of rehabilitation is based

*Correspondence to: Neuro-Bioengineering Research Group, Miguel Hernández University, Avda. de la Universidad S/N, 03202 Elche, Spain. E-mail: lledo@umh.es

on the cerebral plasticity concept to benefit from the capacity of functional adaption of the central nervous system for the reduction of the physiological alterations [6]. These therapeutic exercises must be performed repeatedly in order to promote the post-stroke cortical organization [7].

Robotic devices such as prosthesis, exoskeletons or end-effector configurations robots have been used to assist the patients in neurorehabilitation therapies [8]. There are several studies in the scientific literature that suggest the use of robotic devices to improve the quality in the neurological rehabilitation [9]. These systems are complemented with Virtual Reality (VR) to increase the patient motivation, improving their implication in the therapy. VR can be used to develop therapeutic games with virtual rehabilitation environments. These games can be oriented towards the recovery of functional abilities to perform ADL or imaginary tasks. Independently, multiple advantages of the VR have been demonstrated in several studies [10] for stroke rehabilitation. For instance, a study with 376 subjects concluded that "VR rehabilitation in post-stroke patients seems more effective than conventional interventions in restoring upper limb motor impairments and motor-related functional abilities" [11]. In short, the authors pointed out that the improvements in the group exposed to virtual rehabilitation therapy were greater than in the group exposed to classical rehabilitation therapy [11]. An overview of sensorimotor rehabilitation studies based in virtual reality has been performed in [12]. Furthermore, VR has been used in therapies with other type of cerebral injuries such as cerebral palsy [13] or Alzheimer's disease [14]. Moreover, VR environments have been used in teleoperated robotic systems [15].

There are a lot of software tools that uses VR in the development of virtual tasks. For a fast development of this type of VR-based therapeutic game, a new concept of software modulation called Game Engine [16] appeared in the 1990's. In this manner, the core architecture is divided in different modules, such as the visualization rendering, dynamic simulation and sound reproduction systems. The component separation facilitates the generation of interactive applications of the same type with graphics in real time without changing the programming core. Thus, the major part of the effort is brought to design new virtual elements.

The current evolution of these type of systems has proved that virtual environments play a strong part in neurological rehabilitation. However, there are still many challenges to be resolved. An important issue is the need for promoting the active participation of the patient inside of the system control loop. This refers to the fact that the user have to cooperate with this kind of systems. In recent years, new approaches have been addressed to deal with this issue and improve the robotic rehabilitation methods through bio-cooperative competencies [17]. In that way, the patient is not only considered as a passive receiver of the trajectories performed by robotic devices, but it is the responsible member of close the system control loop providing physiological information about their emotional state. This fact allows the adaptation of the therapy intensity and maximize the patients participation [18]. Therefore, an ideal neurorehabilitation system would be able to decide the difficulty level in different environment taking into account the biomechanical and physiological information of the patients.

In this work, a software architecture system has been proposed in order to help the generation of virtual tasks in a quick and adaptable way to be used in neuro-rehabilitation therapies assisted by robotic devices. Basically, the patients are be able to control a robotic device while a virtual environment is shown to complete the goal of the tasks. This architecture allows to optimize the process generation of virtual tasks using the minimal resources and the low possible costs. As this system has been design to post-stroke patients with different injury levels, it is able to modify therapy parameters such as the number of repetitions, the time to complete the trajectories and the assistance level of the robotic device, without change the source code. In addition, the adaptation capacity of the difficulty level of the virtual tasks has been added to address the bio-cooperative level.

2. DESIGN CONSIDERATION

The reason why this software platform has been developed and the design requirements are summarized in this section. Neurorehabilitation therapies are linked to persons who need the

DOI: 10.1002/spe

recovery of the physiological condition lost due to some injury or neurological disease. This recovery process is carried out through the performance of repetitive tasks due to the brain plasticity. A task variety is required to involve a better treatment adherence by the patients and improve their motivation-attention state. Thus, the frustration and boredom feelings are avoided, the main factors that prompt to the patient to leave the therapy.

The standard generation of virtual tasks involves the individual development of a software program to each virtual environment. The design of virtual elements in a modeling program, establishing the organization of the environment with the physical behavior manually and task goals definition is required. This process can be laborious and slow when it is necessary to implement high number of tasks. Therefore, this software platform pretends to unify the implementation process of different personalized neurorehabilitation tasks in only one program. Unification of this process is possible because all exercises, in this type of rehabilitation, share the same goal: repeat movement paths. The only difference between tasks is the visual rendering of the virtual environment.

An object-oriented design methodology has been used in order to implement the software platform. The core architecture has been divided in different modules and routines, following the Game Engines design. Then, a process of development time optimization of a different virtual tasks is possible through the implementation of the tasks behavior inside the source code using a scripts system. Regarding the simulation of the virtual tasks, a series of requirements has been proposed to generate convincing tasks with the sufficient realism and interaction capacity through the patients senses. These requirements are the following:

1. The virtual objects must be displayed with an appearance as much as possible to reality through high density of polygons and quality textures.
2. The elements involved in the physical simulation have to behave in a similar way to reality.
3. A real-time interaction [19] is required for a correct immersion trying to avoid little interruptions in the visual perception of the patients.
4. Sound reinforcement to indicate actions or goals accomplished inside the virtual task.

The system has been designed in order to interact with robotic devices based on an end-effector configuration to upper limb movements. The interaction that the patient can perform with these robotic devices is a displacement of the end-effector inside a workspace to perform rehabilitation exercises with specific upper-limb movements. In a virtual environment, the patient leads the end-effector to move a controllable object or an avatar in order to achieve the targets and complete the task.

2.1. Performance context

This software system has been designed to interact with neurorehabilitation systems divided in three components: an end-effector robotic device, a physiological signal acquisition system and a software launcher. An overview on the proposed system and the context where TaskENGINE is applied within a neurorehabilitation assisted therapy is shown in Fig. 1.

Due to the global system architecture, three possible user profiles have been identified with their own roles: Patient, Therapist and Task Designer. Each role has different situations when they interact with the system. Task Designer is responsible for the tasks implementation and data storage, obtaining virtual elements, configuration files, sounds, and graphical interfaces. This role establishes the task environments following the therapist's indications about how the distribution of the scene is and what elements interact with the Patient, as well as the targets to be accomplished. Therapist is a clinical person who carry out the therapy tracking and setting the objectives that the Patient has to complete in order to achieve a notable recovery of the motor skills. Therapist has to guide the Task Designer in the tasks development to determine what environments are better adjusted to a proper movement recovery. The software launcher is used by the Therapist to select the task that the Patient has to complete and modify the internal parameters to adapt the system to a specific patient. The person who has been affected by a neurological injury and performs the virtual neurorehabilitation therapy assisted by robotic devices is the Patient under the under Therapist supervision. Patient controls the robotic device to manage the position and orientation of a avatar inside of the virtual

DOI: 10.1002/spe

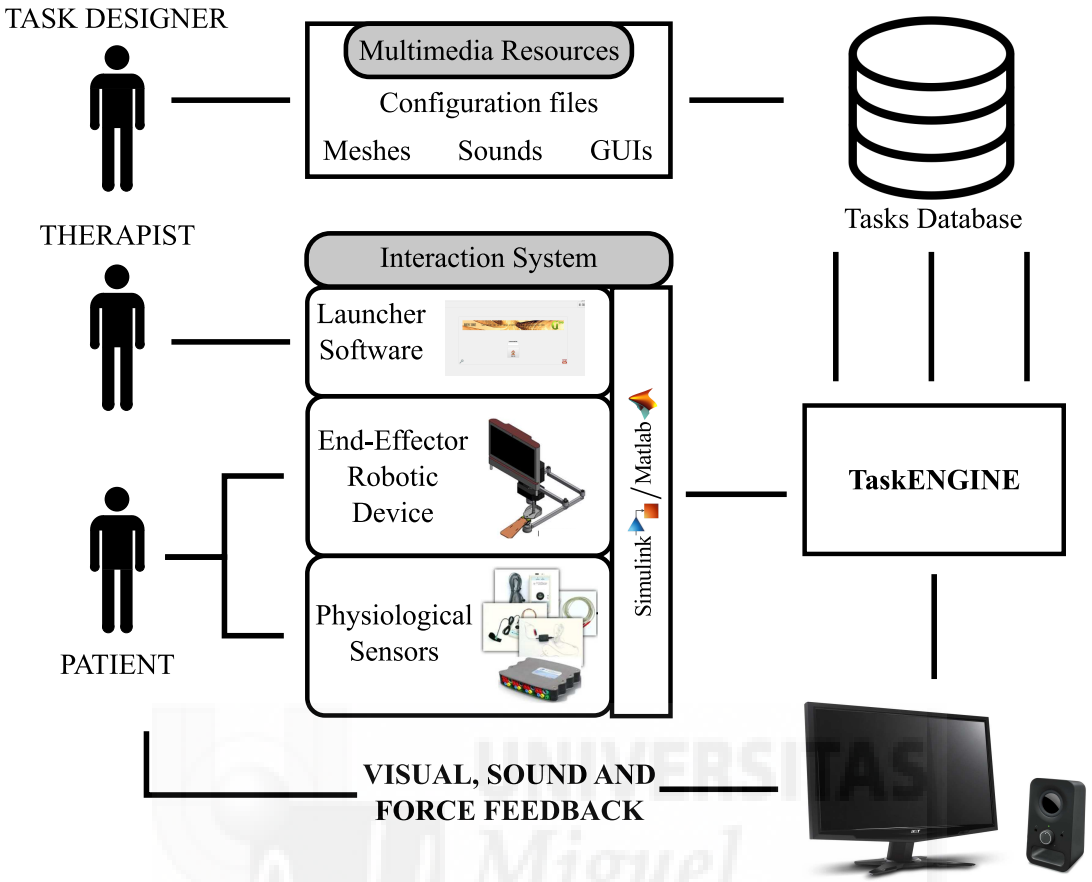


Figure 1. Overview of the elements that interact with TaskENGINE.

task and interact with the visual elements to accomplish the task goals. On the other hand, the acquisition system estimate the emotion state of the Patient through the physiological sensors to online adapt the difficulty level of the task [20]. The acquired bio-signals were pulse rate, respiration rate, skin conductance level (SCL), skin conductance response (SCR) and skin temperature. The estimation of the emotional state and the robot control were performed by a Matlab/Simulink-based complementary program.

3. SOFTWARE PLATFORM ARCHITECTURE

The proposed virtual reality system (TaskENGINE) is designed to provide a tool in the design and execution of adaptable and customized virtual tasks with any type of 3D environment based on physics principles. These tasks provide a visual feedback in real-time the movements performed by patients in neurorehabilitation therapies assisted by robotic devices. TaskENGINE is a software platform that loads multimedia resources files such as visual objects, physical shapes, sounds or Graphical User Interfaces (GUI) to simulate a task scenario with path goals. Multimedia resources files compose a resource databases created during the design of the new virtual task. The resources files can be reused and merged for fast prototyping of other tasks. The Therapists requirements are taking into account in order to perform the task. TaskENGINE is composed of programming modules responsible of accomplish automatically a series of basic assignments:

- Organize the virtual environment.
- Manage physical and visual behavior.

DOI: 10.1002/spe

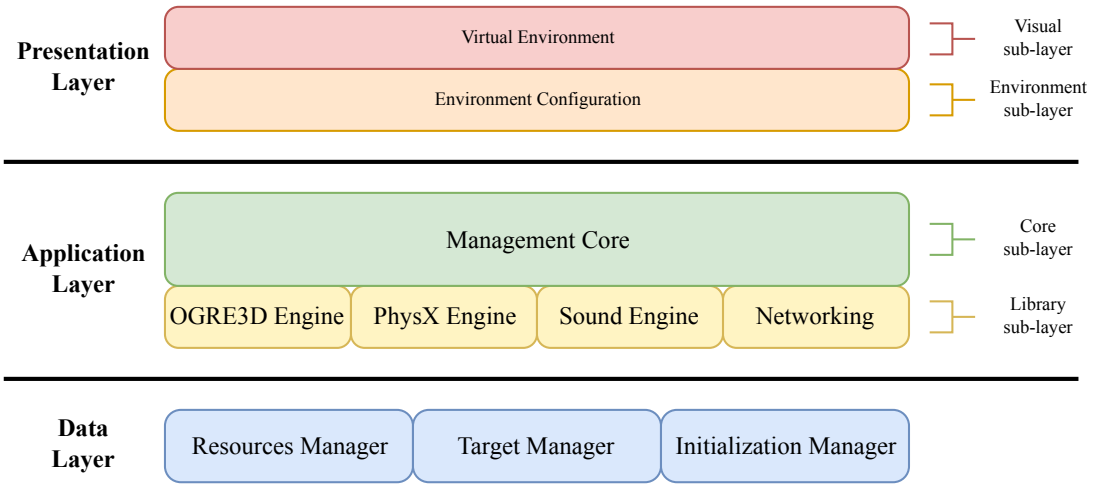


Figure 2. Modular three layer-based architecture of TaskENGINE.

- Handle the sound events and graphic interfaces.
- Communicate the virtual environment with the robotic device.
- Assign task goals and track its compliance.
- Adapt the task difficulty level regarding the emotion state estimation.
- Modify the internal task behavior according the therapist's suggestions.

The software architecture of TaskENGINE has been designed following a programming model based on three layers [21]. Each layer is composed by various modules that can be organized depending on the managing operation. This architecture provides a software segmentation to isolate the application logic in different modules. Thus, the software platform can be developed in parallel. Fig. 2 shows layer distribution: Data, Application and Presentation.

TaskENGINE has been defined through programming modules, formed by classes, to model the three layers named above being structured by a class scheduling. The resulting class diagram of all modules is presented in Fig. 3. Therefore, the implemented procedure has been designed following an object-oriented scheme based in routines, protocols and open source tools. The three layers are explained in the following sections.

3.1. Application Layer

Application layer sustains the software execution establishing all the logic rules. It constitutes a link with the remaining layers to exchange requests about storage or retrieval data, present results and run the tasks life cycle. This layer unifies the graphics rendering engine with the audio and physics engine.

3.1.1. Library sub-layer The main components of the Game Engines are taken into account to compose the Library sub-layer. These components are the graphical, physical and sound engines. Therefore, free libraries written in C++ has been selected. OGRE3D [22] has been used as graphics engine for rendering quite realistic, interactive and real-time environments. NVIDIA PhysX [23] has been chosen as physics engine to simulate elements with a high degree of realism through collision detection algorithms able to calculate interactions between all the elements of the physics scene. A sounds engine has been also incorporated to play sound effects according to the task objectives and the scene elements.

Regarding networking component, a socket systems through UDP communication has been implemented to perform the data transference between the software platform and the external elements. Position and orientation data of the end-effector are received from the robotic device. Moreover, commands to change the difficulty level of the task are obtained from the external system

DOI: 10.1002/spe

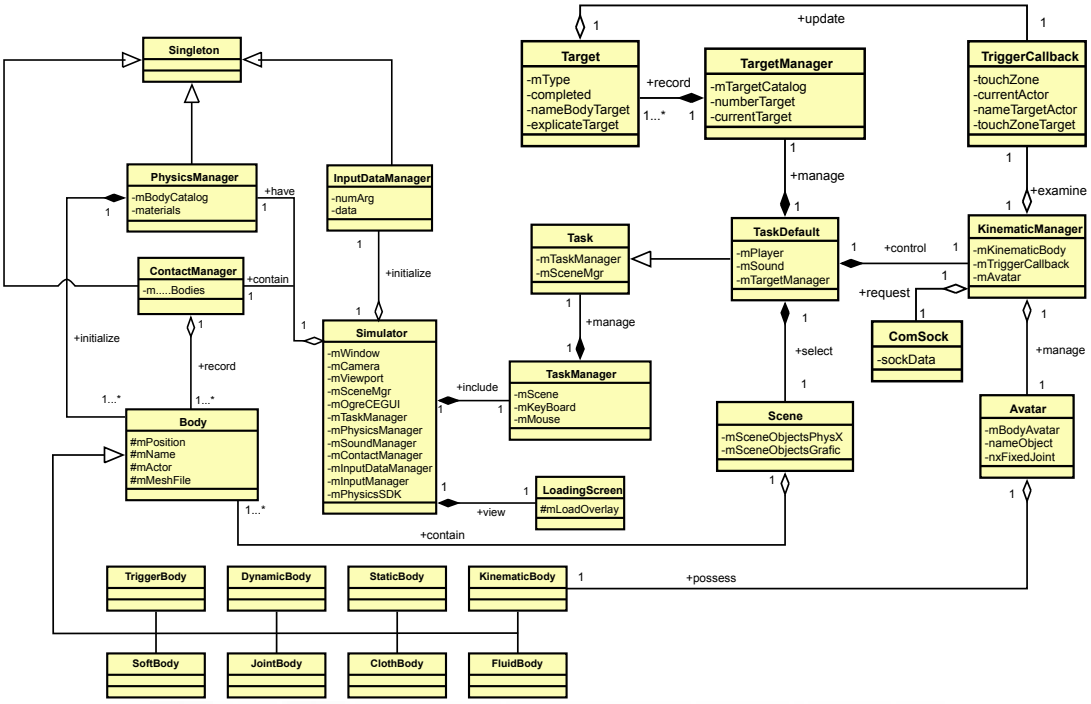


Figure 3. Class diagram of software platform.

that acquires, processes and classifies the Patient's physiological signals [24]. On the other hand, the system can send information to the robotic device about the goals of the current task in order to interact with the patient.

This sub-layer initializes the main features of the graphic engine such as the root elements statement, load rendering plugins and obtain a keyboard events handler. The physics engine is also initialized to generate the physics simulation scene. When the physics engine is initialized, a physic world is defined to associate the collision shapes with the visual elements. Therefore, each element of the virtual environment will have a physic shape, that acts inside of the physic simulation, and a visual mesh attached to a node managed by OGRE3D. With this node the collisions generated from the physics engine can be shown in screen. The update process of the physics simulation is carried out by this sub-layer. On the other hand, sound engine is started allowing the load of the sounds. The only class developed to this sub-layer is *ComSock*. It is responsible of the network communication establishment storing and registering data from the external systems.

3.1.2. Core sub-layer The connection with all the modules implemented in this software is performed by the Core sub-layer. Its main goal is to manage the system initialization and to control the render loop. This fact involves the initialization of the action engines and the data managers, as well as handles the internal operation of the tasks and the registers of the keyboard events. These actions are carry out by *TaskManager* and *Simulator* classes.

The state of the task simulation are modeled by *Task* and *TaskDefault* classes. These classes give the scenario load order as well as the elements update order and the management of the compliance with targets to send the performed results of the patient. Furthermore, it records all the collision elements that they must be updated in the main loop using *ContactManager* class.

This sub-layer generates all the visual elements and configures the collision shapes that will act inside the physic simulation. Then, each physics model consists in a simplified geometry of each graphical model to provide the collision core within the simulated physics space.

Regarding the avatar control, this module displaces the avatar through the physical simulation using the data obtained by networking component and sends this information to the Presentation

DOI: 10.1002/spe

layer to be render by *Avatar* class. That process is performed by *KinematicManager*. *KinematicBody* is in charge of position and orientation management deciding if one target object has to be picked up or released to complete the clamping features. It also conducts complementary calculations about the collision between the controlled avatar and their bodies, i.e. elements such as soft bodies, cloths, fluids or joints.

Difficulty level can be updated depending on the patient's physiological state received from networking module. Thus, this module dynamically changes the features that determine the difficulty level acquired by the Data layer. Finally, the targets update is also performed by this module and a collision management between the scene elements is required. For this reason, a class of event capture, called *TriggerCallback*, is added in order to indicate when one element is introduced inside of a target trigger. Apart from the update process, also several visualization markers have been defined to indicate the position or the object to be reached. Additionally, sound reproductions and object animations can be administered by this sub-layer to improve the task interaction.

3.2. Presentation Layer

This layer presents the virtual task to the patients transmitting the visual information. Normally, the capture data process from patients is carried out by this layer using any type of GUI. However, in this case this process is performed by the Application layer due to the only interaction between the patient and the system is produced via networking communication.

Visual sub-layer manages the necessary elements to visualize the virtual environments in order to create a window in the screen where the virtual content is shown and to initialize a camera with its viewport and place it in the appropriate place. Furthermore, the visualization of the Patient's avatar is also conducted by this model. Thus, the displacement of the robotic device is translated into a displacement of the avatar inside of the virtual environment. Before the virtual environment visualization, a Loading Screen is presented to show the resources loading. In addition, this module is be able to render GUIs providing information about how to complete the task. GUIs inclusion is completely optional as these GUIs only consist in one static text that shows a description about the targets and an scrollbar to indicate the time remained to complete the task.

Environment sub-layer is responsible for the control and the organization of the virtual scene. It accesses to the render loop and records a list of the objects that have to update their position and orientation depending on the collisions detection carried out by the physics engine, including the controllable avatar. This component constitutes a bridge between the patient and the virtual environment, and provides the visual entities to the Visual sub-layer to render them.

3.3. Data Layer

This layer consists in a modules series that are in charge of load management of the required resources to run the task by reading meshes, materials, data scripts and XML files. This is the layer where the resources data are stored and recovery information from the Application layer is received. Basically, all managers are initialized to register, process and organize external resources, and convert them in understandable elements to the software platform.

3.3.1. Resources Manager This module handles the generation and the access management to the virtual environment elements. It deals with two activities: Physical and Visual settings. The first defines the scene physical part and, hence the physics data file is processed to initialize a catalog of collision bodies that can form part of the task environment. These collision shapes are represented by boxes, capsules, planes, spheres and convex bodies. This process is performed by *PhysicsManager* class. The second activity generates the visual part using virtual meshes associated to the physics shapes. Furthermore, it processes the scene data presented in XML files and stores an instance of the visual element and its scene node. *Scene* class accomplishes this role. The resources that this module can deal are:

DOI: 10.1002/spe

- File with *.physics* extension using XML language that collects the position, orientation and scale from each of the physic shapes that overlaps all the visual geometry.
- File with *.scene* extension that contains the spatial organization of the scene where the position, orientation and scale of all elements are defined. Furthermore, this file incorporates additional features such as the mesh name, node name, shadow projection, environment lights, etc.
- Files with *.mesh* extension for each mesh formed by vertices and faces. This file contains information about the position and rotation of each vertex. Also, this file references the materials and animations of the object. The selected rendering engine uses this own resource format to optimize the scene performance.
- Files with *.material* extension where the surface features are detailed.

Consequently, this module create associations between visual meshes, scene nodes and physical shapes to get several element environment with different physical behavior. The objects that can be implemented are: static elements; dynamic elements; Kinematic elements; volumetric deformable elements; plane deformable elements; particles system with physical behavior to simulate fluids; elements linked by restrictions; zones that marks target positions; visual particles system; and graphic elements without collision geometry. These collision elements are defined by an hierarchy classes inherited of the *Body* class. Each class contains all the methods and attributes needed to manage the creation of the graphic and physical parts.

There are two other types of resources that this module can deal. One of them are the sounds that can be used to assign audio content to each interactive object. Hence, a better immersion of the patient inside the virtual task can be created. The last resource consists in a GUI with a compatible format to the rendering engine.

3.3.2. Target Manager Once the virtual environment is established and visual elements are associated to their physical models, it is necessary to provide functionality to the rehabilitation task through targets. These targets define the specific movements that the patients have to perform to complete tasks, indicating the virtual objects that interacts in each moment. Targets definition and management is carried out by Target Manager module. *TargetManager* class is used to read the targets file and convert it in *Target* class instances.

TaskENGINE is oriented to be used by robotic devices with an end-effector configuration. For this reason, the targets have been defined depending on the possibilities of this end-effector type. As the end-effector only provides a displacement, the targets are based on reach specific points in a workspace. Therefore, targets that can be implemented with this software are:

1. Select one dynamic object through the controllable avatar.
2. Approximate the controllable avatar to a specific point in the scene.
3. Deliver the selected dynamic body at any specific point of the scene or over other dynamic body.
4. Hold the selected object at any point of the scene or over other dynamic body.

Basically, these targets consist in performing trajectories through the avatar interacting with virtual object that can be selected or moved from one point of the scene to another. This module processes the targets data file with XML extension. All information about the elements that participate in a concrete target is saved by this module. In addition, it verifies the state of the targets and decides if the conditions required to pass a concrete target is accomplished and change or not to the next target.

On the other hand, this module also is responsible to structure the difficulty levels of the tasks, processing the file where the difficulty criteria is defined. Difficulty criteria are formed by the time total to perform one trajectory, time to complete the task and the number or velocity of interactive elements that participate to accomplish one target. These properties can be modified in real time according to the physiological state of the patients.

DOI: 10.1002/spe

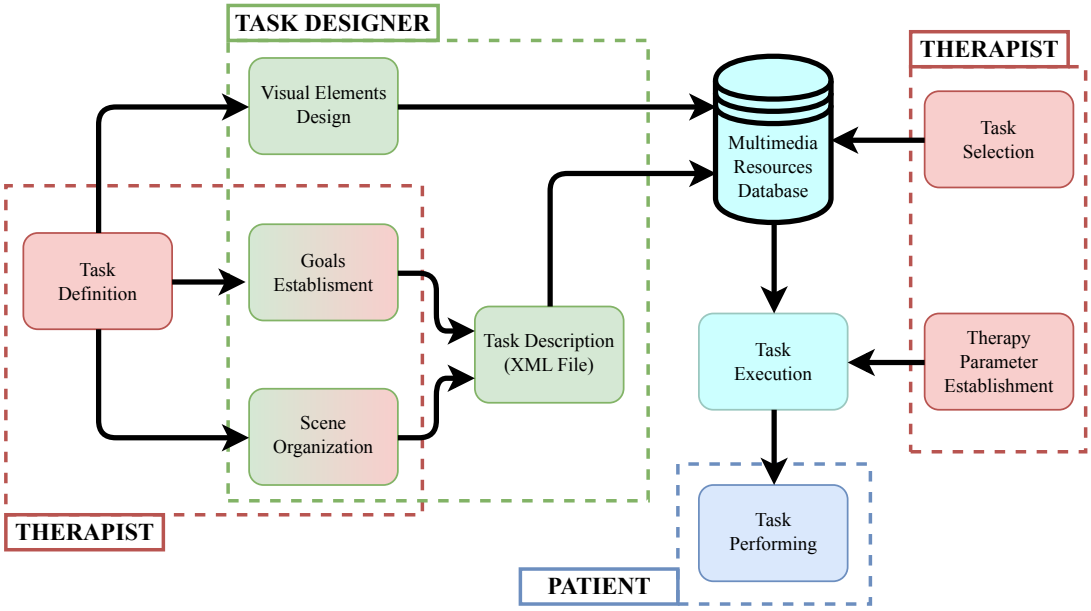


Figure 4. Usage steps of TaskENGINE to execute virtual tasks.

3.3.3. Initialization Manager Generally, the targets of the tasks are similar between them but with different visual elements, i.e. displacement of the avatar from one point to another performing trajectories. However, not all the patients have the same injury degree. This fact carries the need to adapt some therapy properties. A manager of input data have been implemented to solve this problem. Initialization Manager module records this therapy values. The input data that can be recorded by *InputDataManager* class are:

- Task name in order to have a reference about the required multimedia resources.
- Repetitions number of the trajectories.
- Time to perform one trajectory and total time to complete a task.
- Laterality of the patient's injury.
- Assistance level, movement amplitude or assistance force directly related to the parameters of the robotic device.

4. IMPLEMENTATION OF VIRTUAL TASKS

TaskENGINE allows the execution of any virtual task sharing the same type of targets and the same avatar control. However, visual elements that defines the targets and the avatar may differ in each task. The usage steps of the software platform are depicted in Fig. 4. The roles of the Therapist, Patient and Task Designer are involved in the process of the task design and their execution. These roles are highlighted in this figure. Initially, the Therapist have to decide the type of exercise that will be more beneficial to the patient in the neurorehabilitation therapy. Then, the Task Designer and the Therapist determine what elements have to appear in the scene, their spatial organization and the avatar type. The next step consists in setting the targets inside of the scene to provide interactive features. Thus, the scene elements that take part in the accomplishment of the targets have to be indicated. This way, the patients can perform trajectories to reach points within the scene, showing elements to interact with. During this step, the properties that determine the difficulty level are also defined.

Visual creation of the scene is carried out by the *Task Designer*, always following the supervision and agreement of the Therapist. This process involves the modeling of 3D meshes, material surfaces and textures. There are a lot of modeling tools software in order to generate virtual environments

DOI: 10.1002/spe

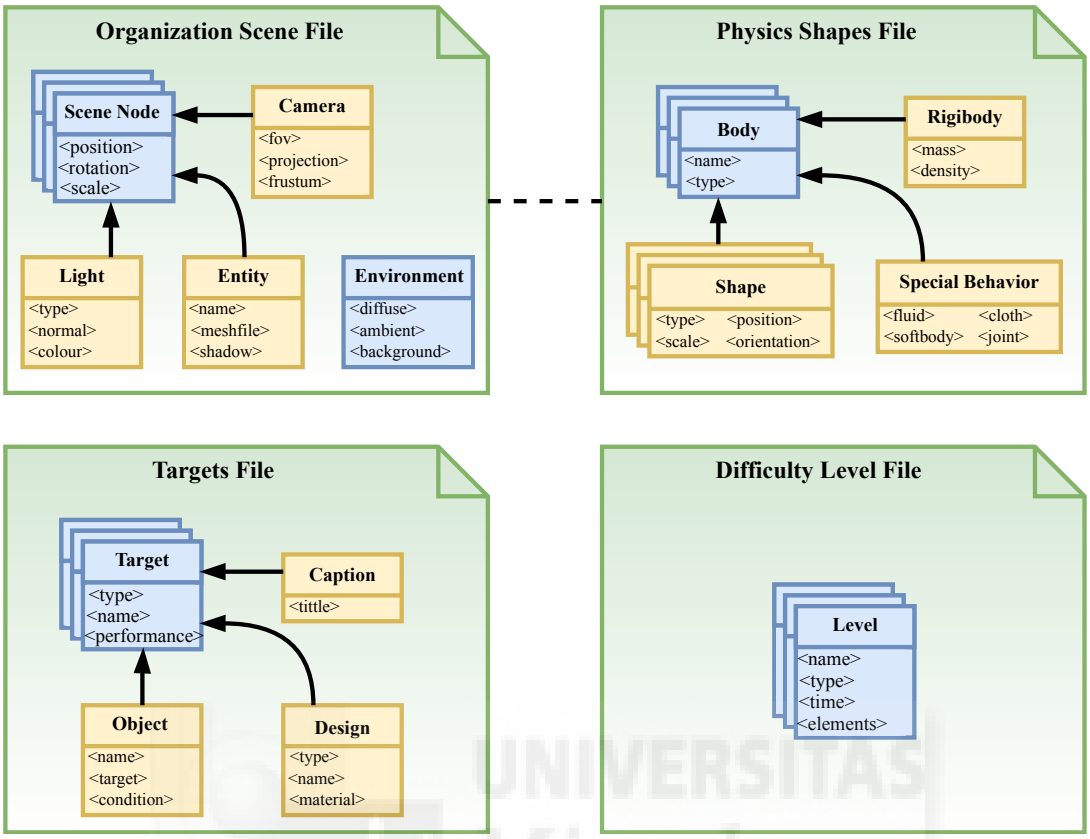


Figure 5. Configuration files structure.

with decorative elements, even the avatar. Thus, the multimedia resources database can be filled and these elements can be used in the prototyping of more tasks.

The virtual task consists of the creation of these type of visual resources and a four XML files. These four files contain the description of the virtual environment, the physics shapes inside the physics simulation, the corresponding targets and the change properties of the difficulty levels. Each file has a specific XML tags to define the features that can used in the software platform. In Fig. 5, the most important tags with their main attributes can be observed.

The scene file *Scene Node* can be used to establish the position, orientation and scale of one element such as entities, lights or cameras. There are as many Scene Node tags as elements in the virtual environment. In addition, *Environment* is required to set the background lighting properties. Physics data file is related with scene file. Each element in scene has to have a physics body associated. These physics bodies are defined by *Body* tag and can own a list of simple shapes that wrap the specific visual element. There are some tags to assign special physical behaviors such as fluids, clothes, soft bodies or joints.

On the other hand, the type of targets are defined using *Target* tag. *Object* tag is responsible for the assignment of the elements name that participate in the target. For example, it determines what element have to collision with the other or which position the avatar has to reach. *Design* tag configures the type of target marker that appears in screen indicating the goal points or target elements. Meanwhile *Caption* tag gather additional task information. In addition, the specific sounds are defined in this file. Difficulty Level file gathers elements such as the targets times or properties of interactive objects. This way different difficulty levels are assigned in a same task.

Therapist has to select the desired task from the database and establish the therapy parameter. These actions are performed using the software launcher embodied in the system control of the robotic device. The task execution involves the XML description definition, processing its tags

DOI: 10.1002/spe

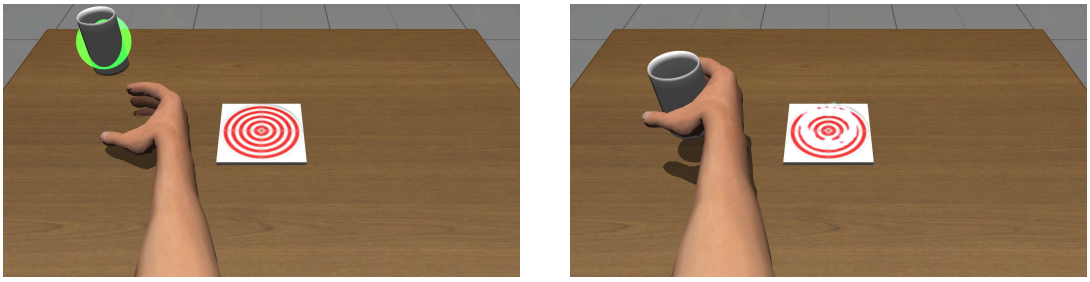


Figure 6. Targets signalization in the virtual environment.

content and interpreting the information to organize the corresponded virtual environment. It shows the interaction between the avatar and the scene, and define the interaction rules to accomplish task targets. The required resources are extracted from the database. Finally, Patient can use adaptable tasks to perform rehabilitation trajectories while having a visual feedback about the movement involving.

4.1. Motor activities of daily living

Following the therapists suggestions, several virtual task taking into account the motor activities of daily living have been implemented. These tasks must be simple and clear so that the patients can understand the targets quickly. The first task is a simple activity: pick a glass. In this case, the virtual environment is composed by two static elements (a table and a coaster), one dynamic element (a glass), one kinematic element (a virtual arm) and five zone trigger (target positions). The main purpose is to simulate the action of picking and dropping a glass.

At the beginning of the task, the glass appears on the table in a random position. The Patient has to pick it up and leave it on the coasters. This routine can be performed a number of times that the Therapist considers appropriate. Each time that the glass is left on the coasters, it appears again in a new random position. In this task, the difficulty level are determined by the time to complete one trajectory or target. Then, the auto-adaptation of the difficulty level is produced increasing the time when the patient are in a stress state and it is reduced in the relax state. Fig. 6 shows two images where the task targets are indicated.

The task explained above can be extended with more elements adding new targets. This way, the task can be modify to simulate a waiter's job providing different types of drinks depending on customer demands. Thus, new elements such as a glass dispenser, a drink dispenser and three coasters with different colors are added. A stack of glasses upon a kitchen towel has been added as a decorative element. An extra virtual arm with a tray is added to indicate the place where the Patient has to deliver the glass.

The main target of the task consists in grasp a glass, fill it with a fluid and deposit it on a tray. Consequently, the patient has to perform three actions or targets. First, the Patient has to wait until the tray that indicates the color of the drink appears. Then, the Patient approaches the virtual arm to the glass, placed in the glasses dispenser. After, the glass has to be brought to the drink dispenser, above the corresponding coaster regarding the color tray. The fluid feature of the physics engine is used to provide greater realism when the glass is filled. Therefore, the glass is placed in the drink dispenser. Then, a fluid torrent is poured so that the patient has to remain in the same position until the glass is filled. The following step constitutes the delivering goal. This three actions can be repeated any number of times depending of the Therapist criterion. Every time the tray appears a new color is shown. In this task, the success performance is measured in the time that the Patient has needed to complete all the actions. In this case, the difficulty level are determined by the time to complete these three actions. Thus, the time will be increased or reduced depending on the emotional state of the Patient. Fig. 7 shows a image of this virtual task.

DOI: 10.1002/spe

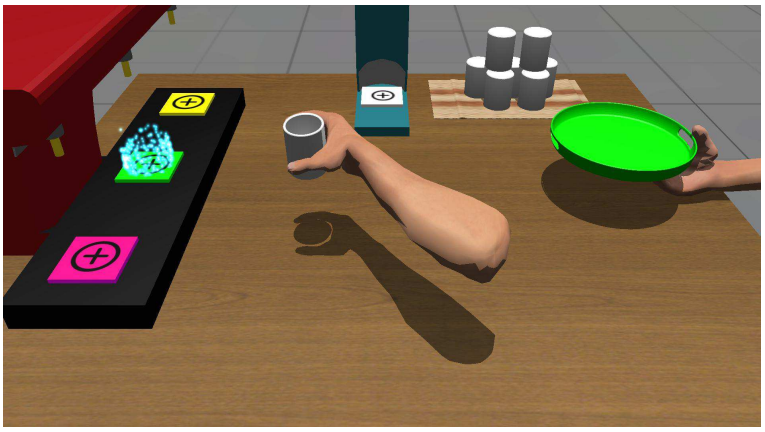


Figure 7. Virtual environment of the waiter's job task.

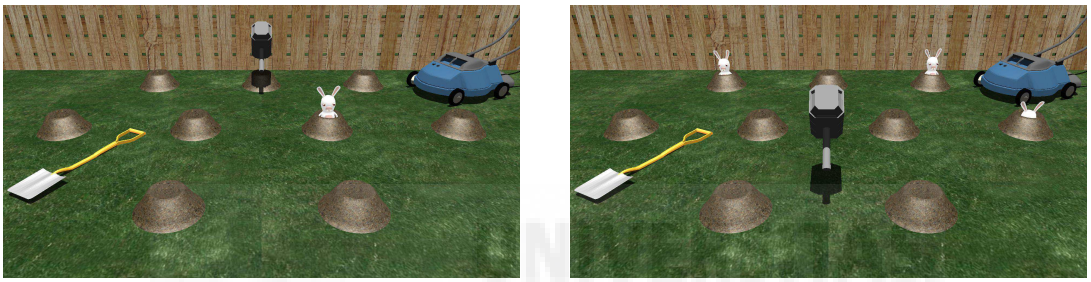


Figure 8. Screenshots of the garden task. Easy (left) and hard (right) difficulty levels.

4.2. Imaginary activities

Aside from simulation of real activities to train daily life situations, imaginary tasks can be implemented to meet needs of specific movements. First task example of this type of activity is a hitting game to elements randomly situated. This task consist in the location of elements that appear randomly in order to hit them with the controlled avatar. The scene simulates a garden formed by a wooden fence, grass and nine burrows. Garden tools were added to provide more details to the virtual environment. Patient's avatar is represented by a hammer.

Hitting rabbits that randomly appear in scene in either of the nine burrows is the main target of the task. A simulation blow is run when the patient approaches the hammer to the rabbit. The rabbits are volumetric deformable bodies, hence their elastic topology is deformed when the hammer contact them. The direction of the received forces and the internal physical behavior can cause changes in the response of the global impact. Total time that one rabbit stays in the scene and the number of repetitions can be modified by the Therapist. In this task, the difficulty level depends on two factors with respect to rabbits: time spent on scene and the number of rabbits that appear in the same time. Hard difficulty levels involves more number of rabbits in scene and less time to hit them. Furthermore, in harder levels each rabbit appear in scene with different speeds. Fig. 8 shows the virtual environment of the task with easy and hard difficulty levels.

The last example task simulates a box factory. The scene is comprised by eight platforms and a central deposit. The eight platforms are placed uniformly around the deposit, maintaining always the same distance. The box is the only interactive element that react with the avatar represented by a wrench. The task structure allows to perform two targets: pick the box situated in one of the eight platforms and drop the box inside of the central deposit. The box can appear in the scene in two ways: sequentially in the clockwise or counter-clockwise rotation, or randomly. Then, the number of repetitions and the time limit to achieve one target is required. The position of the box is warned by a color change of the platform where the box is placed. When the Patient picks the box, the

DOI: 10.1002/spe

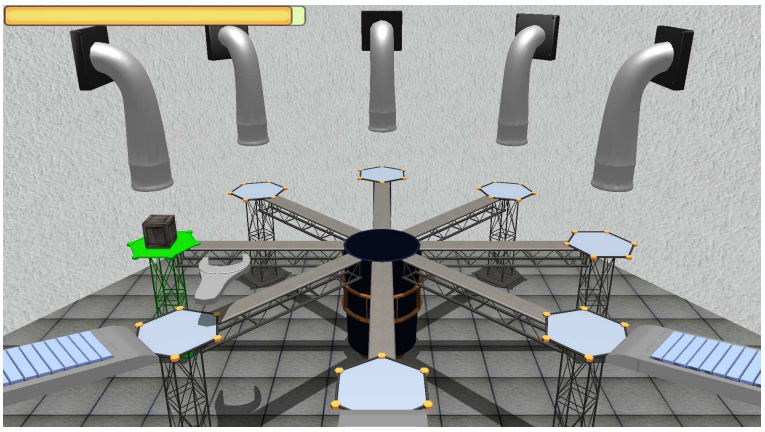


Figure 9. Virtual environment of the boxes factory task.

central deposit changes the color to indicate the next target point. In this task, the difficulty level is determined by the total time to select a box. As has been seen in previous tasks, the time will be increased in easier levels and will be reduced in harder levels. Particularly, a GUI has been added to show the remaining time to complete the picking target. A image of this task environment can be observed in Fig. 9.

5. EXPERIMENTAL RESULTS

A validation study of motor recovery of post-stroke patients has been performed to verify the effectiveness of the proposed system during robot-assisted neurorehabilitation therapies. TaskENGINE has been executed in the neurorehabilitation system called PUPArm [25] to provide the simulation and display of therapeutic games in coordination with the robot movements. This robotic system was developed by Biomedical Neuroengineering Group at Miguel Hernandez University of Elche as a rehabilitation robot for patients with stroke. The robotic structure formed by four metallic bars allows the horizontal movements of upper limbs. In addition, it is capable of record information about the patient's movements, through several robot sensors, in terms of position, velocity and forces. All data are processed and analyzed to provide an objective assessment of the neurorehabilitation therapy.

Four post-stroke patients with disorders of the upper limbs have participated in the validation study. This study has been carried out on a hospital of attention to chronic patients. All patients were informed properly about all aspect of the study, and they gave written consent before starting.

The therapy treatment consisted in perform the four activities explained in the section above using the PUPArm robot. The patients were supervised by a physical therapist in any time. During three months, the patients have received a therapy treatment. In each session, patients completed two of the four implemented tasks and the robot recorded all trajectories. The only task that have been repeated in all sessions is the task that simulates a box factory. This task is used to control the assessment of the patients. Each session is organized in three steps:

- Firstly, the patient has to complete 32 objectives in one of the four possible task. This task is executed randomly. Approximately, the tasks are completed in 4-5 minutes.
- When the first task is finished, the patient has a rest period of three minutes.
- Then, the second task is performed completing 32 objectives too. But if the first activity was not the control task, then it is execute in this step.

After completion of neurorehabilitation therapy, the trajectories performed by patients at the beginning and at the end are compared to evaluate his/her motor recovery. Fig. 10 shows the trajectories generated by the movements patients in the first and the last session during the

DOI: 10.1002/spe

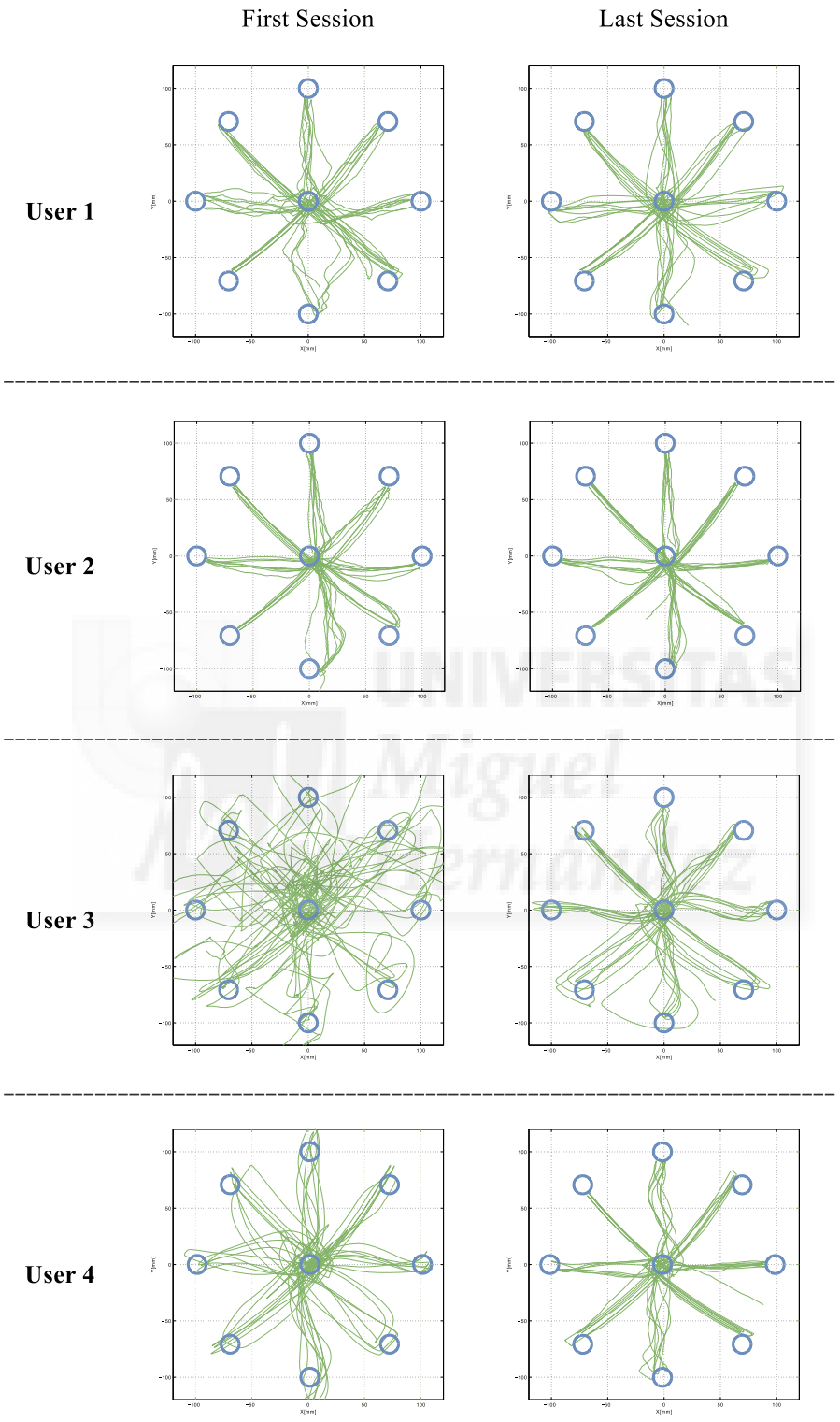


Figure 10. Trajectories performed by all patients in the control task.

DOI: 10.1002/spe

Figure 11. In the image at the left, a post-stroke patient of the study is using PUPArm robot to perform horizontal trajectories and complete a neurorehabilitation therapy in a public hospital. The image at right shows the test of the software platform in a robotic device [29] with three degrees of freedom to perform horizontal and vertical movements.

performance of the control task. The circles indicates the peripheral targets to reach. In all sessions, these targets remain fixed in 10 cm away of the central target. Patients had to begin the movement on the central target and reach the peripheral target performing a straight trajectory. Then, one task objective is completed when the patient reachs a peripheral target and comes back to the center target.

In the last session, all the patients improve their performance as it is shown by the trajectories in Fig. 10. These trajectories are carried out straighter upon at the end of the therapy. This fact suggests patients have improved their motor capabilities due to the movements repetition guided by the robotic device, while they perform the virtual tasks simulated by this software platform. Therefore, the sensorimotor performance of the patients is increased as with other similar studies [26] that use simpler virtual tasks. With this preliminary study, the effectiveness of the software platform can be verified to use it in robot-assisted neurorehabilitation therapies. The obtained results indicates that the using of realistic virtual tasks is equally beneficial than other simpler tasks in this kind of therapy.

In addition, this software platform has been used in a comparative study about the performance of post-stroke patients performing a virtual therapy assisted by a robotic device [27] and to implement an auto-adaptative system based in neuro-fuzzy concepts [28]. Several robotic devices have been used to perform experimental tests (Fig. 11).

6. CONCLUSIONS AND FUTURE WORK

In this paper, a software platform for virtual tasks implementation with auto-adaptable difficulty level regarding the psycho-physiological state of the post-stroke patients has been presented. The system has provided virtual tasks destined for neurorehabilitation therapies of upper limb assisted by robotic devices where the patient is able to interact with the virtual environment controlling the end-effector. This software offers the possibility of simulating any type of scene in a simple, quick and economical manner. The main potential is the ability to add these visual elements and its physical behaviors without adding new code lines in the source code. This fact improves the efficiency, speed and effectiveness of the proposed system to generate different types of tasks in virtual therapies.

The system is responsible for providing targets and interaction movements to the virtual environments. Therefore, it is possible to build multiple virtual tasks without modifying any line of source code. However, the configuration of some task type may eventually require the incorporation for some new code source block to control some special element. Despite this fact, the added blocks can contribute in the design and the control of more elements in the virtual environments and the incorporation of different types of targets. A number of virtual tasks were implemented to check the efficacy of the software platform. These tasks has been used in the robot-assisted neurorehabilitation therapy of four post-stroke patients to demonstrate its effectiveness in this type of treatment.

DOI: 10.1002/spe

In addition, this system reduces the costs arising from the task production process. It generates virtual environments with targets to complete trajectories with the robotic device, adding configuration XML files which are processed by the reading modules to automate the scene organization and the management of the reachable targets. Another important point is the adaptation ability to each patient through the modification of the number of repetitions, time to complete the trajectories, maximum movement length or assistance level and even the adaptation of the difficulty level with the estimation of the patient's emotional state [20]. Therefore, an adaptable neuro-rehabilitation therapy can be provided [30].

Future plans will involve extending the software to add more visualization and performance features such as new types of targets, new methods of visualization of targets or new types of elements to be included. Future work may involve improvements in the rendering quality and physics performance. Moreover, a creation of a graphic interface to implement configuration files can be proposed to facilitate the virtual task prototyping. This interface can be developed following the drag & drop method to a visual generation of these files.

ACKNOWLEDGEMENTS

This work has been supported by the European Commission (ICT-22-2014: Multimodal and Natural computer interaction) through the project AIDE: "Adaptive Multimodal Interfaces to Assist Disabled People in Daily Activities"(Grant agreement no: 645322) and through the project HOMEREHAB: "Development of Development of Robotic Technology for Post-Stroke Home Tele-Rehabilitation - Echord++"(Grant agreement no 601116), by the Ministry of Economy and Competitiveness through the project DPI2015-70415-C2-2-R and The Biomedical Research Networking Center(CIBER).

REFERENCES

1. Castellanos-Pinedo F, Cid-Gala M, Duque P, Ramirez-Moreno JM, Zurdo-Hernandez JM. Acquired brain injury: a proposal for its definition, diagnostic criteria and classification. *Revista de Neurologia* 2012; **54**(6):357–366.
2. Soyuer F, Öztürk A. The effect of spasticity, sense and walking aids in falls of people after chronic stroke. *Disability and Rehabilitation* 2007; **29**(9):679–687, doi:10.1080/09638280600925860.
3. Hyndman D, Ashburn A. People with stroke living in the community: Attention deficits, balance, adl ability and falls. *Disability and Rehabilitation* 2003; **25**(15):817–822, doi:10.1080/0963828031000122221.
4. Khan F, Baguley I, Cameron I. Rehabilitation after traumatic brain injury. *Medical Journal of Australia* 2003; **178**(6):290–295.
5. Kleim J, Jones T. Principles of experience-dependent neural plasticity: Implications for rehabilitation after brain damage. *Journal of Speech, Language, and Hearing Research* 2008; **51**(1), doi:10.1044/1092-4388(2008/018).
6. Bach-y Rita P. Brain plasticity as a basis for recovery of function in humans. *Neuropsychologia* 1990; **28**(6):547–554, doi:10.1016/0028-3932(90)90033-K.
7. Staines W, McIlroy W, Graham S, Black S. Bilateral movement enhances ipsilesional cortical activity in acute stroke: A pilot functional mri study. *Neurology* 2001; **56**(3):401–404.
8. Maciejasz P, Eschweiler J, Gerlach-Hahn K, Jansen-Troy A, Leonhardt S. A survey on robotic devices for upper limb rehabilitation. *Journal of NeuroEngineering and Rehabilitation* 2014; **11**(1), doi:10.1186/1743-0003-11-3.
9. Garcia N, Sabater-Navarro J, Gugliemeli E, Casals A. Trends in rehabilitation robotics. *Medical and Biological Engineering and Computing* 2011; **49**(10):1089–1091, doi:10.1007/s11517-011-0836-x.
10. Saposnik G, Levin M. Virtual reality in stroke rehabilitation: A meta-analysis and implications for clinicians. *Stroke* 2011; **42**(5):1380–1386, doi:10.1161/STROKEAHA.110.605451.
11. Turolla A, Dam M, Ventura L, Tonin P, Agostini M, Zucconi C, Kiper P, Cagnin A, Piron L. Virtual reality for the rehabilitation of the upper limb motor function after stroke: A prospective controlled trial. *Journal of NeuroEngineering and Rehabilitation* 2013; **10**(1), doi:10.1186/1743-0003-10-85.
12. Fluet GG, Deutsch JE. Virtual reality for sensorimotor rehabilitation post-stroke: The promise and current state of the field. *Current Physical Medicine and Rehabilitation Reports* 2013; **1**(1):9–20, doi:10.1007/s40141-013-0005-2.
13. Martín-Ruiz ML, Máximo-Bocanegra N, Luna-Oliva L. A virtual environment to improve the detection of oral-facial malfunction in children with cerebral palsy. *Sensors (Switzerland)* 2016; **16**(4), doi:10.3390/s16040444.
14. García-Betances R, Arredondo Waldmeyer M, Fico G, Cabrera-Umpirrez M. A succinct overview of virtual reality technology use in alzheimer's disease. *Frontiers in Aging Neuroscience* 2015; **7**(APR), doi:10.3389/fnagi.2015.00080.
15. Sabater J, Azorín J, Reinoso O, Neco R, García N. Dynamic virtual environment to test teleoperated systems with time delay communications. *Journal of Robotic Systems* 2005; **22**(4):167–181, doi:10.1002/rob.20057.
16. Gregory J. *Game Engine Architecture*. 2 edn., CRC Press: Boca Raton, Florida, 2014.
17. Simonetti D, Zollo L, Papaleo E, Carpino G, Guglielmelli E. Multimodal adaptive interfaces for 3d robot-mediated upper limb neuro-rehabilitation: An overview of bio-cooperative systems. *Robotics and Autonomous Systems* 2016; **85**:62 – 72, doi:http://dx.doi.org/10.1016/j.robot.2016.08.012.

18. Novak D, Mihelj M, Ziherl J, Olensek A, Munih M. Psychophysiological measurements in a biocooperative feedback loop for upper extremity rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 2011; **19**(4):400–410, doi:10.1109/TNSRE.2011.2160357.
19. Tomas Akenine-Mller NH Eric Haines. *Real-Time Rendering*. 3 edn., CRC Press: Natick, Massachusetts, 2008.
20. Badesa F, Morales R, Garcia-Aracil N, Sabater J, Casals A, Zollo L. Auto-adaptive robot-aided therapy using machine learning techniques. *Computer Methods and Programs in Biomedicine* 2014; **116**(2):123–130, doi: 10.1016/j.cmpb.2013.09.011.
21. Gat E. Artificial intelligence and mobile robots. chap. Three-layer Architectures, MIT Press: Cambridge, MA, USA, 1998; 195–210.
22. Ogre3d: The object-oriented graphics rendering engine accessed on 1 September 2016. URL http://ogre3d.org/.
23. Nvidia physx accessed on 1 September 2016. URL https://developer.nvidia.com/.
24. Lledó L, Badesa F, Almonacid M, Cano-Izquierdo J, Sabater-Navarro J, Fernández E, Garcia-Aracil N. Supervised and dynamic neuro-fuzzy systems to classify physiological responses in robot-assisted neurorehabilitation. *PLoS ONE* 2015; **10**(5), doi:10.1371/journal.pone.0127777.
25. Badesa F, Llinares A, Morales R, Garcia-Aracil N, Sabater J, Perez-Vidal C. Pneumatic planar rehabilitation robot for post-stroke patients. *Biomedical Engineering - Applications, Basis and Communications* 2014; **26**(2), doi:10.4015/S1016237214500252.
26. LLinares A, Badesa FJ, Morales R, Garcia-Aracil N, Sabater J, Fernandez E. Robotic assessment of the influence of age on upper-limb sensorimotor function. *Clinical interventions in aging* 2013; **8**:879, doi:10.2147/CIA.S45900.
27. Lledó LD, Díez JA, Bertomeu-Motos A, Ezquerro S, Badesa FJ, Sabater-Navarro JM, García-Aracil N. A comparative analysis of 2d and 3d tasks for virtual reality therapies based on robotic-assisted neurorehabilitation for post-stroke patients. *Frontiers in Aging Neuroscience* 2016; **8**:205, doi:10.3389/fnagi.2016.00205.
28. Lledó LD, Bertomeu A, Díez J, Badesa FJ, Morales R, Sabater JM, Garcia-Aracil N. Auto-adaptative robot-aided therapy based in 3d virtual tasks controlled by a supervised and dynamic neuro-fuzzy system. *International Journal of Interactive Multimedia and Artificial Intelligence* 03/2015 2015; **3**(2):63–68, doi:10.9781/ijimai.2015.328.
29. Díez JA, Catalán JM, Lledó LD, Badesa FJ, Garcia-Aracil N. Multimodal robotic system for upper-limb rehabilitation in physical environment. *Advances in Mechanical Engineering* 2016; **8**(9), doi:10.1177/1687814016670282.
30. Morales R, Badesa F, Garcia-Aracil N, Perez-Vidal C, Sabater J, Papaleo E, Salerno A, Zollo L, Guglielmelli E. Patient-tailored assistance: A new concept of assistive robotic device that adapts to individual users. *IEEE Robotics and Automation Magazine* 2014; **21**(3):123–133, doi:10.1109/MRA.2014.2304051.