



Universidad de Valladolid

Gestión y análisis de una base de datos para una aplicación de ingeniería biomédica.

TRABAJO DE FINAL DE GRADO

Miguel Zatarain Valles

02/09/2015



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Electrónica Industrial y Automática

**Gestión y análisis de una base de datos para una
aplicación de ingeniería biomédica.**

Autor:

Zatarain Valles, Miguel

Tutor:

**González Sánchez, José Luis
Departamento de Ingeniería de
Sistemas y Automática.**

Valladolid, Septiembre 2015.



AGRADECIMIENTOS:

En primer lugar quisiera agradecer a Juan Carlos Fraile y a Javier Pérez Turiel por haberme ofrecido la oportunidad de trabajar en este proyecto y por el seguimiento y la ayuda que me han aportado, y también a José Luis González Sánchez.

A mis compañeros de Cartif, Pablo, Alex y Rubén por motivarme cada día y ayudarme en los problemas que ido teniendo y a Mar y Javier porque al trabajar con ellos acababa con una sonrisa en la cara.

A mis padres, por haber estado conmigo durante todo el camino, por orientarme, por darme sus consejos y haber querido siempre lo mejor para mí. Y por supuesto, a mi hermana por aguantarme todos estos años, a mis abuelos y al resto de mi familia.

A todos los profesores, entrenadores y gente que me ha educado y enseñado desde el colegio a la universidad, por todo lo que he aprendido gracias a vosotros.

A todos los compañeros y amigos que he conocido en la universidad, en especial a Laura y a Dani, porque sin todos vosotros y vuestra ayuda no estaría escribiendo estas líneas.

A toda la gente de BEST Valladolid, por hacer que me esfuerce y darme tantos buenos momentos.

A mis amigos y amigas que son como hermanos y que siempre están ahí tanto a las buenas como a las malas, es impensable haber llegado hasta aquí sin vosotros.

Gracias.





RESUMEN:

Análisis offline gráfico y estadístico de la base de datos generada por el sistema robotizado háptico Physiobot para ingeniería biomédica, mediante la adaptación de métricas ya existentes y creación de nuevas métricas para facilitar la comprensión de las terapias médicas realizadas a los pacientes, se analizan las sesiones de forma individual para saber cómo se han desarrollado cada una así como la evolución que ha tenido el paciente en varias sesiones, también se ha elaborado la interfaz gráfica (empleando C#) con el usuario para que pueda acceder fácilmente a las gráficas, las cuales se generarán con el lenguaje de programación R, que se comunica mediante la interfaz con el usuario.

Palabras clave: ingeniería biomedica, métricas, base de datos, R, C#, interfaz

ABSTRACT:

Offline graphic and statistical analysis of the database generated by the haptic robotic system Physiobot for biomedical engineering by adapting existing metrics and creating new metrics to facilitate understanding of medical therapies done by the patients, the sessions are analyzed individually to see how they have developed and the evolution that has taken the patient in several sessions, it has also been made the graphical user interface (using C #) with the user so he can easily access the graphics, which are generate with the R programming language, that communicates with user by the interface.

Keywords: biomedical engineering, metrics, database, R, C #, interface





ÍNDICE

| | |
|---|-----|
| 1.INTRODUCCIÓN Y OBJETIVOS..... | 7 |
| 1.1. INTRODUCCIÓN..... | 9 |
| 1.1.1. HARDWARE Y SOFTWARE DEL ROBOT PHYSIOBOT | 11 |
| 1.1.2 SEGURIDAD | 12 |
| 1.1.3 TIPO DE TERAPIAS..... | 13 |
| 1.1.4 ANÁLISIS OFFLINE DE DATOS..... | 15 |
| 1.1.5. EJES DEL ROBOT PHYSIOBOT Y DE LAS TERAPIAS..... | 16 |
| 1.2. OBJETIVOS | 17 |
| 2. MEMORIA TÉCNICA..... | 19 |
| 2.1. CONSULTA DE DATOS | 21 |
| 2.1.1 TABLAS..... | 22 |
| 2.1.2 CONSULTA DE DATOS CON SQL..... | 27 |
| 2.2 ANÁLISIS DE DATOS | 28 |
| 2.2.1 MÉTRICAS DE TERAPIAS INDIVIDUALES..... | 30 |
| 2.2.2. METRICAS DE EVOLUCIÓN DE TERAPIAS | 50 |
| 2.3. INTERFAZ Y SOFTWARE DE GESTIÓN DE INFORMACIÓN | 63 |
| 2.3.1 ACTUALIZACIÓN DE FICHEROS..... | 63 |
| 2.3.2 GENERACIÓN DE INFORMES EN PDF | 67 |
| 2.3.3 INTERACCIÓN ENTRE R Y WPF | 70 |
| 2.3.4 INTERFAZ GRÁFICA DEL SOFTWARE DEL ROBOT PHYSIOBOT | 72 |
| 3. CONCLUSIONES | 95 |
| 4. FUTURAS LÍNEAS DE TRABAJO..... | 99 |
| 5.BIBLIOGRAFÍA..... | 103 |
| 6.ANEXOS | 107 |





1.INTRODUCCIÓN Y OBJETIVOS.





1.1. INTRODUCCIÓN

Los pacientes con ictus cerebral tienden a sufrir parálisis en su movilidad, ya sea en las extremidades superiores o inferiores, esta movilidad y/o pérdida de la coordinación se traduce directamente en una disminución de la independencia de sus movimientos y calidad de vida. Aquellos pacientes que se insertan rápidamente en programas de neuro-rehabilitación, pueden recuperar algunas de las capacidades perdidas que solían tener y por lo tanto la mejora de su calidad de vida. [1]

El uso de dispositivos robóticos para fines de rehabilitación ha demostrado ser una excelente herramienta que ayuda en la administración de terapia física para pacientes que sufrieron lesiones y enfermedades neurológicas. Varios estudios han demostrado que la terapia de brazo tiene efectos positivos sobre el progreso de rehabilitación de pacientes con accidente cerebrovascular [2].

El objetivo de algoritmos de control de la terapia robótica es conducir dispositivos robóticos diseñados para ejercicios de rehabilitación, de modo que los ejercicios seleccionados a realizar por el participante provocar plasticidad motora, y por lo tanto mejorar la recuperación motora.

Hay muchos ejemplos de plataformas modernas de rehabilitación [3] [4] [5] [6]. La mayoría de ellas incluyen interfaces multimodales mejorados con realidad virtual interactiva, robots hápticos de alto rendimiento y exoesqueletos.

El uso de las tecnologías robóticas en neuro-rehabilitación ha sido objeto de gran desarrollo en los últimos 20 años [7]. El uso de sistemas de robótica en este campo [5] nos proporciona terapias más controladas, interactivas y orientadas a tareas. También nos proporciona nuevas herramientas para medir el progreso de los pacientes. Varios estudios [8] [9] se han publicado con el fin de demostrar la eficacia de la formación asistida por robot en pacientes con lesión cerebral traumática. Casi todos los estudios [10] [11] muestran un crecimiento significativo en la función motora del paciente. La conclusión de estos estudios es que las terapias robóticas pueden ayudar a la recuperación de la función motora la extremidad superior, que se convierten en una mejora en el control motor, por lo que se puede concluir que es mejor que la terapia tradicional.

Physiobot[12], es un robot háptico para neuro-rehabilitación de miembros superiores que permite administrar terapias orientadas a tareas, impartidas en forma de “juegos virtuales” que aumentan la motivación de los pacientes. Este factor es clave en el proceso de reorganización neuronal necesario para volver a adquirir las capacidades motoras perdidas a raíz del accidente. Una pantalla de ordenador proporciona realimentación visual al paciente, tanto de la posición del objetivo, como de la posición real del robot que maneja. Un marcador



con la puntuación obtenida proporciona al paciente información inmediata sobre su desempeño con el objetivo de aumentar su motivación en ciertas terapias.

También permite al médico o fisioterapeuta programar tareas específicas para cada paciente, en función de sus necesidades. El proceso de grabación es tan simple como apretar un botón y mover el conjunto robot / miembro afectado a lo largo de la trayectoria deseada para la tarea.



Fig. 1 Robot Physiobot

Physiobot proporciona terapias activas de rehabilitación orientadas a tareas, basadas en las necesidades y capacidades de cada individuo. Estas terapias son de tipo activo, aplicando estrategias de control háptico basadas en el paradigma “assisted as needed”. Esto supone que este dispositivo robotizado de rehabilitación asiste al paciente sólo cuando lo necesita, promoviendo la intencionalidad de movimiento del sujeto, lo cual es crítico para el reaprendizaje motor del miembro afectado.

1.1.1. HARDWARE Y SOFTWARE DEL ROBOT PHYSIOBOT

La Figura 2 muestra un diagrama de bloques, que describe toda la plataforma Phisiobot. En resumen, un robot háptico equipado con un sensor de fuerza montado en el efector del extremo se utiliza como centro de la plataforma de hardware para el sistema. Existen varios tipos de flujo de datos a través de una red IP para intercambiar información de los datos de movimiento, que nos permiten, recopilar datos en tiempo real a diferentes frecuencias de muestreo. Toda la información en bruto y procesada se almacena en una base de datos para su posterior análisis e investigación. Varios datos fisiológicos y dinámicos se registran y almacenan en línea como la serie temporal para el control en línea y análisis offline. El Controlador que vamos a emplear para la obtención de datos de los sensores va a ser una FPGA (del inglés Field Programmable Gate Array).

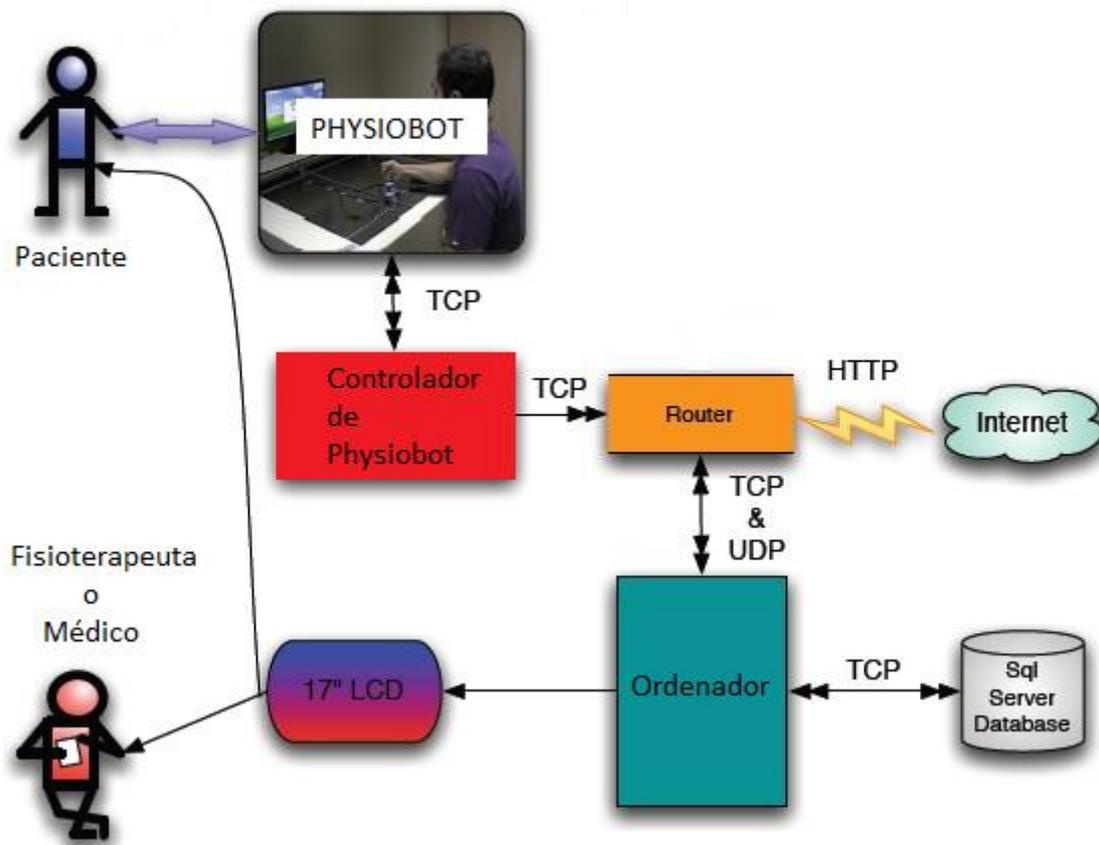


Fig. 2 Arquitectura del robot Phisiobot.

Physiobot (ver Figura 3), fue concebido para ser un robot intrínsecamente seguro, buscando tener una dinámica transparente y fácil de manejar.

La estructura mecánica cuenta con las siguientes características:

- El tensor de inercia isotrópico se puede lograr debido a la mecánica similar en ambos ejes.
- El diseño más simple, hace cinemática directa e inversa más sencilla de calcular, por lo tanto, la reducción de los costos computacionales.
- Fuerzas debidas a la gravedad, están estructuralmente compensadas, lo que no hay necesidad de incluir un modelo de compensación de gravedad.

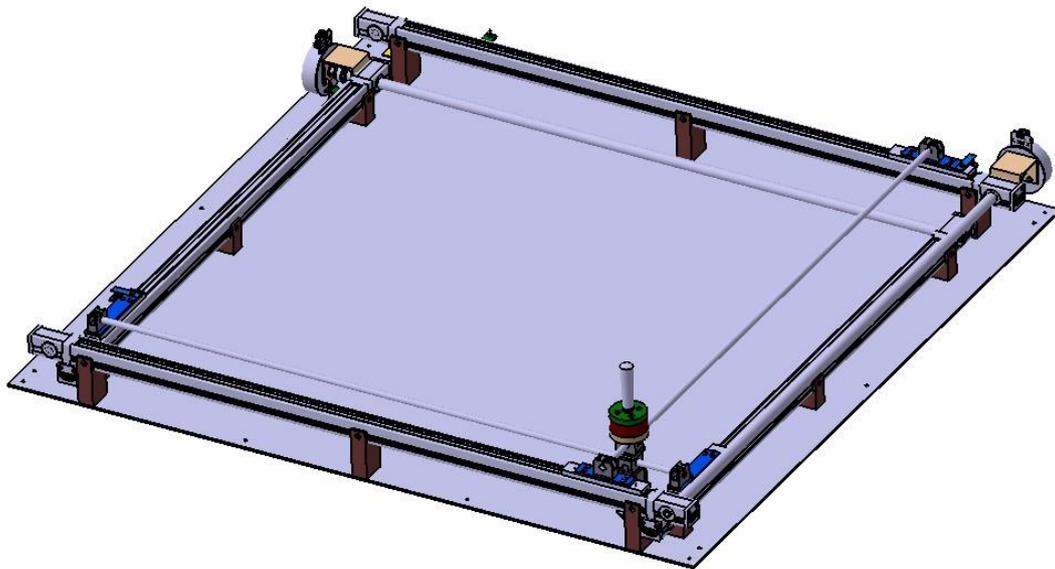


Fig. 3. Plataforma de soporte de Physiobot

1.1.2 SEGURIDAD

- **Seguridad del Hardware:** Physiobot cuenta con seguridad redundante por medio de hardware y software. Hay 4 finales de carrera (véase el Fig. 3, los sensores negros sobre una placa azul en los extremos de cada carril) que cortan la corriente del motor



cuando se acerca a cada parada mecánica de cada carril. Esto ayuda no sólo a la seguridad, sino para evitar la degradación de hardware a través del tiempo por los posibles impactos.

- **Seguridad del software:** Junto con la seguridad de hardware, se han implementado paredes viscoelásticas virtuales cerca de los bordes de los límites de robots, con el fin de prevenir los impactos.

1.1.3 TIPO DE TERAPIAS

Con el robot Physiobot se puede realizar una serie de terapias que son sesiones que realiza el paciente con el robot, inicialmente, las terapias tenían un duración de tiempo variable pero finalmente terminaron teniendo una duración de cinco minutos.

Existe un conjunto de terapias que ya están desarrolladas y las que están registrando datos actualmente son las siguientes:

- Activa Asistida
- Pasiva Guiada

Vamos a analizar estas terapias más en profundidad

1.1.3.1 TERAPIA TIPO 1: ACTIVA ASISTIDA

El objetivo de esta terapia es que el paciente mueva la mano hasta alcanzar el ratón que se mueve en la pantalla.

Para “capturar” al ratón, el paciente deberá intentar que la mano y el ratón ocupen el mismo lugar. El ratón se mueve de manera autónoma y errática de forma pseudoaleatoria, huyendo rápidamente cuando es alcanzado. Decimos que el movimiento es pseudoaleatorio debido a que está condicionado a la dificultad de la terapia, que va de 1 a 4 siendo 1 la de menor dificultad y 4 la de mayor dificultad. En realidad nunca se captura al ratón, sino que se ganan puntos por el tiempo que se logra hacer coincidir ambos objetos, a mayor tiempo que coincidan, mayor será la puntuación.

A esta terapia se la va a denominar también como terapia 1. Cuando se captura el ratón la mano cambia a un color rojo.

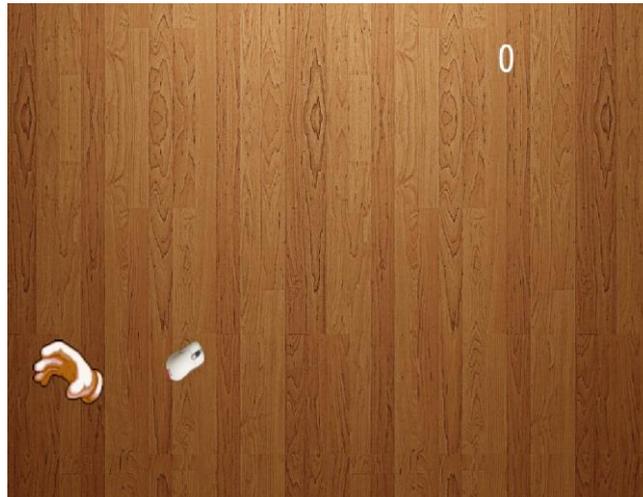


Fig. 4 Terapia Activa Asistida

1.1.3.2. TERAPIA TIPO 2: PASIVO GUIADA

En esta el terapeuta realiza y graba una trayectoria desde un punto inicial a uno final. El paciente deberá repetir la terapia grabada previamente y cuando llegue al punto final deberá realizar la trayectoria inversa hasta llegar al punto inicial, repetirá este proceso varias veces. Todo este proceso para el paciente consiste en perseguir a una mano que se muestra en la pantalla, el robot le asistirá en la medida que sea necesario debido al control háptico. Se va a denominar a esta terapia como terapia 2.

La mano azul será la que maneje el paciente mientras que la mano blanca será la que el paciente tendrá que seguir.

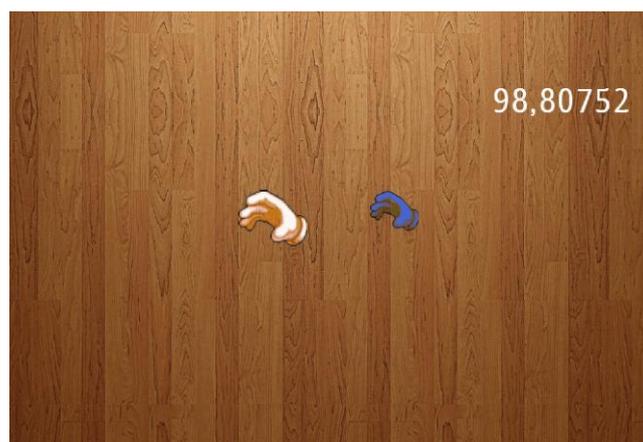


Fig. 5 Terapia Pasivo Guiada



1.1.4 ANÁLISIS OFFLINE DE DATOS

Debido a que el paciente empleando el Physiobot produce una gran cantidad de información, se genera una amplia base de datos, vamos a necesitar el uso de programas y filtros para poder analizarlos.

Todos los datos que se generan en el robot se trasladan a un servidor, este servidor funciona con SQL Server 2008, para analizar estos datos que hemos almacenado previamente vamos a emplear el lenguaje de programación R, [13][14], R es un conjunto integrado de programas para manipulación de datos, cálculo y gráficos.

Con R vamos a realizar también métricas como conjunto del análisis de datos. Para mostrar estas métricas hemos implementado una interfaz en Visual Studio 2010 con C#, de esta forma la interacción entre el fisioterapeuta o médico a la hora de consultar los datos, lo pueden hacer directamente sin la necesidad de analizar los datos ellos mismos o tener que emplear el lenguaje de programación R.

Las métricas que vamos a utilizar las vamos a dividir en dos grupos, un grupo será el de gráficas de evolución que se encargarán de medir ciertos parámetros en función de varias terapias realizadas y otro grupo de análisis de terapia, que serán encargadas de analizar las terapias individualmente.

La interfaz permitirá analizar una terapia, comparar dos terapias distintas, ver la evolución de un paciente, comparar dos evoluciones, además permitirá generar un pdf en el cual se vea los resultados obtenidos por un paciente.

Existen tres versiones del robot Physiobot, que tienen diferentes configuraciones y bases de datos así que tendremos que variar los parámetros y partes del software, por ello para analizar el robot Physiobot depende que versión empleemos. Las distintas versiones y su ubicación son:

- Versión 1: Hospital Beata María Ana de Madrid.
- Versión 1: Hospital Clínico Universitario de Valladolid.
- Versión 2: Centro Hospitalario Benito Menni de Valladolid.
- Versión 3: Fundación IDES de Zamora.

El método de trabajo seguido es coger una base de datos, en nuestro caso “Beata María” de la primera versión y luego ampliar el programa y configurarlo a las otras dos versiones. Estos tres robots y sus versiones se han desarrollado en el marco del proyecto E2REBOT de la fundación Cartif.



1.1.5. EJES DEL ROBOT PHYSIOBOT Y DE LAS TERAPIAS.

Las terapias generadas por el robot guardan datos de su ubicación, velocidad y fuerzas entre otros parámetros en la base de datos y los hace con respecto a dos ejes, el eje x y el eje y, y guarda los datos generados tanto de la pantalla del ordenador como de los la plataforma del Physiobot.

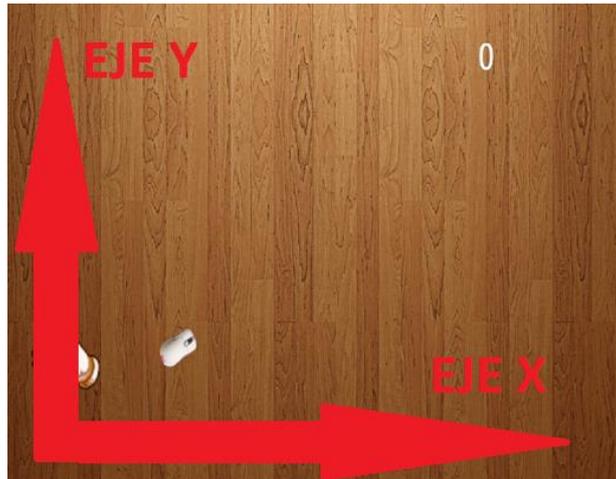


Fig. 6 Ejes en la pantalla.

Podemos ver como los datos generados en la pantalla como vemos en la figura 6 se guardan en bajo estos ejes en la base de datos.

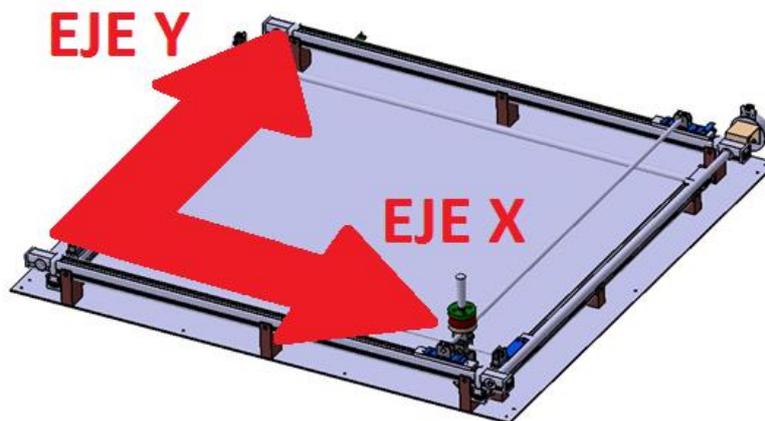


Fig. 7. Ejes en la plataforma del Physiobot.

Podemos ver como los datos generados en la plataforma como vemos en la figura 7 se guardan en bajo estos ejes en la base de datos.



1.2. OBJETIVOS

Debido a que el robot Physiobot genera una gran cantidad de datos lo que buscamos es poder analizarlos y para ello tendremos que emplear el lenguaje de programación R para crear gráficas que podamos emplear para analizar los datos visualmente, y para que acceder a estos datos sea más simple emplearemos una interfaz gráfica, para ello deberemos de cumplir los siguientes objetivos:

- Aprender el funcionamiento del robot Physiobot, así como la forma de guardar la información en la base de datos.
- Buscar información de las métricas de robots similares y adaptarlas al robot Physiobot de ser posible.
- Crear nuevas métricas comprensibles para médicos y fisioterapeutas que se adecuen a los datos generados.
- Desarrollar un procedimiento para poder acceder a la base de datos del robot de forma remota y aprender a realizar consultas SQL para poder manejar los datos que tenemos almacenados.
- Graficar estas métricas de manera automática con el lenguaje de programación R y comunicar de forma automática R con la base de datos.
- Implementar los scripts de R dónde realizamos las gráficas para que funcionen en el mínimo tiempo posible.
- Lograr una comunicación entre C# y R.
- Implementar con C# empleando Visual Studio, una interfaz gráfica para que el usuario pueda acceder a las gráficas realizadas con R de la forma más simple y práctica posible.





2. MEMORIA TÉCNICA





2.1. CONSULTA DE DATOS

Para la realización de las métricas hemos empleado el lenguaje de programación R[13][14], que es un conjunto integrado de programas para manipulación de datos, cálculo y gráficos.

Para realizar cada métrica hemos creado diversos scripts independientes en R, para su realización nos hemos ayudado del entorno gráfico RGui, utilizamos un encabezado de “Inicialización”, el cual incluye las librerías y el código necesario para acceder a la base de datos, que está en SQL Server, y nos permite comunicarnos ella.

Vamos a analizar la base de datos de “BeataMaria”, aunque como se expuso en la introducción también tenemos otras dos bases de datos, las tablas que se emplean en ellas son exactamente las mismas que para “BeataMaría”, y las vamos a ver en el siguiente punto.

- **Inicialización**

Es el encabezado de todos los scripts, con el que nos encargamos de conectar con la base de datos para poder tener acceso a los datos, para ello empleamos la librería “RODBC”[15]. Esta conexión la llevamos a cabo a través de una DN (Domain Name) específica, y generamos un vector con el nombre de usuario, la contraseña y la propia DN. Este comando nos va a permitir obtener información de las tablas que tenemos almacenadas en SQL Server, de forma que recibiremos tanto los datos como el encabezado de las columnas, para nuestra graficación y análisis de métricas. Para analizar los datos de la base de datos “BeataMaria” vamos a emplear dos tablas.

En caso de utilizar otra base de datos distinta a “Beata María” se necesitaría cambiar en esta parte de código la DN, como el servidor que empleamos es el mismo no hace falta cambiarlo, por ejemplo:

```
con<-odbcDriverConnect( "driver=SQL Server;database=BeataMaria;server=Nombre  
Equipo//BBD; uid=*****; pwd=*****" )  
  
con<-odbcDriverConnect( "driver=SQL Server;database=Nueva ;server= Nombre Equipo  
//Nuevo BBD; uid=*****; pwd=*****" )
```

También puede ocurrir que instalemos el programa en otro ordenador, y solo ciertos ordenadores tienen permisos para acceder a la base de datos, de ser así deberíamos cambiar:



```
con<-odbcDriverConnect( "driver=SQL Server;database=Nueva ;server= Nuevo Nombre Equipo  
//BBD; uid=*****; pwd=*****" )
```

2.1.1 TABLAS

La información que almacenamos en la base de datos está ordenada por tablas, de forma que para poder acceder a todos los datos vamos a tener que acceder a distintas tablas.

Esto implica que todos los datos no están en una única tabla, de forma que para poder acceder a los datos que queremos puede que tengamos que emplear más de una tabla, algunos datos de unas tablas están repetidos en otras, lo cual nos ayudará a concatenar estas tablas en caso de necesidad.

Las tablas están divididas en filas y columnas, pero solo las columnas tienen encabezado que nos van a indicar con qué tipo de datos estamos tratando.

A continuación vamos a analizar todas las tablas que empleamos para poder ver la información que vamos a tratar.

- **Tabla de terapias:**

Esta tabla nos va a mostrar quién, cuando, con que dificultad y que tipo de terapia se ha realiza en todas las terapias que se han realizado.

| | IdTerapia | Fecha | IdPaciente | Dificultad | TipoTerapia |
|--|-----------|-------------------|------------|------------|-------------|
| | 44 | 2014-06-02 15:... | 2 | 1 | 1 |
| | 45 | 2014-06-02 15:... | 2 | 1 | 1 |
| | 46 | 2014-06-02 15:... | 2 | 1 | 1 |
| | 47 | 2014-06-02 15:... | 2 | 1 | 1 |
| | 48 | 2014-06-02 15:... | 1 | 3 | 1 |
| | 49 | 2014-06-02 15:... | 1 | 3 | 1 |
| | 50 | 2014-06-02 16:... | 1 | 3 | 1 |
| | 51 | 2014-06-02 16:... | 1 | 3 | 1 |
| | 52 | 2014-06-02 16:... | 1 | 1 | 1 |
| | 53 | 2014-06-02 16:... | 1 | 3 | 1 |
| | 54 | 2014-06-02 16:... | 1 | 1 | 1 |

Fig. 8. Tabla de terapias.

Sus elementos son:

IdTerapia: Nos dice la terapia que realiza, cada vez que el paciente realiza una sesión con el robot, es una terapia.

Fecha: Nos muestra la fecha y la hora en la que se inició la terapia.

IdPaciente: Es un número asociado a un paciente para garantizar su anonimato.

Dificultad: La dificultad que tiene la terapia cuando es una terapia de tipo 1, en nuestro caso puede ser 1,2 o 3.

TipoTerapia: La terapia que sigue, si es activa asistiva es un 1 y si es pasivo guiada es un 2.

- **Tabla de muestras:**

Esta tabla nos registra todos los valores de los sensores, el tiempo, el marcador, y todas las terapias realizadas, una terapia realizada correctamente tiene alrededor de 8000 filas, en total esta tabla con más de mil terapias realizada, tiene más de seis millones de filas. Es la tabla más grande que tenemos en nuestra base de datos.

| IdTerapia | IdMuestra | Tiempo | PosicionX | PosicionY | PuntoDeseadoX | PuntoDeseadoY | Fuer... | FuerzaY | FuerzaAsistiva | Marcador | VelocidadX | VelocidadY |
|-----------|-----------|------------------|-----------|-----------|---------------|---------------|---------|----------|----------------|----------|------------|------------|
| 625 | 3487574 | 00:00:20.0201451 | 338,00 | 168,00 | 332,00 | 407,00 | 0,00 | 8802,00 | 0,00 | NULL | 0,00 | 215,00 |
| 625 | 3487575 | 00:00:20.0541470 | 338,00 | 177,00 | 332,00 | 399,00 | 0,00 | 8222,00 | 0,00 | NULL | 0,00 | 241,00 |
| 625 | 3487576 | 00:00:20.0871489 | 338,00 | 182,00 | 332,00 | 396,00 | 0,00 | 8814,00 | 0,00 | NULL | 0,00 | 241,00 |
| 625 | 3487577 | 00:00:20.1201508 | 338,00 | 187,00 | 332,00 | 392,00 | 0,00 | 9760,00 | 0,00 | NULL | 0,00 | 253,00 |
| 625 | 3487578 | 00:00:20.1541527 | 338,00 | 203,00 | 332,00 | 380,00 | 0,00 | 10419... | 0,00 | NULL | 0,00 | 279,00 |
| 625 | 3487579 | 00:00:20.1871546 | 338,00 | 209,00 | 332,00 | 376,00 | 0,00 | 10010... | 0,00 | NULL | 0,00 | 291,00 |
| 625 | 3487580 | 00:00:20.2201565 | 338,00 | 214,00 | 332,00 | 372,00 | 0,00 | 9162,00 | 0,00 | NULL | 0,00 | 304,00 |
| 625 | 3487581 | 00:00:20.2531584 | 338,00 | 227,00 | 332,00 | 364,00 | 0,00 | 10504... | 0,00 | NULL | 0,00 | 317,00 |
| 625 | 3487582 | 00:00:20.2871603 | 338,00 | 233,00 | 332,00 | 360,00 | 0,00 | 9784,00 | 0,00 | NULL | 0,00 | 304,00 |
| 625 | 3487583 | 00:00:20.3201622 | 338,00 | 252,00 | 332,00 | 348,00 | 0,00 | 9693,00 | 0,00 | NULL | 0,00 | 317,00 |
| 625 | 3487584 | 00:00:20.3531641 | 338,00 | 258,00 | 332,00 | 344,00 | 0,00 | 9369,00 | 0,00 | NULL | 0,00 | 317,00 |
| 625 | 3487585 | 00:00:20.3871661 | 338,00 | 264,00 | 332,00 | 340,00 | 0,00 | 9339,00 | 0,00 | NULL | 0,00 | 329,00 |
| 625 | 3487586 | 00:00:20.4201680 | 338,00 | 283,00 | 332,00 | 327,00 | 0,00 | 8649,00 | 0,00 | NULL | 0,00 | 304,00 |
| 625 | 3487587 | 00:00:20.4531698 | 338,00 | 295,00 | 332,00 | 319,00 | 0,00 | 8918,00 | 0,00 | NULL | 0,00 | 291,00 |
| 625 | 3487588 | 00:00:20.4871718 | 338,00 | 301,00 | 332,00 | 315,00 | 0,00 | 8527,00 | 0,00 | NULL | 0,00 | 291,00 |
| 625 | 3487589 | 00:00:20.5201737 | 338,00 | 307,00 | 332,00 | 310,00 | 0,00 | 8454,00 | 0,00 | NULL | 0,00 | 279,00 |
| 625 | 3487590 | 00:00:20.5531756 | 338,00 | 323,00 | 332,00 | 297,00 | 0,00 | 7904,00 | 0,00 | NULL | 0,00 | 266,00 |
| 625 | 3487591 | 00:00:20.5871775 | 338,00 | 328,00 | 332,00 | 293,00 | 0,00 | 7575,00 | 0,00 | NULL | 0,00 | 241,00 |

Fig. 9. Tabla de muestras.

IdTerapia: Nos dice la terapia que realiza, cada vez que el paciente realiza una sesión con el robot, es una terapia, este número se repite siempre que la terapia es la misma.



IdMuestra: Nos indica un número de cada registro que se ha hecho en orden, en esta tabla, por ejemplo en la figura 9 podemos ver como es la fila 3.487.591 que registra.

Tiempo: Empieza a 0 cuando la terapia se inicia, y es el tiempo en el que se registra la información de los sensores. No toma los valores de forma periódica, esto es debido a que a pesar de que la FPGA nos envía datos que recoge de los sensores cada 50ms, el ordenador tiene que hacer una petición para saber qué tiempo es, y dependiendo de los ciclos, ese tiempo varía, aunque suele tender a 0.03 segundos.

PosiciónX: Nos da la posición del robot en el eje de x en milímetros.

PosiciónY: Nos da la posición del robot en el eje de y en milímetros.

PuntoDeseadoX: Nos da un valor referenciado a la posición que tiene el objetivo en la pantalla en el eje de las x, más adelante explico cómo lo transformamos en un valor con el que podamos trabajar.

PuntoDeseadoY: Nos da un valor referenciado a la posición que tiene el objetivo en la pantalla en el eje de las y, más adelante explico cómo lo transformamos en un valor con el que podamos trabajar.

FuerzaX: Nos da el valor de fuerza que realiza el paciente en Newtons en el eje de las x.

FuerzaY: Nos da el valor de fuerza que realiza el paciente en Newtons en el eje de las y.

FuerzaAsistiva: Nos da una constante k que nos permite calcular la fuerza asistiva que estamos empleando.

Marcador: En el tipo de terapia 1, el último resultado del marcador nos indica los puntos que ha conseguido el paciente realizando la terapia, se almacena este resultado en la última columna de cada terapia, a partir de la terapia 667 se ha producido un error en el robot y a dejado de registrarlos.

VelocidadX: Nos da el valor de la velocidad que realiza el paciente en el eje de las x en mm/s.

VelocidadY: Nos da el valor de la velocidad que realiza el paciente en el eje de las y en mm/s.

También podemos usar la siguiente tabla de datos para consultar detalles del paciente como el nombre o la lesión que sufre, pero de momento como en “Beata María” no tenemos nombres, prescindimos de utilizarla, esta tabla no se usa en ningún momento en los programas, pero podría usarse a la hora de imprimir los datos para que el medico pudiese verlos.



Es decir se usa un *idpaciente*, con esta identidad del paciente que es solo un número que se asocia con el nombre del paciente, de forma que se podría enlazar las tablas de forma que no se operase con el nombre del paciente salvo cuando el medico hiciese específicamente una consulta de su nombre y datos personales.

- **Tabla de pacientes:**

Tal y como hemos comentado es una tabla que no empleamos debido a ocultar la identidad del paciente, pero q se podría emplear para q aun sin saber la identidad nosotros enlazarla para que exclusivamente el médico pudiese saber de quién se trata.

| | IdPaciente | Nombre | Apellidos | Edad | Discapacidad | sexo |
|--|------------|----------|-----------|------|----------------------|--------|
| | 1 | Prueba 1 | Prueba 1 | 1 | Paciente de pru... | Hombre |
| | 2 | | | 25 | hemiplegia | Hombre |
| | 3 | | | 25 | paciente de pru... | Mujer |
| | 4 | | | 30 | hemiparesia der... | Hombre |
| | 5 | | | 55 | hemiparesia izqu... | Hombre |
| | 6 | | | 80 | hemiparesia dcha | Hombre |
| | 7 | | | 54 | Hemiplejia izquie... | Mujer |
| | 8 | | | 13 | Hemiplejia derecha | Hombre |
| | 9 | | | 62 | Paresia miembro... | Hombre |
| | 10 | | | 35 | Hemiplejia derec... | Hombre |
| | 11 | | | 26 | HEMIPLEJIA DE... | Mujer |
| | 12 | | | 52 | HEMIPARESIA I... | Hombre |
| | 13 | | | 15 | Hemiparesia Izq... | Hombre |
| | 14 | | | 85 | HEMIPARESIA D... | Hombre |

Fig. 10. Tabla de pacientes.

IdPaciente: Es un número asociado a un paciente para garantizar su anonimato.

Nombre: Es el nombre del paciente.

Apellidos: Son los apellidos del paciente.

Edad: Es la edad del paciente en años.

Discapacidad: Es la lesión que tiene el paciente.

Sexo: Es el sexo del paciente (Hombre o Mujer).



- **Transformaciones y filtrado de variables.**

Dado que tenemos variables en la base de datos que no corresponden con las medidas que vamos a utilizar necesitamos transformar estos datos en datos útiles, en nuestro caso, para Beata María son los siguientes:

Filas iniciales: Al menos las 4 primeras filas de la tabla Muestra son valores residuales así que les eliminaremos antes de empezar a trabajar.

Número de datos: Al trabajar con gráficas de evolución del paciente no todas las terapias son válidas, porque en algunas el número de datos tomados es muy pequeño, la mayoría de las veces es porque el paciente no inicia bien la terapia y cierra esa terapia y abren una nueva, las terapias que hemos tomado como válidas son aquellas que registran un número mínimo de 4500 filas, siendo lo normal para una terapia registrar del orden de las 8000 filas.

Puntos Deseados: Dado que los puntos deseados son los que se muestran en la pantalla las medidas no están en milímetros y no están calibradas, para pasarlos a milímetros debido a su diferente calibración en x sumaremos 1 a los datos Puntos Deseados en X, mientras que en y restaremos a 600 los Puntos Deseados en Y, que tenemos en la base de datos, con esto las medidas que utilizaremos ahora pasarán a estar en milímetros.

Fuerzas en Terapia 2: Las fuerzas en la terapia 2 tanto en x, como en y se guardan en $\text{kg}\cdot\text{mm}/\text{s}^2$ en lugar de en Newton, por ello cuando necesitemos las fuerzas en la terapia 2 vamos a necesitar dividir las entre 1000 para tener $\text{kg}\cdot\text{m}/\text{s}^2 = \text{Newtons}$.

Tiempo: El tiempo bien en un formato 00:00:00.0000000, existen funciones y librerías que te pasan el tiempo de 00:00:00.0 a segundos pero no funcionan con tantos decimales, y nosotros lo necesitamos, así que lo que hacemos es pasar cambiar el formato que tiene que es tipo "form" a carácter y utilizamos los dos puntos como separación de términos de forma que tendremos "a":"b":"c", pasamos "a","b" y "c" a numérico y luego realizamos la operación $3600*a+60*b+c$.

Tiempo: No registra el tiempo en de forma periódica, como ya explicamos antes, pero a mayores, a veces la petición para registrar los datos no llega de manera correcta y pueden pasar hasta 15 segundos sin que el robot registre los datos en la base de datos.

Si cambiamos a otro robot del mismo tipo habría que volver a calibrar los puntos deseados.



2.1.2 CONSULTA DE DATOS CON SQL

Para poder acceder a todas estas tablas necesitamos hacer consultas SQL, en R podemos hacerlo de forma automática, el lenguaje de desarrollo que empleamos es Transact-SQL (TSQL), que es una implementación del estándar ANSI del lenguaje SQL.

Una consulta SQL estándar de las que vamos a utilizar para ver una tabla de la base de datos puede ser:

```
SELECT    IdTerapia, Fecha, IdPaciente, IdExperimento, Dificultad, TipoTerapia
FROM      TerapiaBM
```

En esta consulta lo que hacemos es pedir a través del comando FROM que escoja la tabla TerapiaBM seleccionamos mediante el comando SELECT las columnas IdTerapia, Fecha, IdPaciente, IdExperimento, Dificultad, TipoTerapia.

Pero en algunos casos debemos hacer consultas más específicas realizando un filtrado en los datos que queremos obtener:

```
SELECT    IdTerapia, IdPaciente, TipoTerapia
FROM      TerapiaBM
WHERE     (IdPaciente = 7) AND (IdTerapia <= 1000) AND (IdTerapia >= 200) AND
(TipoTerapia = 2)
```

En esta consulta es parecida a la anterior, seleccionamos la tabla y las columnas, pero a mayores con el comando WHERE filtramos los datos que queremos tener, en este caso queremos solo datos del paciente con el número de identidad 7, con las terapias que haya realizado entre la 1000 y la 200 y que sean de tipo 2.

Otro caso que vamos a emplear es el de unir dos tablas, por ejemplo si necesitamos saber en la tabla de muestras que tipo de terapia está realizando necesitamos unirla con la tabla TerapiaBM, para ello podemos emplear un código como este:



```
SELECT      m.IdMuestra, m.PosicionX, m.PosicionY, m.PuntoDeseadoX,
m.PuntoDeseadoY, t.IdTerapia, m.Tiempo, m.FuerzaX, m.FuerzaY, m.FuerzaAsistiva,
m.VelocidadX, m.VelocidadY, p.IdPaciente

FROM        Muestra AS m INNER JOIN

            TerapiaBM AS t ON m.IdTerapia = t.IdTerapia INNER JOIN

            PacientesBM AS p ON t.IdPaciente = p.IdPaciente

WHERE      (p.IdPaciente = 7) AND (t.IdTerapia <= 1000) AND (t.IdTerapia >= 200)
```

Vemos que ahora hemos a través del comando INNER JOIN hemos conseguido añadir a la tabla Muestra, las tablas PacientesBM y TerapiaBM para así poder tener los datos que nos interesan en la misma tabla para poder trabajar mejor con ellas.

2.2 ANÁLISIS DE DATOS

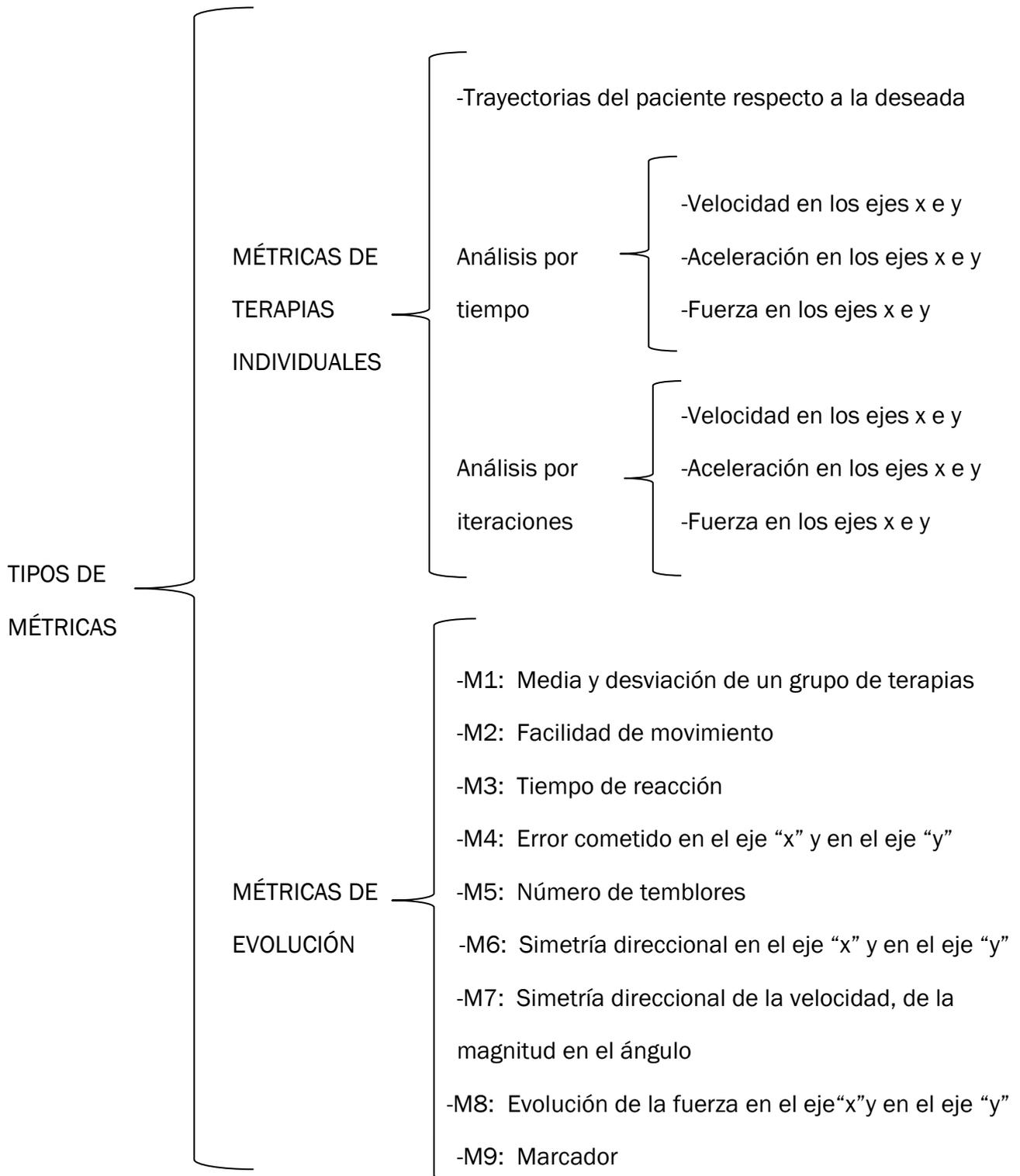
El análisis de datos se va a realizar a través de distintas métricas y gráficas algunas de estas muestran como ha sido una terapia de forma individual y otras muestran una evolución del paciente en varias terapias, teniendo en cuenta varios parámetros, en estas se analizan varias terapias, para poder ver así una evolución temporal del paciente.

Dentro de las métricas individuales vamos a tener por un lado la trayectoria del paciente con respecto a la deseada, y luego se van a dividir según su análisis en terapias según el tiempo o las iteraciones, o según el tipo de terapia, si es activo asistida o pasivo guiada,

En nuestro caso como ya explicaremos vamos a analizar las terapias activo asistida con las gráficas según el tiempo y la pasivo guiadas según las iteraciones.

En cuanto a las métricas de evolución tenemos nueve tipos de métricas distintas según los parámetros que vamos a analizar.

A continuación en la figura 11 vamos a ver un esquema de su organización.



*La "M" de las métricas de evolución significa métrica, de esta forma M1, significa métrica 1.

Figura 11. Esquema del tipo de métricas



2.2.1 MÉTRICAS DE TERAPIAS INDIVIDUALES

- Trayectorias del paciente respecto a la deseada.

A partir de los datos de los puntos deseados en “x” y en “y”, los cuales son los que guardamos en la base de datos de la trayectoria mostrada por pantalla, en la cual el paciente puede ver como se mueve el ratón o que trayectoria debe de seguir con la mano. También tomamos los datos de las posiciones en “x” y en “y” que realiza el paciente cuando mueve el robot. Todos los datos están en milímetros.

Con estos datos graficamos como “trayectoria deseada” con los puntos deseados que son los que se hacen en la pantalla, en el caso de ser una terapia pasivo guiada, los puntos van a parecer siempre los mismos porque la trayectoria va a ser siempre la misma y se grafican unos puntos encima de los otros, mientras que en los datos que graficamos como “trayectoria realizada” son todos los puntos en función de x e y que ha realizado el paciente en cada una de las repeticiones si es la terapia dos o si es la terapia uno todos los puntos de la persecución del ratón. Como hemos explicado antes, a los puntos deseados debemos de hacerles una calibración para que se puedan medir en la misma escala.

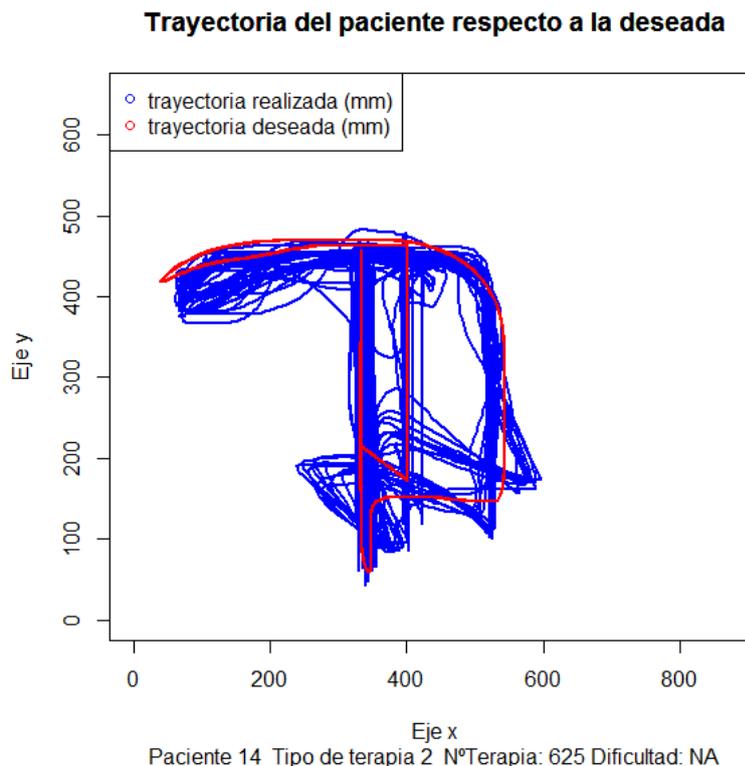


Fig. 12. Trayectoria del paciente respecto a la deseada.



Podemos ver como la gráfica que obtenemos las unidades están en milímetros además en la parte de abajo nos indica el paciente, el tipo de terapia, el número de la terapia y la dificultad de la misma, en este caso al ser una terapia de tipo 2 no tiene dificultad dado que solo las terapias de tipo 1 la tienen.

2.2.1.1 ANALISIS INDIVIDUAL CON RESPECTO A ITERACIONES

En este tipo de gráficas vamos a analizar varias vueltas, de estas vueltas vamos a poder elegir entre el valor máximo, valor mínimo o valor medio que se producen en una vuelta de las iteraciones que realiza el robot. Sólo funciona para la terapia 2 porque es en la única en la que el recorrido se produce varias veces.

Además estas graficas tendrán dos escalas una escala será para la distancia en mm situada a la derecha, y a la izquierda la correspondiente a aceleración, velocidad o fuerza, también podremos analizar en tanto en el eje de las “x” como de las “y”.

El tiempo que se muestra es el tiempo por vuelta, es decir que siempre empezará desde cero, y si la vuelta ha sido en el minuto 2 o en el 3 no se tiene en cuenta.

A la hora de escoger las vueltas el programa no siempre funciona correctamente, porque los datos que tenemos en la base de datos no son exactos, la primera idea fue fijarse en el punto inicial del robot y ver cuando el robot volvía a pasar por él, y a esto lo consideraríamos una vuelta, pero el robot tiene el error de que un número inicial de valores variable no es correcto, entonces no podemos seleccionar el valor inicial, por ello hemos encontrado un valor que es el valor 600 del vector que a partir de ahí ninguno da este error inicial, vale ahora podríamos coger un punto y cuando volviese a pasar decir que ha dado una vuelta, pero tenemos otro fallo, dado que los ratos los recoge según el tiempo y no la posición, que tenga el misma posición no es posible porque que das dará la posición unos milímetros adelante o atrás, por lo tanto hemos supuesto que todos los puntos dentro del área de un cuadrado con 20 milímetros de lado cuyo centro sea ese punto que hemos tomado como inicial en el valor 600. Cada vez que pase por esa área y para evitar que haya cruces se hayan recogido como mínimo 650 datos se dirá que ha recorrido una vuelta. El código que empleamos y que hemos explicado es el siguiente:

```
limites<-vector()
for(i in 1:(maxleng)){
  if(filtro[i]>a+650) {
    limites[b]=i
    b<-b+1
    a<-i}}}
```



Dónde “*maxleng*” es la máxima longitud del vector de puntos deseados en el eje x de la terapia, “*filtro*” es un vector formado por ceros y por los puntos deseados que a la vez en el eje x y en el eje y están dentro del cuadrado de 20 milímetros que hemos explicado antes. En cuanto a “*a*” y “*b*” son contadores y límites va a ser un vector que guarde los puntos de inicio de cada vuelta.

Para que de un error en esta terapia tiene que pasar que un robot no haga una trayectoria larga y por lo tanto nos muestre más de una vuelta en pantalla a la vez, o que el robot haga una trayectoria menor de un minuto en cuyo caso no podrá haber realizado apenas una vuelta y no podremos realizar nada.

Para seleccionar que vuelta es la máxima o la media empleamos un proceso iterativo:

-Para calcular la vuelta máxima, hacemos la media en valor absoluto de la primera vuelta y guardamos el resultado como valor máximo y guardamos como vuelta máxima la primera vuelta, comparamos con la segunda vuelta si el valor máximo de la segunda vuelta es mayor que el de la primera se guarda como valor máximo y se guarda que vuelta es y pasaría a compararse de nuevo, y así sucesivamente hasta que no hubiese más vueltas.

-Para calcular la vuelta mínima haríamos lo mismo que en la máxima solo que en lugar de comparar que valor es el máximo lo haríamos con el mínimo y se guardaría en caso de ser un valor menor.

-Para calcular la vuelta media hacemos la media absoluta total de los valores, luego hacemos la media de cada vuelta, y la que más se aproxime es la que llamamos la vuelta media.

- **Velocidad en una terapia pasiva**

Hay estudios que utilizan esta métrica para analizar la fluidez del movimiento como en [16] y poder con ella establecer una evolución del paciente, de forma que a mayor soltura o facilidad de movimiento más rápido irá el paciente.

Velocidad en una terapia pasiva en el eje x.

Tomamos como referencia los puntos deseados del robot de forma que 0 será cuando el robot este a la izquierda y en 0,8 el robot está a la derecha de forma que si sube la gráfica de puntos deseados el robot se moverá hacia la derecha y si baja se moverá hacia la izquierda, las medidas serán en metros para los puntos realizados por el paciente, además representaremos las gráficas de la velocidad que están en metros/segundos con el mismo principio cuando la pendiente sea positiva se estará moviendo a la derecha, y cuando sea negativa hacia la izquierda.

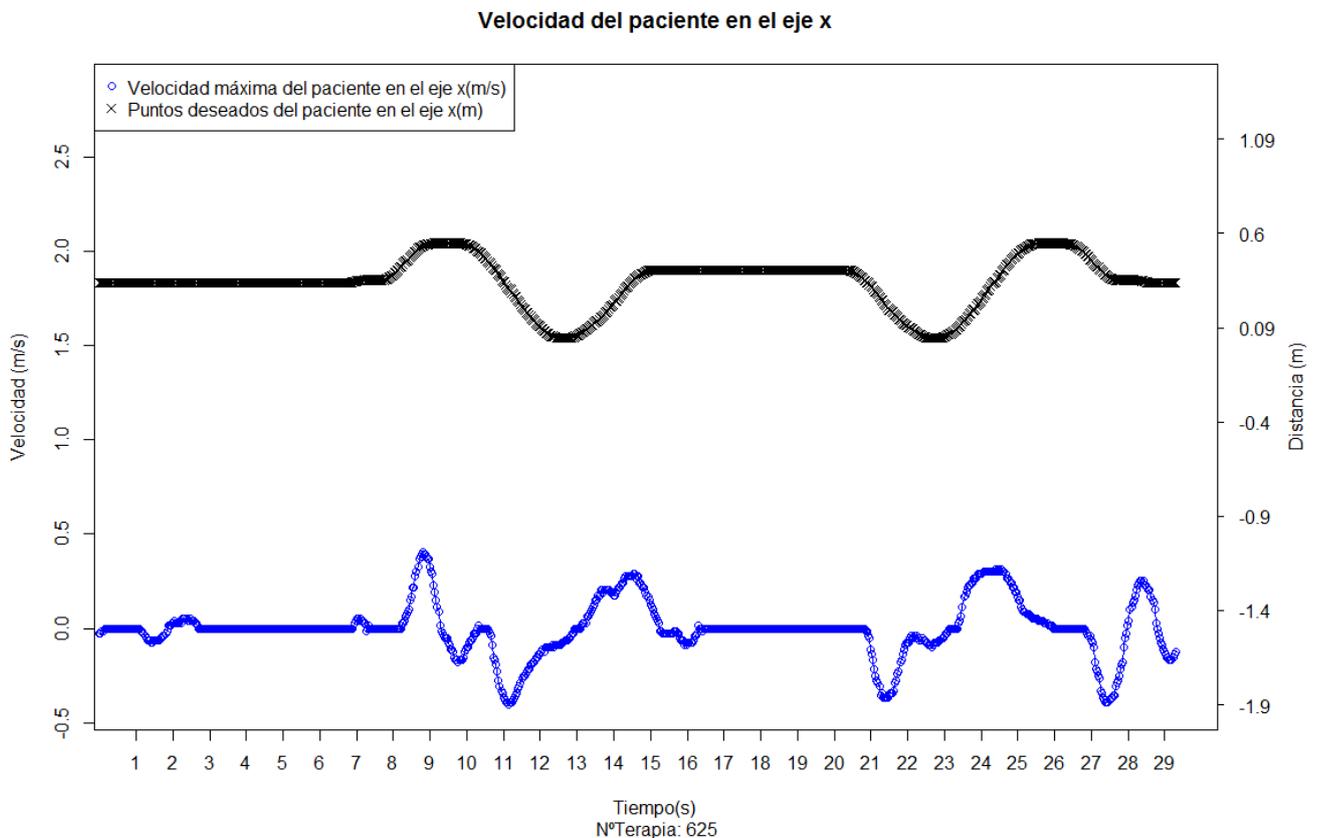


Fig. 13 Velocidad en una terapia pasiva en el eje x

En la figura (Fig. 13) que podemos ver arriba hemos seleccionado la velocidad máxima. La velocidad está en metros partido segundos y la distancia en metros, se puede ver la relación entre los puntos deseados y la velocidad porque cuando fluctúa en los puntos deseados lo hace igualmente la velocidad.

Velocidad en una terapia pasiva en el eje y

Tomamos como referencia los puntos deseados del robot de forma que 0 será cuando el robot este abajo y en 0,8 el robot está arriba de forma que si sube la gráfica de puntos deseados el robot se moverá hacia arriba y si baja se moverá hacia abajo, las medidas serán en metros para los puntos realizados por el paciente, además representaremos las gráficas de la velocidad que están en metros partido por segundos con el mismo principio cuando la pendiente sea positiva se estará moviendo hacia arriba, y cuando sea negativa hacia abajo.

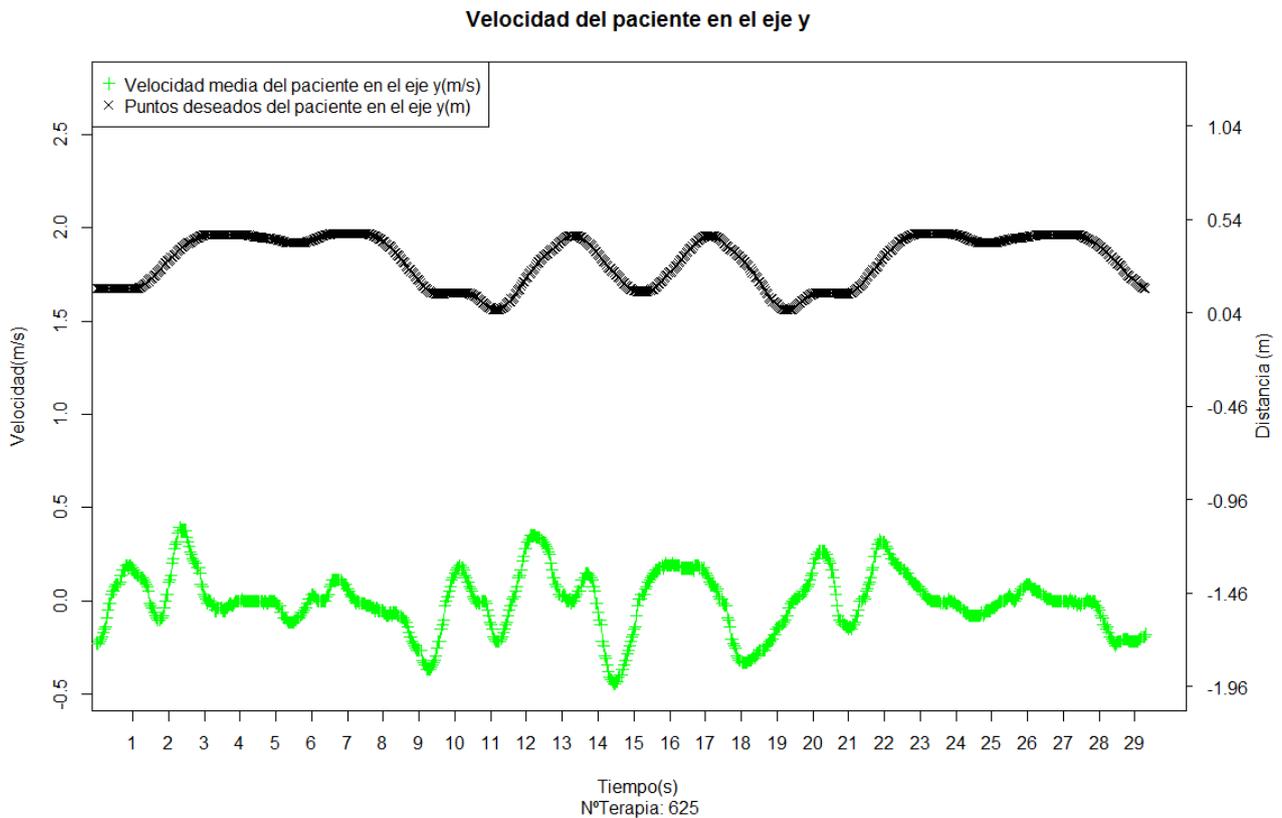


Fig. 14 Velocidad en una terapia pasiva en el eje y

En la figura (Fig. 14) que podemos ver arriba hemos seleccionado la velocidad media. La velocidad está en metros partido por segundos y la distancia en metros, se puede ver la relación entre los puntos deseados y la velocidad porque cuando fluctúa en los puntos deseados lo hace igualmente la velocidad, el tener una velocidad negativa como hemos dicho se corresponde a que está yendo por debajo del punto medio del robot.

- **Aceleración en una terapia pasiva**

Para poder calcular la aceleración debemos de realizar la diferencial de velocidad con respecto del tiempo debido a que en los datos que nos proporciona el robot ninguno de ellos es la aceleración, pero al producir esta transformación en lugar de quedarnos una forma de onda lo que nos quedan son muchos puntos a través de los cuales se puede intuir una onda pero tiene mucho ruido por lo tanto para poder medirla correctamente ha sido necesario hacer un filtro de Kalman, para ello se ha empleado la librería KFK [17][18].



El filtro de Kalman es un algoritmo de predicción y corrección de un sistema dinámico lineal que nos sirve para corregir el ruido producido por el uso de valores diferenciales, hemos escogido este filtro entre otros como podría ser el observador de Luenberger, es debido a que la ganancia de la realimentación del error con el observador de Luenberger hay que escogirlo mientras que con el filtro Kalman se puede escoger de forma correcta tras conocer los ruidos que están afectando al sistema.

El código que hemos empleado para poder desarrollar el algoritmo de Kalman, contiene el algoritmo que hemos empleado para conseguirlo, para ello vamos a tomar por ejemplo el código empleado en la aceleración por vueltas en el eje de x:

```
y <- maxacex
y[c(3, 10)] <- NA # Para que los valores NA sean tratados
## Inicializamos las constantes en los parámetros:
dt <- ct <- matrix(0)
Zt <- Tt <- matrix(1)
a0 <- y[1]      # Variación del primer valor
PO <- matrix(100) # variación de a0
## Parametros estimados:
fit.fkf <- optim(c(HHt = var(y, na.rm = TRUE) * .5,
GGt = var(y, na.rm = TRUE) * .5),
fn = function(par, ...)
-fkf(HHt = matrix(par[1]), GGt = matrix(par[2]), ...) $logLik,
yt = rbind(y), a0 = a0, PO = PO, dt = dt, ct = ct,
Zt = Zt, Tt = Tt, check.input = FALSE)
## Filtramos la aceleracion con los parametros estimados:
amaxxfkf <- fkf(a0, PO, dt, ct, Tt, Zt, HHt = matrix(fit.fkf$par[1]),
GGt = matrix(fit.fkf$par[2]), yt = rbind(y))
```

Vamos a analizar que hace este código y que hace la función FKF:

La funciónFKF permite el filtrado de Kalman rápido y flexible. Ambos, la medición y la transición ecuación puede ser multifactorial y parámetros podrá ser variable en el tiempo. Además los valores del tipo "NA" de las observaciones son compatibles. La función que empleamos es la siguiente:

```
fkf(a0, PO, dt, ct, Tt, Zt, HHt, GGt, yt, check.input = TRUE)
```

Los argumentos que empleamos son los siguientes:



- a0: Un vector que da una estimación inicial de la variable de estado.
- P0: Una matriz que da la varianza de a0.
- Dt: Una matriz dando la intersección de la ecuación de transición.
- Ct: Una matriz dando la intersección de la ecuación de medida.
- Tt: Una matriz que da el factor de la ecuación de transición.
- Zt: Una matriz que da el factor de la ecuación de medición.
- HHT: Una matriz que da la varianza de las innovaciones de la ecuación de transición.
- GGT: Una matriz dando la varianza de las perturbaciones de la ecuación de medición.
- Yt: Una matriz que contiene las observaciones, los valores "NA" están permitidos.
- check.input: Comprobación de si la entrada es coherente.

Forma de espacio de estado:

El modelo de espacio de estado se representa por la ecuación de transición y la ecuación de medición. Sea m la dimensión del estado variable d ser la dimensión de las observaciones, y n el número de observaciones. La ecuación de transición y la ecuación de medición están dados por:

$$\alpha_{t+1} = d_t + T_t \cdot \alpha_t + H_t \cdot \eta_t \quad (1)$$

$$y_t = c_t + Z_t \cdot \alpha_t + G_t \cdot e_t \quad (2)$$

Dónde η_t y e_t son variables aleatorias independientes e idénticamente distribuidas por la normal $N(0, I_m)$ e independiente e idénticamente distribuida variables aleatorias por la normal $N(0, I_d)$, respectivamente, y α_t indica la variable de estado.

Siendo I_m la matriz de números imaginarios e I_d la matriz identidad.

Podemos ver en la siguiente imagen como hemos contrapuesto la aceleración con filtro y sin filtro para ver la diferencia entre usarlo o no.

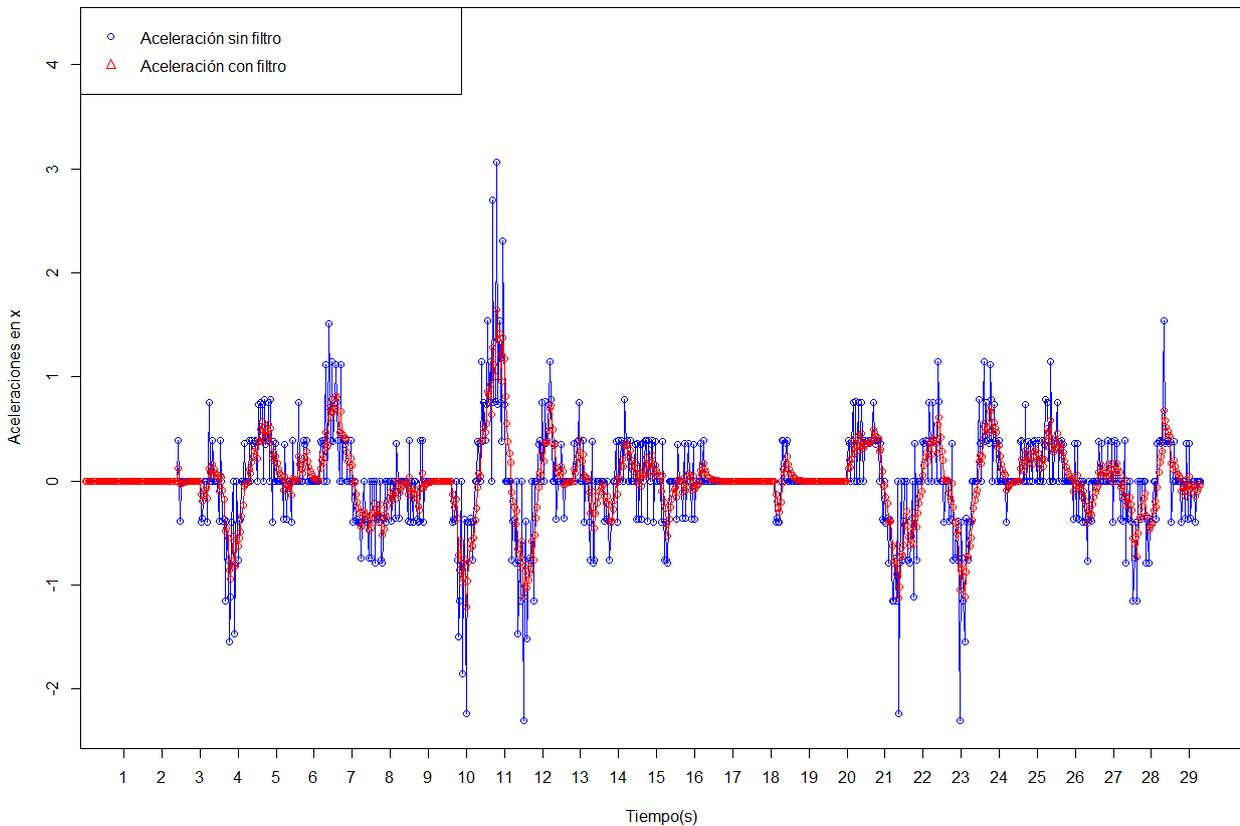


Fig. 15 Efecto del filtro de Kalman sobre la aceleración

Podemos ver como la aceleración sin filtro tiene enorme sobrepicos y aunque se puede intuir la forma de la onda no es continua y tiene mucho ruido, en cambio la aceleración con filtro se ve como una onda continua y sin ruido.

Aceleración en una terapia pasiva en el eje x.

Al igual que en la velocidad tomamos como referencia los puntos deseados del robot de forma que 0 será cuando el robot este a la izquierda y en 0,8 el robot está a la derecha de forma que si sube la gráfica de puntos deseados el robot se moverá hacia la derecha y si baja se moverá hacia la izquierda, las medidas serán en metros para los puntos realizados por el paciente, además representaremos las gráficas de la aceleración que están en metros cuadrados partidos por segundo con el mismo principio cuando la pendiente sea positiva se estará moviendo a la derecha, y cuando sea negativa hacia la izquierda. Para conseguir la forma de onda actual, como hemos dicho antes empleamos un filtro de Kalman.

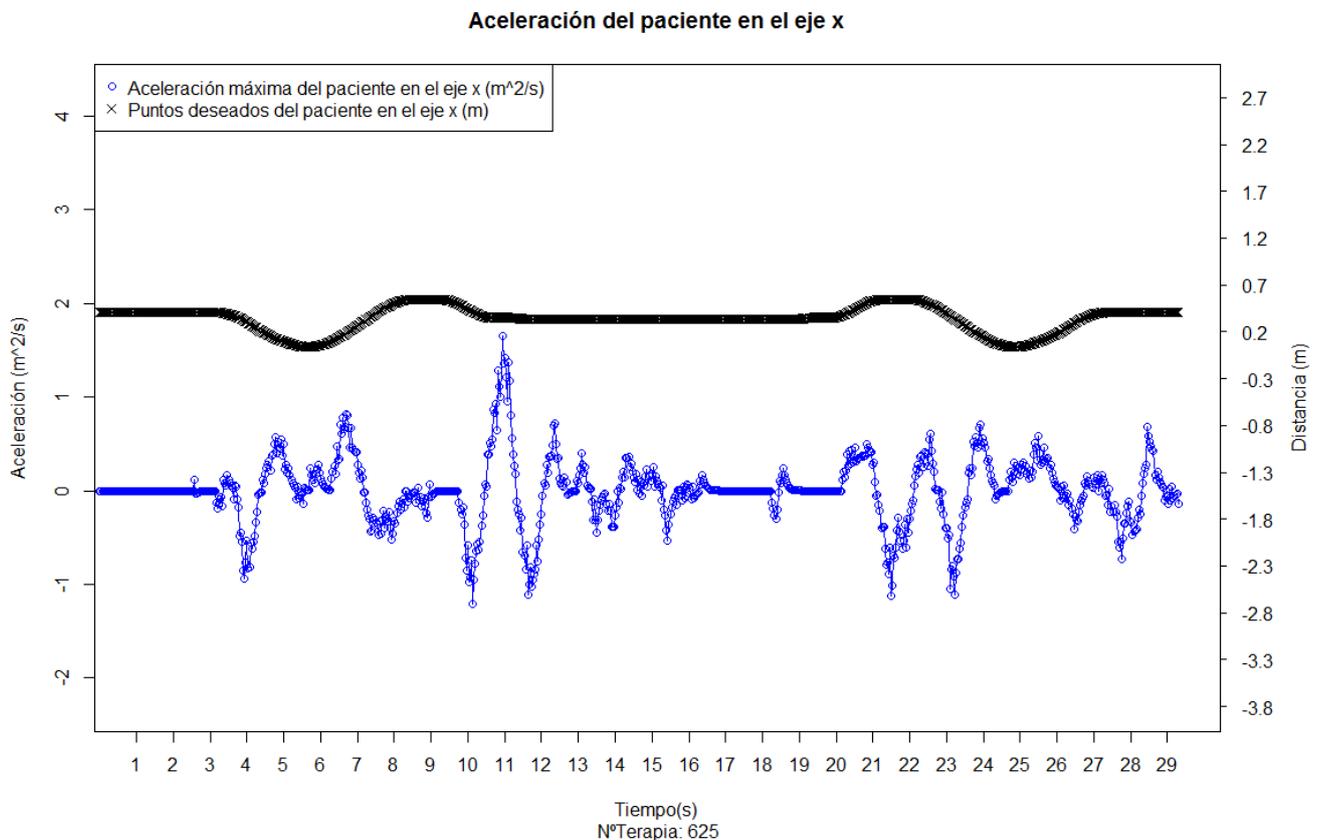


Fig. 16 Aceleración en una terapia pasiva en el eje x

Analizamos la gráfica (Fig. 16) y vemos que no se relaciona tanto la aceleración con los puntos deseados pero aun así vemos como cuando no hay un movimiento en el eje de las x, que los puntos permanecen en 0, la aceleración tiende a cero o no tiene tantos sobrepicos, la aceleración va con algo de retraso con respecto a los puntos deseados porque el paciente tarda en reaccionar.

Aceleración en una terapia pasiva en el eje y.

Al igual que en la velocidad tomamos como referencia los puntos deseados del robot de forma que 0 será cuando el robot este abajo y en 0,8 el robot está arriba de forma que si sube la gráfica de puntos deseados el robot se moverá hacia arriba y si baja se moverá hacia abajo, las medidas serán en metros para los puntos realizados por el paciente, además representaremos las gráficas de la aceleración que están en metros cuadrados partido por segundos con el mismo principio cuando la pendiente sea positiva se estará moviendo hacia arriba, y cuando sea negativa hacia abajo. Para conseguir la forma de onda actual, como hemos dicho antes empleamos un filtro de Kalman.

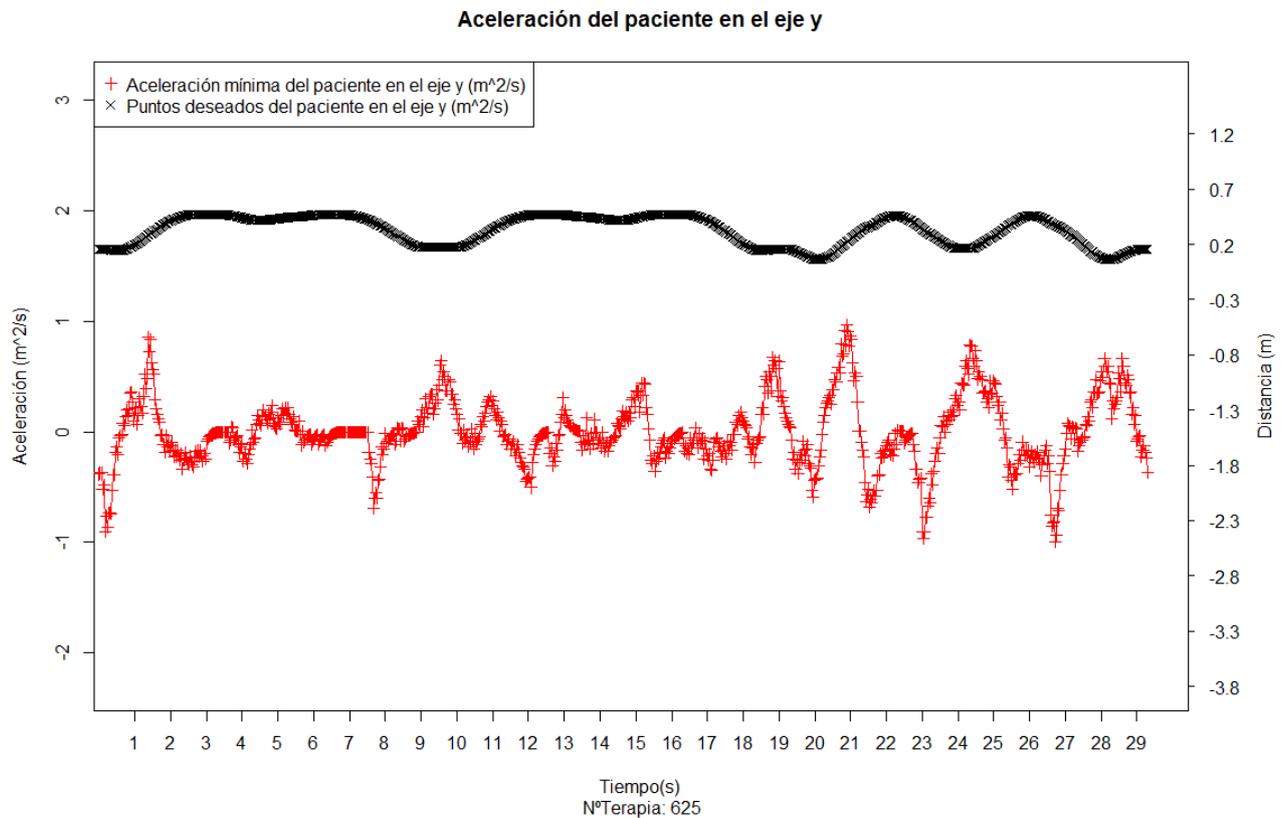


Fig. 17 Aceleración en una terapia pasiva en el eje y

Analizando la figura 17, vemos que se trata de la aceleración mínima del paciente, no coinciden los sobrepicos con los puntos deseados del paciente pero se ve como varía mucho los puntos deseados, es decir que no está en reposo en ningún momento y lo hace igualmente la aceleración.

- **Fuerza en una terapia pasiva**

Para calcular las fuerzas vamos a necesitar conocer varias variables debido a que registramos la fuerza que realiza el paciente un sensor, que es lo que veíamos en el apartado Tablas como FuerzaX y FuerzaY, también tenemos una FuerzaAsistiva en las tablas que es lo que vamos a denominar ganancia K, esta nos va a ayudar a calcular la fuerza asistiva, de esta forma podemos calcular la fuerza que realiza physiobot, la fuerza asistiva que se realiza y la fuerza que ejerce el paciente, esto lo podemos ver en este diagrama de bloques:

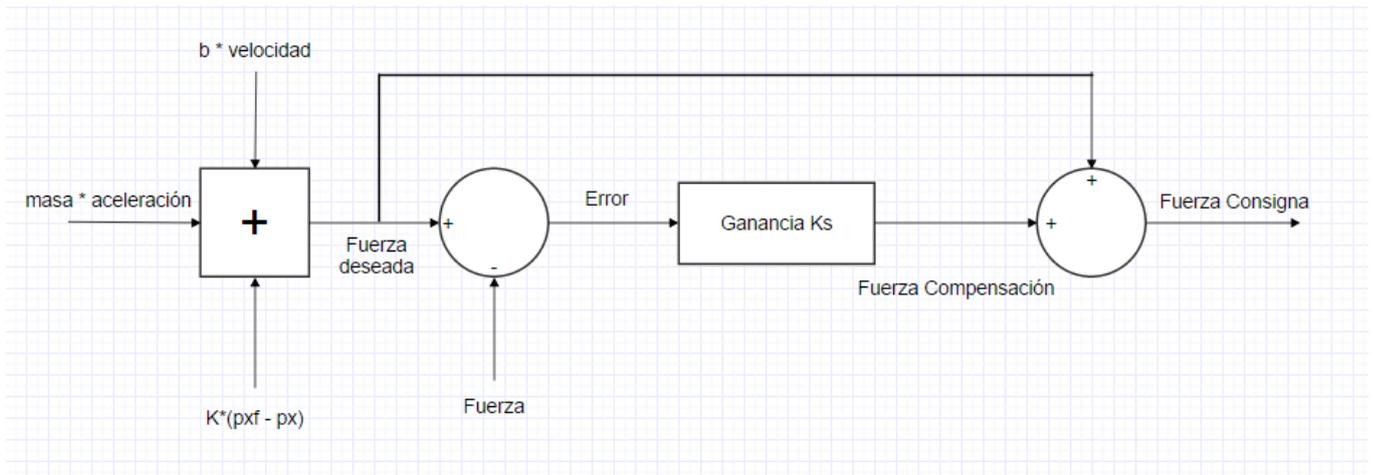


Fig. 18 Diagrama de bloques de la fuerza del Robot

A pesar de poder obtener la fuerza que realiza el robot, que no es la misma que la asistiva, para calcular la asistiva no necesitamos tanto y la podemos obtener de la siguiente ecuación:

$$Fuerza\ deseada = masa * aceleración + b * velocidad + K * (p_{xf} - p_x)$$

Siendo:

-**Fuerza deseada**: fuerza deseada según el algoritmo háptico, es la fuerza asistiva en el eje correspondiente.

-**masa**: masa del soporte del robot, es una constante de un kilogramo.

-**aceleración**: la aceleración producida en uno de los ejes.

-**b**: Es una constante con valor de 18kg/s.

-**velocidad**: Es la velocidad producida en uno de los ejes.

-**K**: Es la ganancia de la fuerza asistiva.

-**(p_{xf} - p_x)**: La diferencia entre el punto deseado y el punto realizado en uno de los ejes.

El resto de fuerzas y elementos del diagrama de bloques de la fuerza del Robot (Fig. 18) tienen el siguiente significado:

-**K_s**: Es la ganancia del sistema, es una constante de valor 5, es adimensional, esta ganancia enmascara la dinámica del sistema.



-**Fuerza Consigna:** Fuerza de consigna final enviada a los motores.

-**Fuerza Compensación:** Fuerza de compensación del error debido a la dinámica del sistema

-**Error:** Error entre la fuerza deseada y la medida por el sensor

Como podemos comprobar para poder calcular la fuerza asistida en el eje necesitamos la aceleración y por lo tanto volveremos a necesitar un filtro de Kalman para poder calcularlo.

Fuerza en una terapia pasiva en el eje x.

Al igual que hemos explicado antes en la aceleración y la velocidad en el eje x sólo que ahora las variables que vamos a tratar es la fuerza ejercida por el paciente y la fuerza que asiste el robot en el eje x, la fuerza está en Newtons, y comparamos fuerza máxima, mínima y media con la fuerza asistiva que se ha realizado en esa vuelta y con los puntos deseados que les medimos en decímetros en lugar de en metros.

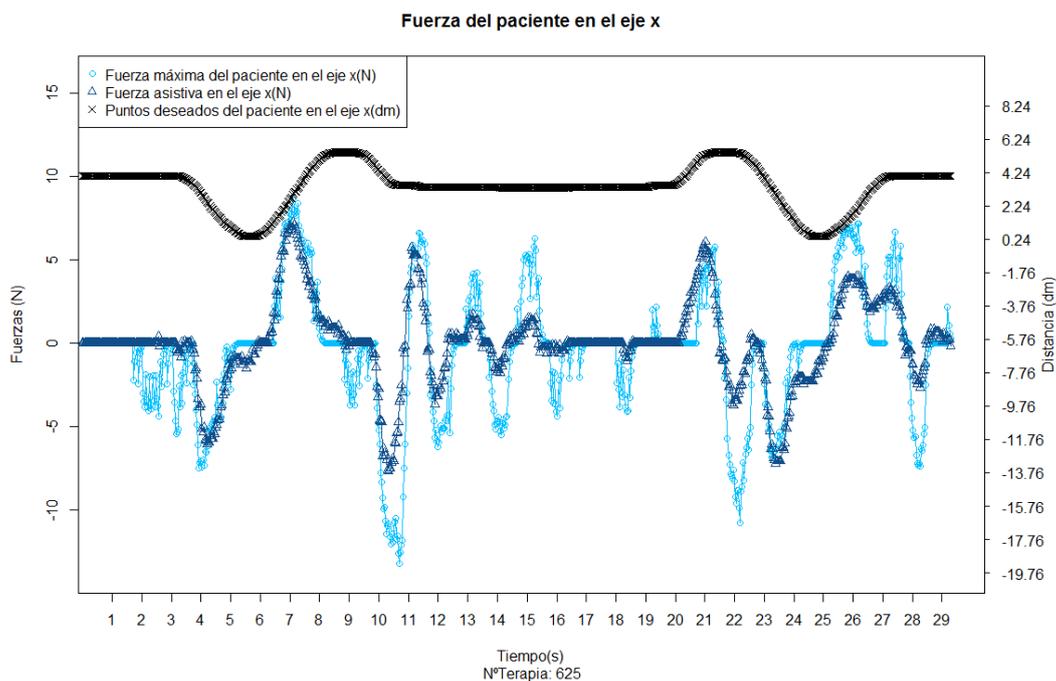


Fig. 19 Fuerza en una terapia pasiva en el eje x

En la figura 19 podemos observar como la fuerza asistiva es parecida pero en menor proporción a la fuerza del paciente, esto se debe a que el paciente no está haciendo la fuerza total que debería y por ello el robot le ayuda a completar el movimiento que debe de hacer.



Fuerza en una terapia pasiva en el eje y.

Al igual que hemos explicado antes en la aceleración y la velocidad en el eje y sólo que ahora las variables que vamos a tratar es la fuerza ejercida por el paciente y la fuerza que asiste el robot en el eje y, la fuerza está en Newtons, y comparamos fuerza máxima, mínima y media con la fuerza asistiva que se ha realizado en esa vuelta y con los puntos deseados que les medimos en decímetros en lugar de en metros.

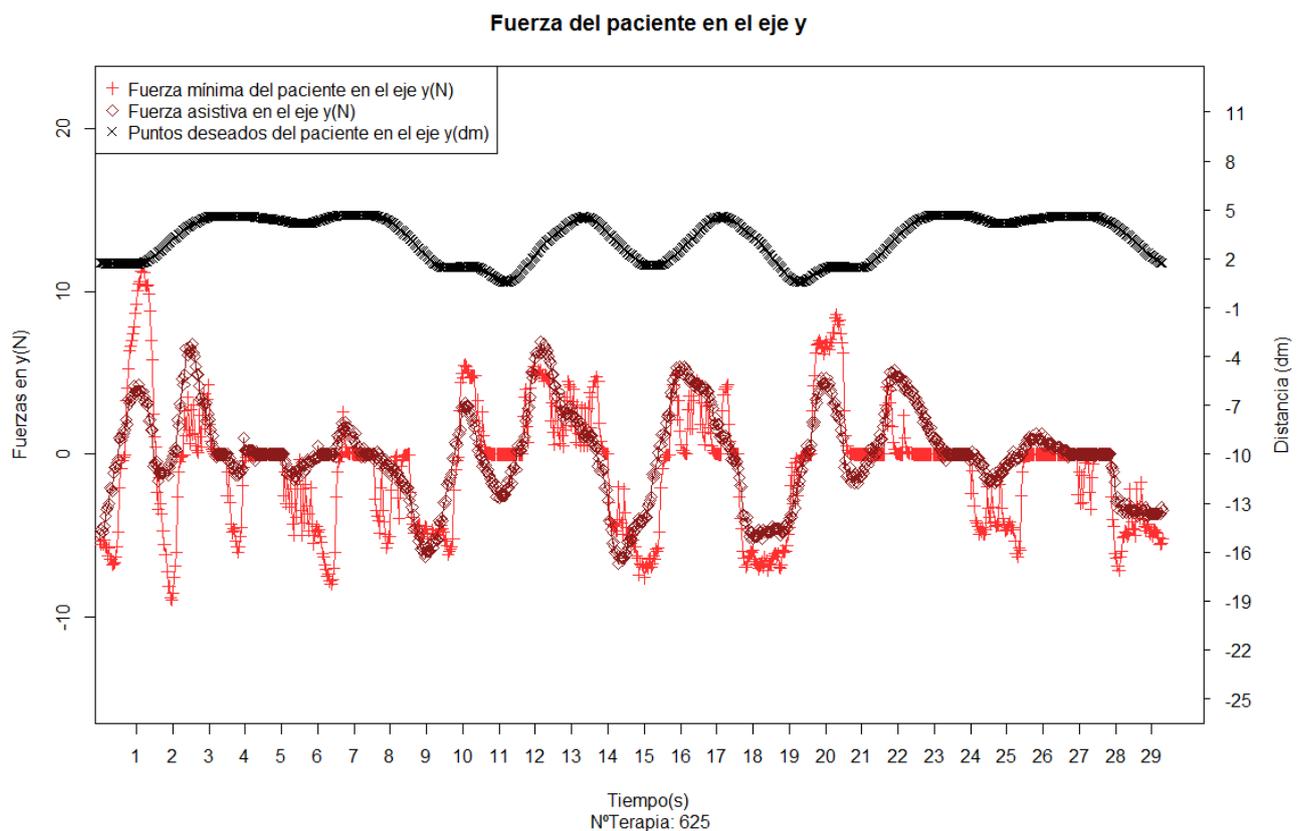


Fig. 20 Fuerza en una terapia pasiva en el eje y

Al igual que en la fuerza en el eje x, en la figura 20. la fuerza asistiva sumada a la fuerza del paciente ayuda al paciente a alcanzar la fuerza que necesita para cumplir su objetivo.



2.2.1.2 ANALISIS INDIVIDUAL CON RESPECTO AL TIEMPO

A diferencia del anterior análisis con respecto a iteraciones, este se va a realizar con respecto al tiempo de forma que se va a introducir un instante inicial y un instante final y se va a graficar ese tiempo, por lo tanto este tipo de análisis van a ser validas ambas terapia tanto las de tipo 1 como las de tipo 2.

En este caso como vamos a medir por tiempo en lugar de por vueltas nos puede causar problemas diferentes, por ejemplo a la hora de calcular el filtro de Kalman para tanto aceleración como para la fuerza, si ponemos un tiempo muy corto al ser un método predictivo nos va a dar un error debido a que le van a faltar datos para calcular, de forma que deberemos de poner un mínimo de 5 segundos.

Las métricas para el análisis con respecto al tiempo van a constar de dos gráficas, la gráfica superior nos va a mostrar los puntos deseada con respecto a los puntos realizados de esa forma vamos a poder analizar cómo el paciente sigue los puntos que va realizando el robot en la pantalla y en la gráfica inferior se va a ver la variable que elijamos entre velocidad, aceleración y fuerza en el eje x o y, de esta forma se va a poder analizar de una forma más realista como varía las variables con respecto al posicionamiento del paciente, y a la vez como varía este posicionamiento con respecto al realizado por el robot

Debido a que las terapias pasivas ya las hemos analizado en análisis iterativo, ahora nos vamos a centrar en las terapias activas.

- **Velocidad en una terapia activa.**

A la hora de analizar la velocidad en una terapia activa va a ser igual que a una pasiva la diferencia ahora es que al perseguir a un ratón la velocidad va a ir ligada a la velocidad del ratón, y por lo tanto va a ser dependiente del nivel de dificultad puesto que a mayor dificultad mayor velocidad alcanzará el ratón.

Velocidad en una terapia activa en el eje x.

Esta gráfica muestra en un intervalo de tiempo de una de las trayectorias del robot, tomamos como referencia los puntos deseados del paciente y los puntos realizados, de forma que 0 será cuando el robot este a la izquierda y en 0,8 el robot está a la derecha de forma que si sube la gráfica de puntos deseados el robot se moverá hacia la derecha y si baja se moverá hacia la izquierda, las medidas serán en metros, además representaremos las gráficas de la velocidad que están en metros/segundos con el mismo principio cuando la pendiente sea positiva se estará moviendo a la derecha, y cuando sea negativa hacia la izquierda.

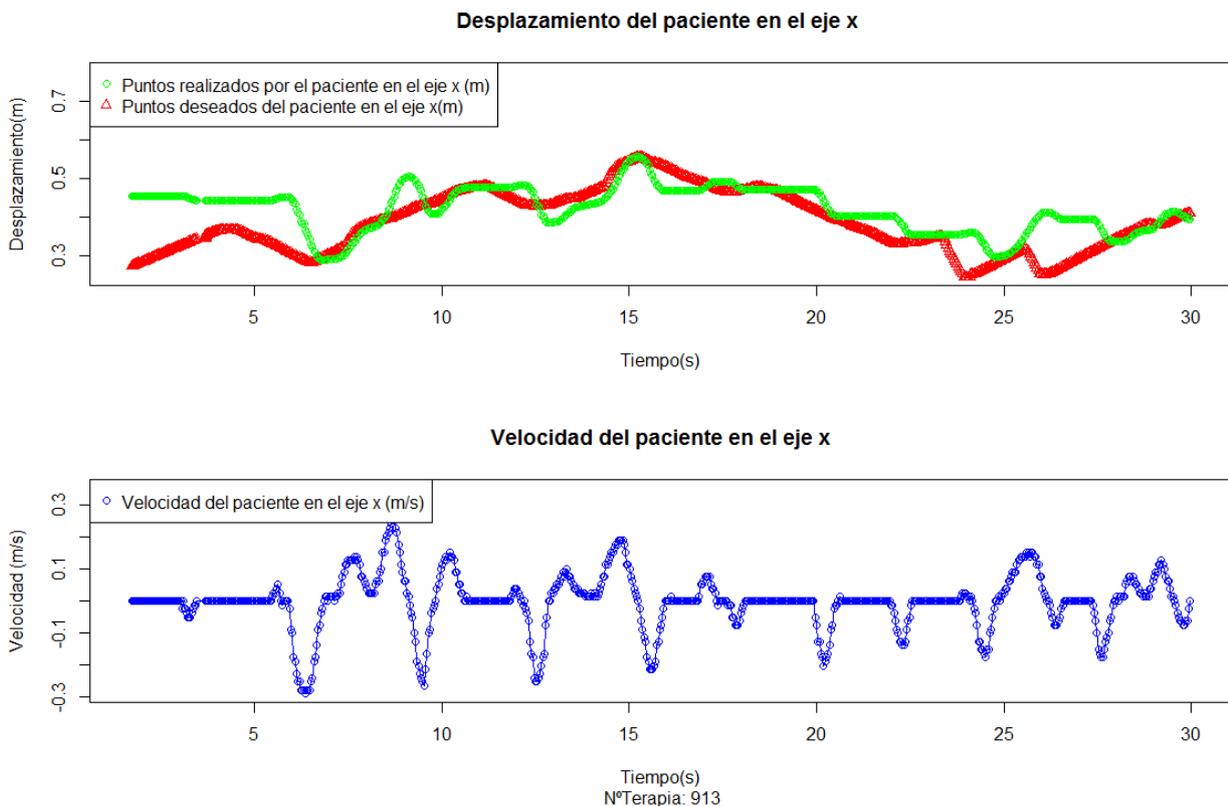


Fig. 21 Velocidad en una terapia activa en el eje x

Podemos ver en esta gráfica como concuerda bastante las variaciones de la velocidad con los puntos realizados por el paciente, de forma que podemos como al principio los cambios les hace más rápido y luego no tanto.

Velocidad en una terapia activa en el eje y

Tomamos como referencia los puntos deseados y los puntos realizados por el paciente de forma que 0 será cuando el robot este lo más abajo que puede y en 0,8 el robot está lo más arriba, de forma que si sube la gráfica de puntos deseados el robot se moverá hacia arriba y si baja se moverá hacia abajo, las medidas serán en metros para los puntos realizados por el paciente y los deseados, además representaremos las gráficas de la velocidad que están en metros partido por segundos con el mismo principio cuando la pendiente sea positiva se estará moviendo hacia arriba, y cuando sea negativa hacia abajo.

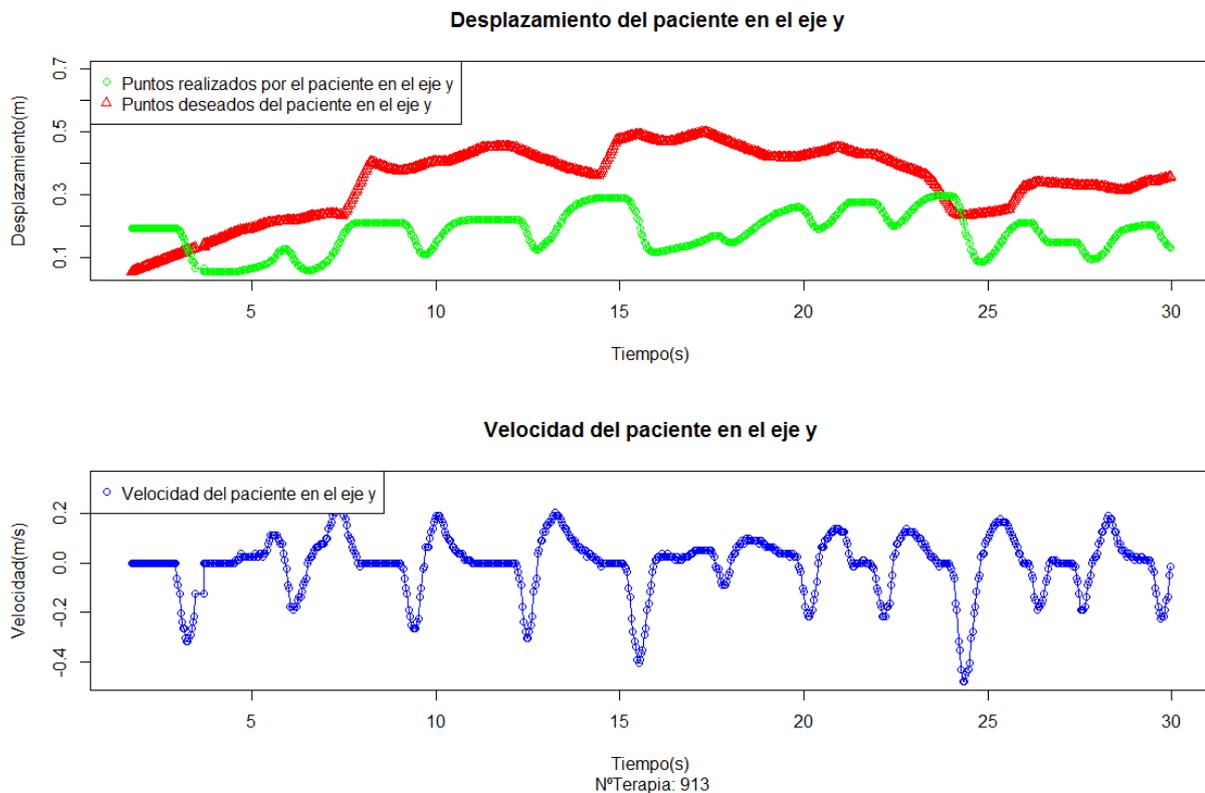


Fig. 22 Velocidad en una terapia activa en el eje y

En esta gráfica podemos ver como haya pocos picos positivos debemos de tener en cuenta las pendientes más que los sobrepicos debido a que al tener pendiente positiva eso implica que la velocidad está siendo positiva pero si parte de un valor negativo no se transforma en un valor positivo, es decir que si tenemos una velocidad negativa que llega hasta $-0,4$ m/s durante si luego tenemos una velocidad positiva de $0,4$ metros durante un metro terminará quedando cero y no dará un sobrepico.

- **Aceleración en una terapia activa.**

Como comentamos en la terapia pasiva para calcular la aceleración debemos de realizar la diferencial de velocidad con respecto del tiempo debido a que en los datos que nos proporciona el robot ninguno de ellos es la aceleración, pero al producir esta transformación en lugar de quedarnos una forma de onda lo que nos quedan son muchos puntos a través de los cuales se puede intuir una onda pero tiene mucho ruido por lo tanto para poder medirla correctamente ha sido necesario hacer un filtro de Kalman, para ello se ha empleado la librería KFK [17]. A mayores debemos de tener en cuenta que como también hemos dicho antes ante tiempos muy reducidos el filtro de Kalman al ser un algoritmo de tipo predictivo no



va a funcionar correctamente por ello a la hora de elegir intervalos de tiempo largos, es decir mayores de 5 segundos para la aceleración.

Aceleración en una terapia activa en el eje x.

Al igual que en la velocidad tomamos como referencia los puntos deseados del robot de forma que 0 será cuando el robot este abajo y en 0,8 el robot está arriba de forma que si sube la gráfica de puntos deseados el robot se moverá hacia arriba y si baja se moverá hacia abajo, las medidas serán en metros para los puntos realizados por el paciente, además representaremos las gráficas de la aceleración que están en metros cuadrados partido por segundos con el mismo principio cuando la pendiente sea positiva se estará moviendo hacia arriba, y cuando sea negativa hacia abajo. Para conseguir la forma de onda actual, como hemos dicho antes empleamos un filtro de Kalman.

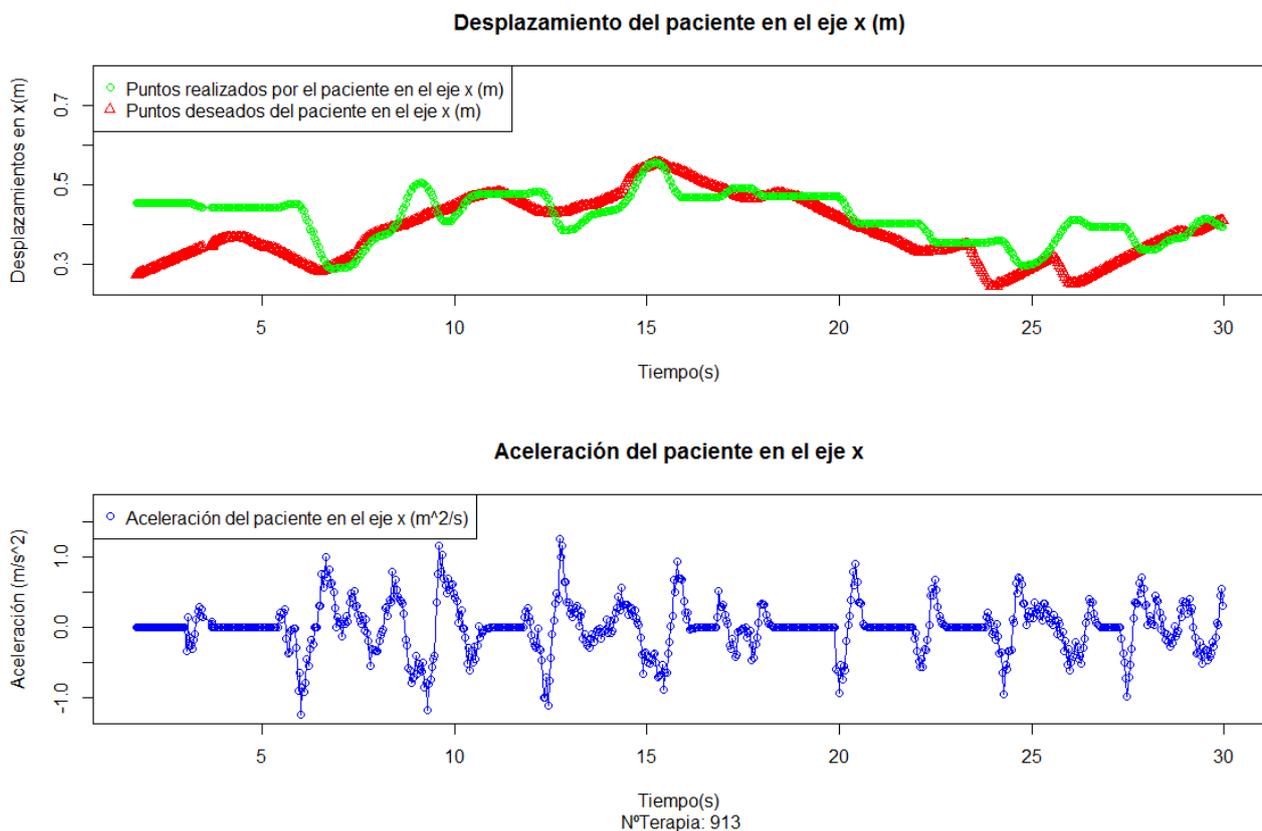


Fig. 23 Aceleración en una terapia activa en el eje x

Podemos analizar en la figura 23 como cada vez que se produce una variación en los puntos realizados eso conlleva una variación en la aceleración pero como se mantiene en cero cuando el movimiento varía esto se debe a que aunque haya velocidad no se produce aceleración.



Aceleración en una terapia activa en el eje y.

Al igual que en la velocidad tomamos como referencia los puntos deseados del robot de forma que 0 será cuando el robot este abajo y en 0,8 el robot está arriba de forma que si sube la gráfica de puntos deseados el robot se moverá hacia arriba y si baja se moverá hacia abajo, las medidas serán en metros para los puntos realizados por el paciente, además representaremos las gráficas de la aceleración que están en metros cuadrados partido por segundos con el mismo principio cuando la pendiente sea positiva se estará moviendo hacia arriba, y cuando sea negativa hacia abajo. Para conseguir la forma de onda actual, como hemos dicho antes empleamos un filtro de Kalman.

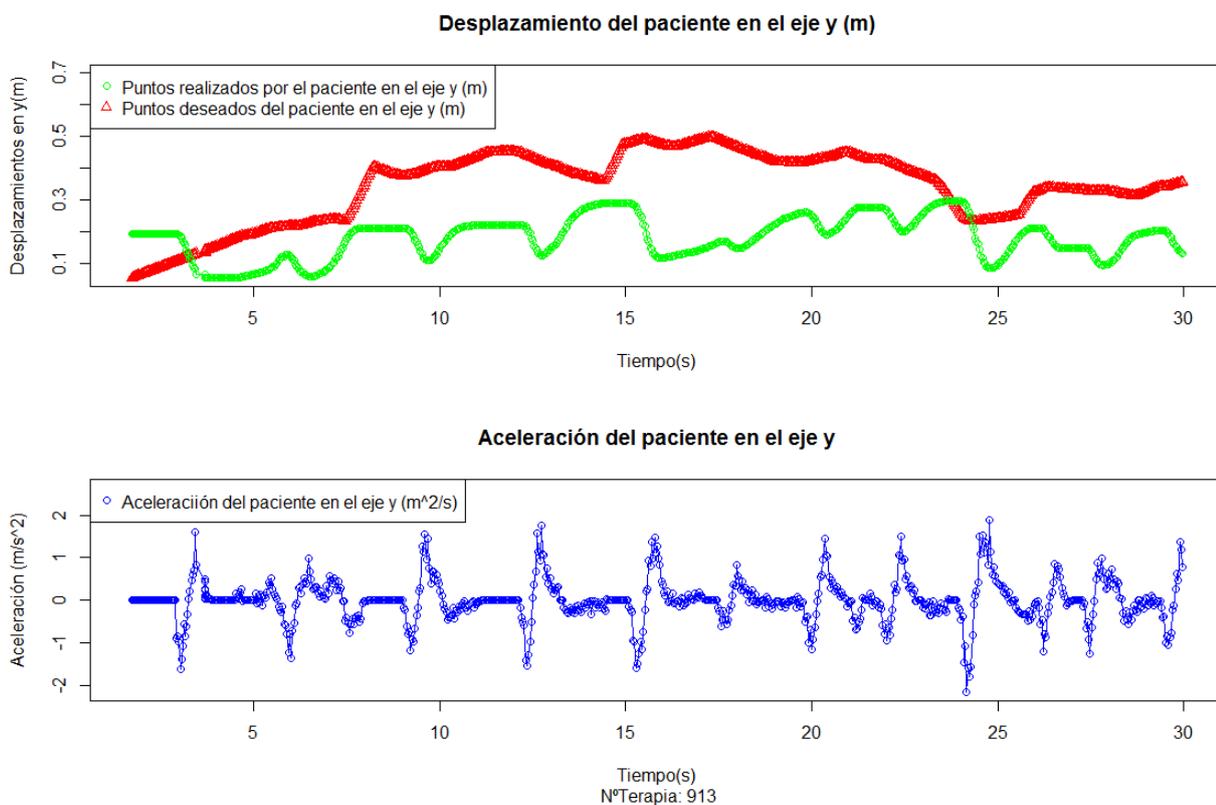


Fig. 24 Aceleración en una terapia activa en el eje y

En esta gráfica podemos observar como el paciente no llega a los puntos deseados y vemos que para intentarlo lo que el paciente realiza es dar impulsos que podemos ver en las aceleraciones.



- **Fuerza en una terapia activa.**

Funciona con el mismo principio que empleábamos en la terapia pasiva, de forma que vamos a volver a tener que emplear filtro de Kalman, también se va a ver afectada y no va a poder aplicarse a intervalos de tiempos pequeños como hemos explicado anteriormente.

Fuerza en una terapia activa en el eje x.

Al igual que hemos explicado antes en la aceleración y la velocidad en el eje x sólo que ahora las variables que vamos a tratar es la fuerza ejercida por el paciente y la fuerza que asiste el robot en el eje x, la fuerza está en Newtons, y comparamos fuerza máxima, mínima y media con la fuerza asistiva que se ha realizado en esa vuelta y con los puntos deseados así como con los realizados que les medimos en metros.

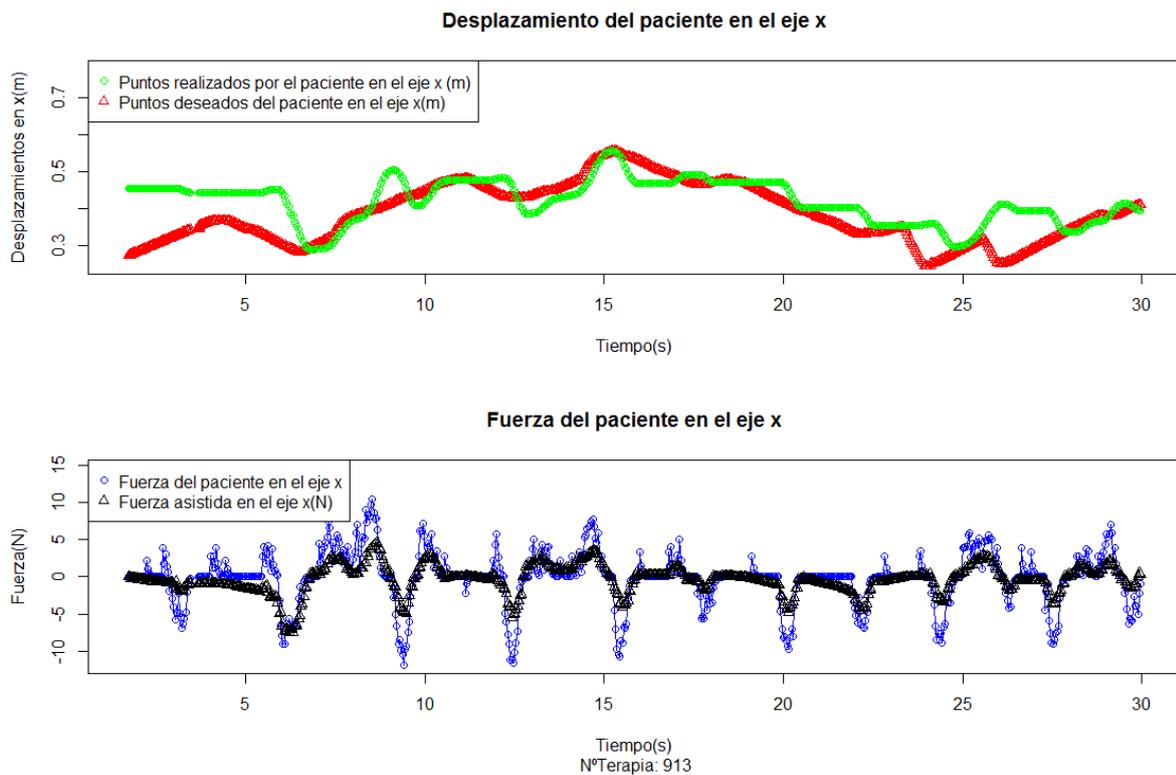


Fig. 25 Fuerza en una terapia activa en el eje x

Podemos analizar en la figura 25 vemos como cuando no llega a realizar el desplazamiento que se quiere coincidiendo así los puntos deseados con los realizados la fuerza asistiva le ayuda, de forma que aunque los puntos coincidan sigue actuando para que puedan seguir coincidiendo.

Fuerza en una terapia pasiva en el eje y.

Al igual que hemos explicado antes en la aceleración y la velocidad en el eje x sólo que ahora las variables que vamos a tratar es la fuerza ejercida por el paciente y la fuerza que asiste el robot en el eje y, la fuerza está en Newtons, y comparamos fuerza máxima, mínima y media con la fuerza asistiva que se ha realizado en esa vuelta y con los puntos deseados así como con los realizados que les medimos en metros.

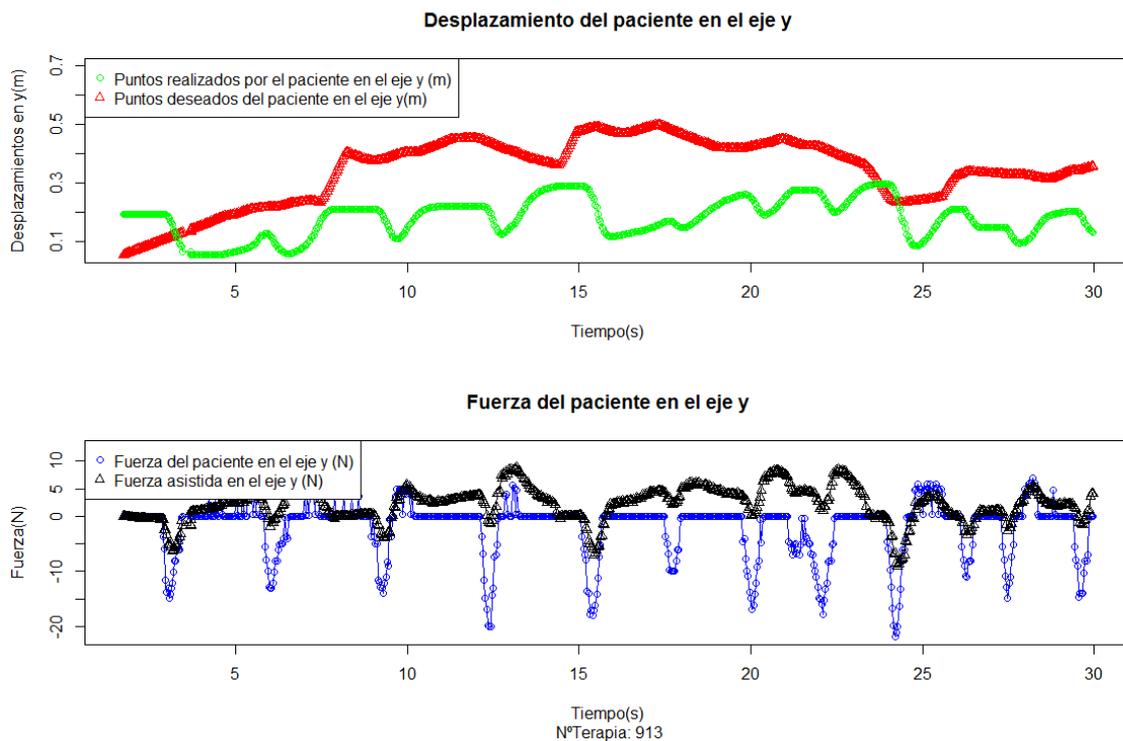


Fig. 26 Fuerza en una terapia pasiva en el eje y

En esta figura (Fig. 26) podemos ver cómo el paciente le está costando mucho subir hacia arriba es decir si nos fijamos en la fuerza del paciente vemos que en la parte positiva del eje y apenas la realiza y por ello la fuerza asistiva llega en ocasiones hasta 10 Newtons y vemos como en el desplazamiento si se desplaza en el eje de las y pero es debido a la fuerza asistiva y no a la que realiza el paciente.



2.2.2. METRICAS DE EVOLUCIÓN DE TERAPIAS

Las métricas de evolución tienen como objetivo ver la evolución de ciertos parámetros de un paciente durante varias terapias consecutivas, las terapias tienen un número asociado de forma cronológica, pero este número se intercala entre pacientes, es decir la terapia 203 se habrá realizado antes que la 204, pero si el paciente 7 ha realizado la terapia 203 y el paciente 14 las terapias 204,205 y 206, la siguiente terapia del paciente 7 será la 207, que seguirá siendo posterior a la 203. Y en la gráficas del paciente 7 aparecerá la terapia 203 seguida por la 207.

A la hora de analizar los datos sólo tomamos un tipo de terapias, o bien elegimos analizar las terapias activas o bien las terapias pasivas porque no tiene sentido analizar ambas a la vez dado que al ser diferentes terapias los datos varían.

La evolución de las terapias varía si tratamos con terapias activas o pasivas de forma que en alguno de los casos no se puede realizar un métrica para un determinado tipo de terapia y para otras no se puede considerar optimo el resultado debido a que no tiene mucho sentido evaluar esos términos en ese tipo de terapias. Además en alguna de estas terapias debido a los errores del robot a veces puede llegar a algún sobrepico en estas terapias pero no es habitual.

Como datos de entrada para el cálculo de la evolución de terapias tenemos que seleccionar un número de terapia inicial y un número de terapia final, que tipo de métrica queremos tener y que tipo de terapia.

Las gráficas estarán compuestas por un título que será el nombre de la gráfica en la parte superior, dos ejes, el eje de ordenadas será el correspondiente a la medida que estemos tomando mientras que el eje de abscisas estará compuesto por el intervalo de terapias que se ha seleccionado, también debajo del intervalo de terapia, nos especificará el paciente que hemos seleccionado, el tipo de terapia y la terapia inicial y final que han sido seleccionadas. Se incluye también una leyenda que nos especificará que tipo de rectas estamos empleando en la gráfica.

Tenemos nueve métricas de evolución distintas que son: media y desviación de un grupo de terapias, facilidad de movimiento, tiempo de reacción, error cometido en el eje "x" y en el eje "y", número de temblores, simetría direccional en "x" y en "y, simetría direccional de la velocidad, de la magnitud y del ángulo, evolución de la fuerza en "x" y en "y" y marcador. En los siguientes apartados vamos a ir analizando cada una de ellas.



- **Métrica 1: Media y desviación de un grupo de terapias.**

Media: Es la suma de la diferencia de los puntos geométricos en el plano en x y en y recorridos por el paciente con respecto a los puntos deseados, dividido entre el número de puntos recorridos.

La operación que realizamos para calcular la media será:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n \sqrt{(Px_i - Pdx_i)^2 + (Py_i - Pdy_i)^2} \quad (3)$$

Dónde:

Px: Es el punto recorrido en x medido en mm

Pdx: Es el punto deseado en x medido en mm

Py: Es el punto recorrido en y medido en mm

Pdy: Es el punto deseado en y medido en mm

Llamamos punto recorrido al punto que ha recorrido el paciente empleando el robot, mientras que llamamos punto deseado a la proyección del punto que está viendo el paciente sobre la pantalla en el mismo sistema que está empleando el paciente para moverse.

Desviación típica: Es el sumatorio de la raíz cuadrada del cuadrado de la diferencia de los puntos geométrico en el plano en x y en y recorridos por el paciente con respecto a los puntos deseados, dividido entre el número de puntos recorridos.

La operación que realizaremos para la desviación será:

$$s^2 = \frac{1}{n} \sum_{i=1}^n \left(\sqrt{(Px_i - Pdx_i)^2 + (Py_i - Pdy_i)^2} - \bar{x} \right)^2 \quad (4)$$

En esta ecuación “s” sería la desviación típica, y el resto de datos serían igual que en la media, y la \bar{x} que empleamos es la media calculada en el apartado anterior.

En la figura que vamos a ver a continuación cómo evoluciona el paciente en función del tiempo en los términos de desviación típica y de media aritmética, por ello en la gráfica vamos a poder observar dos rectas.

Esta gráfica sirve tanto para el tipo de terapia 1 como para el tipo de terapia dos pero los resultados son diferentes, dado que en las terapias pasivo guiada, al ser una vuelta repetida varias veces el paciente no tiene que reaccionar a nuevas vueltas que se puedan producir pero en la terapia activa asistiva el ratón se mueve por toda la pantalla y además del hándicap de las limitaciones físicas del paciente, tiene que lidiar también con el tiempo que tarda en reaccionar el paciente.

Los índices que tendremos serán la media y la desviación típica de los puntos de cada terapia.

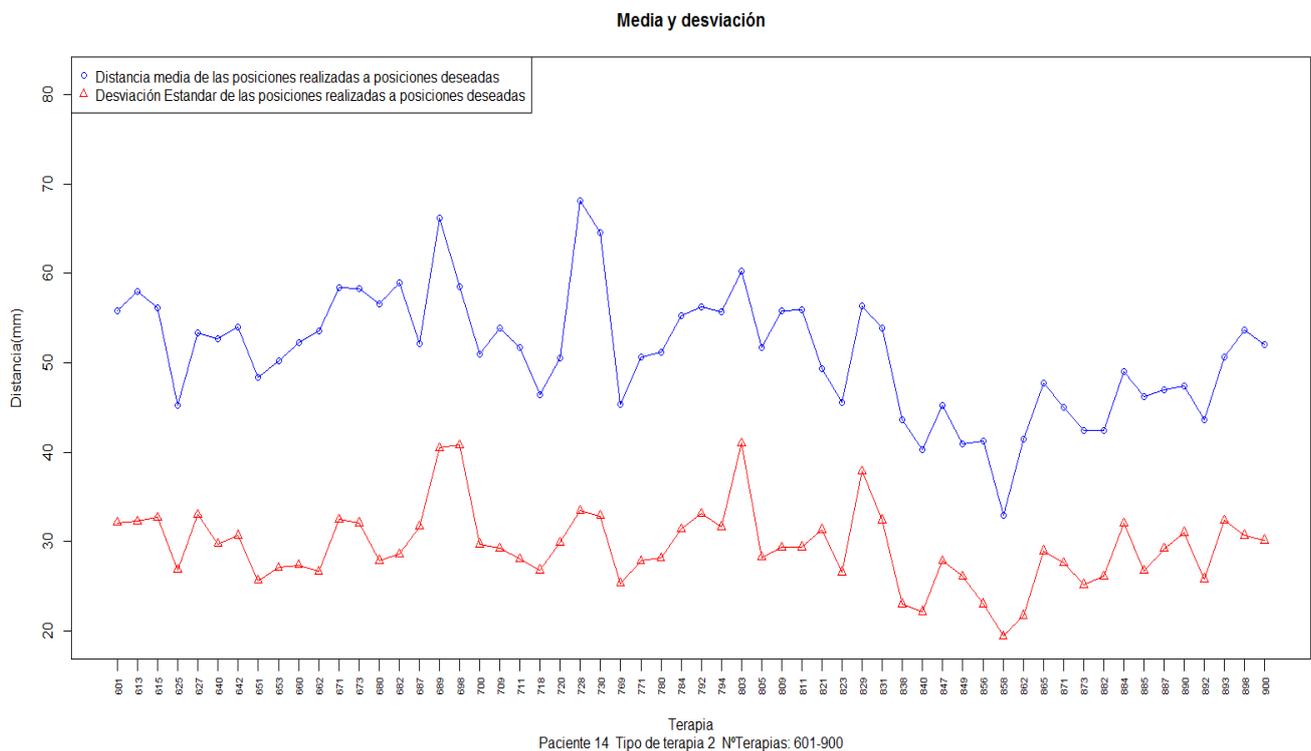


Fig. 27 Media y desviación de un grupo de terapias.

Podemos observar en la gráfica como cada vez que aumenta o disminuye la distancia media, la desviación estándar varía de forma igual y ante una distancia media recorrida menor conlleva consigo una desviación menos, la idea de esta gráfica es que manteniendo una distancia media constante si la desviación estándar va disminuyendo cada vez más, eso implicaría que la distancia que recorre se ajusta cada vez más a la establecida.



- **Métrica 2: Facilidad de movimiento**

Según varios estudios, como el de Nordin [19] dicen que la facilidad de movimiento viene dada o se puede concluir en función de la velocidad, de forma que cuanto más rápido tengas las capacidad de moverte, menos esfuerzo te costará realizar la tarea y por lo tanto tendrás una mayor facilidad de movimiento.

Para ello lo que vamos a realizar en esta gráfica es vamos a incluir una serie de puntos unidos por rectas cada punto asociado a la media de la velocidad de la terapia correspondientes y mediante una recta de regresión hemos unido el primer y el último punto de las terapias para que sea más fácil de entender visualmente.

El índice será la velocidad media medida en m/s.

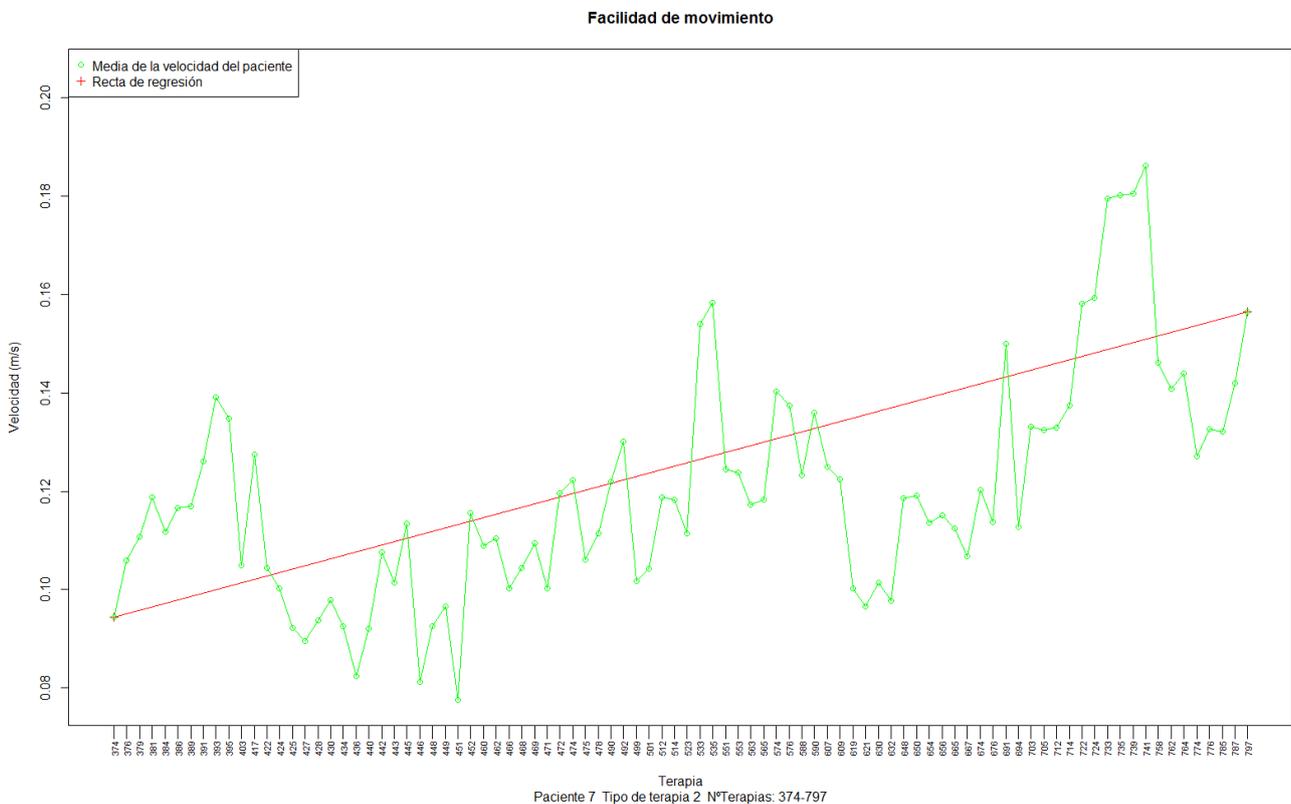


Fig. 28. Facilidad de movimiento



- **Métrica 3: Tiempo de reacción.**

Llamamos tiempo de reacción al tiempo que pasa desde que en la trayectoria deseada se produce un cambio de sentido hasta que en la trayectoria del robot se produce ese cambio de sentido también. Para ello restamos el tiempo en el que se produjo el cambio en el robot menos el tiempo en el que se produjo el cambio en el sentido deseado.

De forma que si el sentido de x o y cambia de los puntos de destino en la fila 5 y se produce un cambio en los puntos x e y en la fila 9, la diferencia de tiempo entre la fila 5 y la 9 será el tiempo que ha tardado en reaccionar.

Vamos a explicarlo mejor viendo el código:

```
dpyf<-diff(pyf)
dpxf<-diff(pxf)
dpx<-diff(px)
dpy<-diff(py)

dpyt<-dpyf*dpy
dpxt<-dpxf*dpx
```

En ese fragmento de código lo que hacemos es restar el punto realizado en x con el punto siguiente, de forma que si tenemos un vector de 8000 números restamos el primer número con el segundo, el segundo con el tercero y así sucesivamente, de forma que si son dos números iguales será 0, si la resta es negativa irá en un sentido y si es positiva irá en otro, de forma que cuando se produzca un cambio entre estos diremos que se ha producido un cambio de sentido, de forma que miramos el movimiento de la pantalla y cuando se produce un cambio de sentido en la pantalla esperamos hasta que se produzca un cambio de sentido en los puntos realizados.

Al multiplicar los puntos realizados con los deseados lo que conseguimos es que se queden los cambios de sentido porque al ser un sentido positivo y el otro negativo, al multiplicarlos solo quedarán números negativos, ceros y positivos, y los negativos serán los cambios de sentidos.

Más tarde se procede a filtrar los valores porque cuando en R multiplicamos cero por cero, no nos da cero sino nos da el valor "NA", por lo tanto convertimos esos valores a cero y nos quedamos con los valores negativos solamente, pero ordenamos en el vector, también

realizamos un filtro por si hay una pequeña variación y nos da dos cambios de sentidos en un tiempo muy breve no nos afecte.

Esta gráfica es sólo válida para terapias pasivo guiadas, porque en las terapias activo asistidas el movimiento del paciente no está siempre relacionado con el del ratón en la pantalla y eso nos puede llevar a errores.

Otra cosa que afecta bastante es un error que produce el robot ciertas veces consistente en que deja de medir el tiempo durante muchos segundos, de forma que si en ese tiempo hay un cambio de sentido en los puntos deseados pero no se produce el de puntos realizados nos puede dar un error de medición muy grande.

El índice que tendremos será el tiempo de reacción y se medirá en segundos.

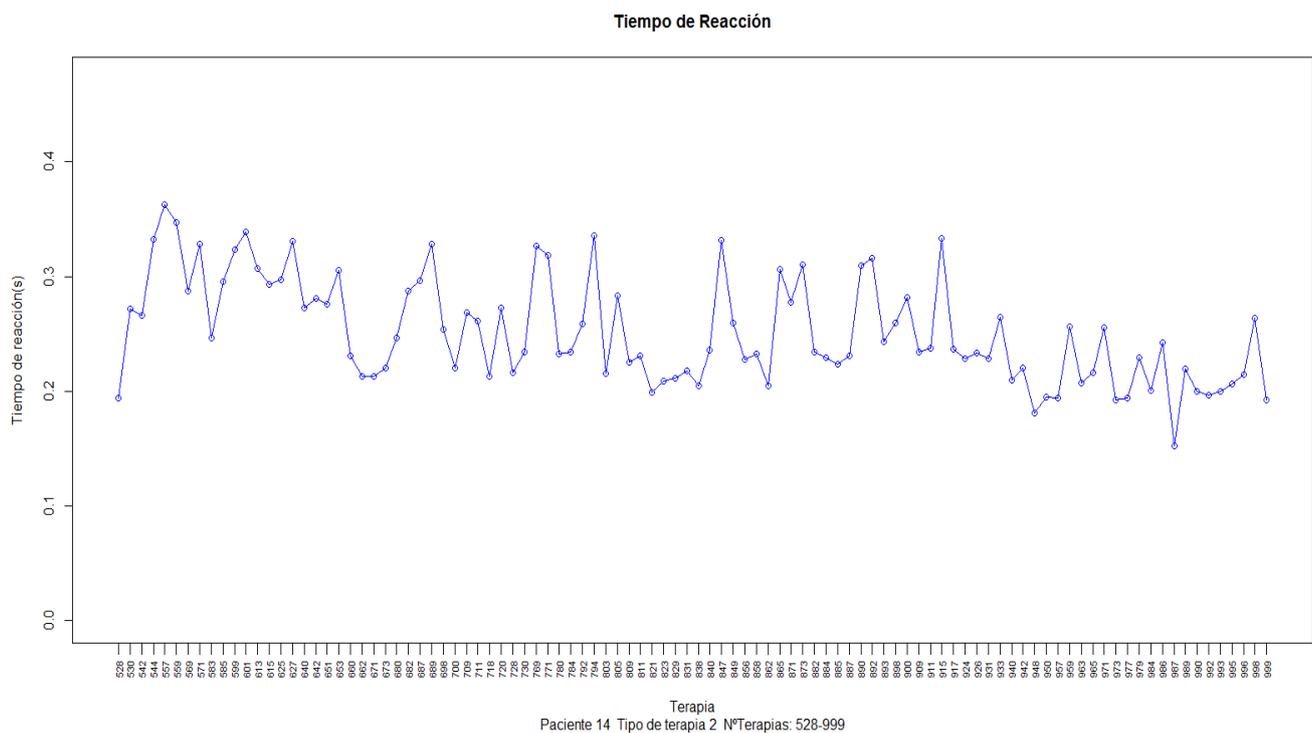


Fig. 29 Tiempo de reacción.

Podemos ver en la gráfica como el paciente varía entre 0.4 y algo menos de 0.2 segundo en reaccionar desde que ve en la pantalla un movimiento y lo que tarda en reaccionar para poder realizar el cambio de sentido.

- **Métrica 4: Error cometido en el eje “x” y en el eje “y”**

Vamos a visualizar la diferencia que hay en un punto entre x e y realizados por el paciente, con respecto a los puntos deseados x e y, para poder ver qué movimiento realiza mejor, para ello hacemos la media del punto que ha movido el paciente menos el punto deseado, entre el punto deseado tanto en x como en y.

Esta terapia va a funcionar correctamente tanto para pasivo guiada como para activo asistiva, la diferencia al igual que nos pasaba con la media y la desviación estándar es que el error de las terapias tipo 1 va a ser mayor que las de tipo 2 porque como hemos dicho antes hay que sumarle el tiempo en el que tarda en reaccionar la persona.

Para calcularlo en cada punto hacemos la media de todos los puntos de una terapia, empleando esta fórmula:

$$\varepsilon = 100 \cdot \frac{1}{n} \sum_{i=1}^n \left| \frac{\text{Posición deseada} - \text{Posición realizada}}{\text{Posición deseada} + \text{Posición realizada}} \right| \quad (5)$$

El índice es el error relativo en “x” y en “y”, y se medirá en %.

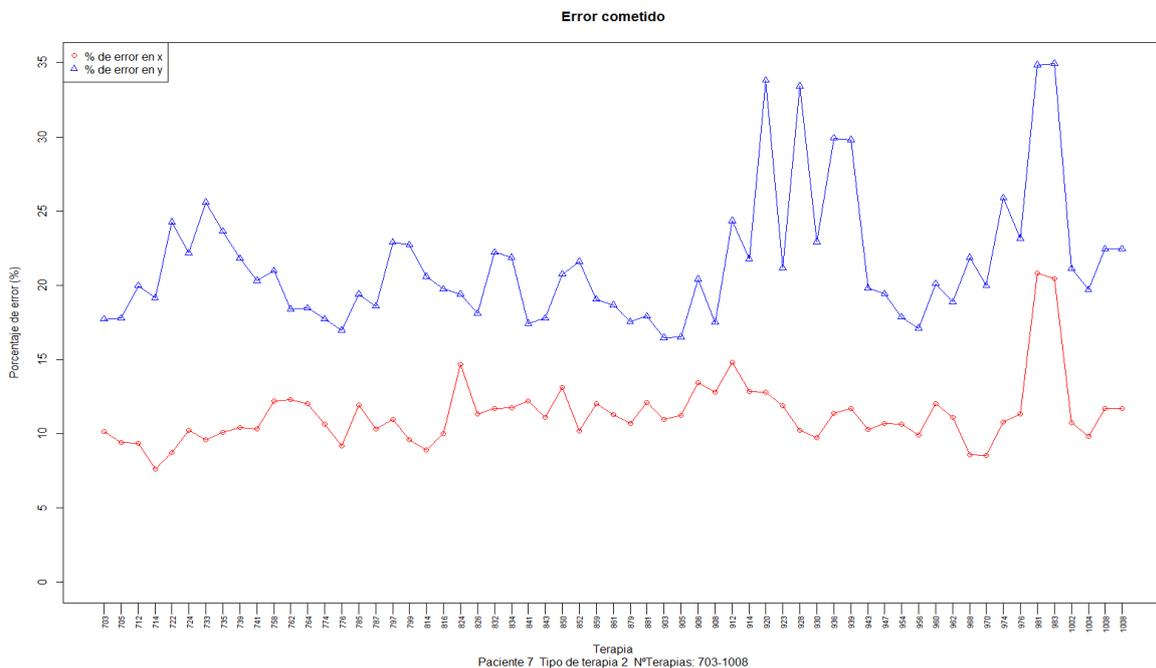


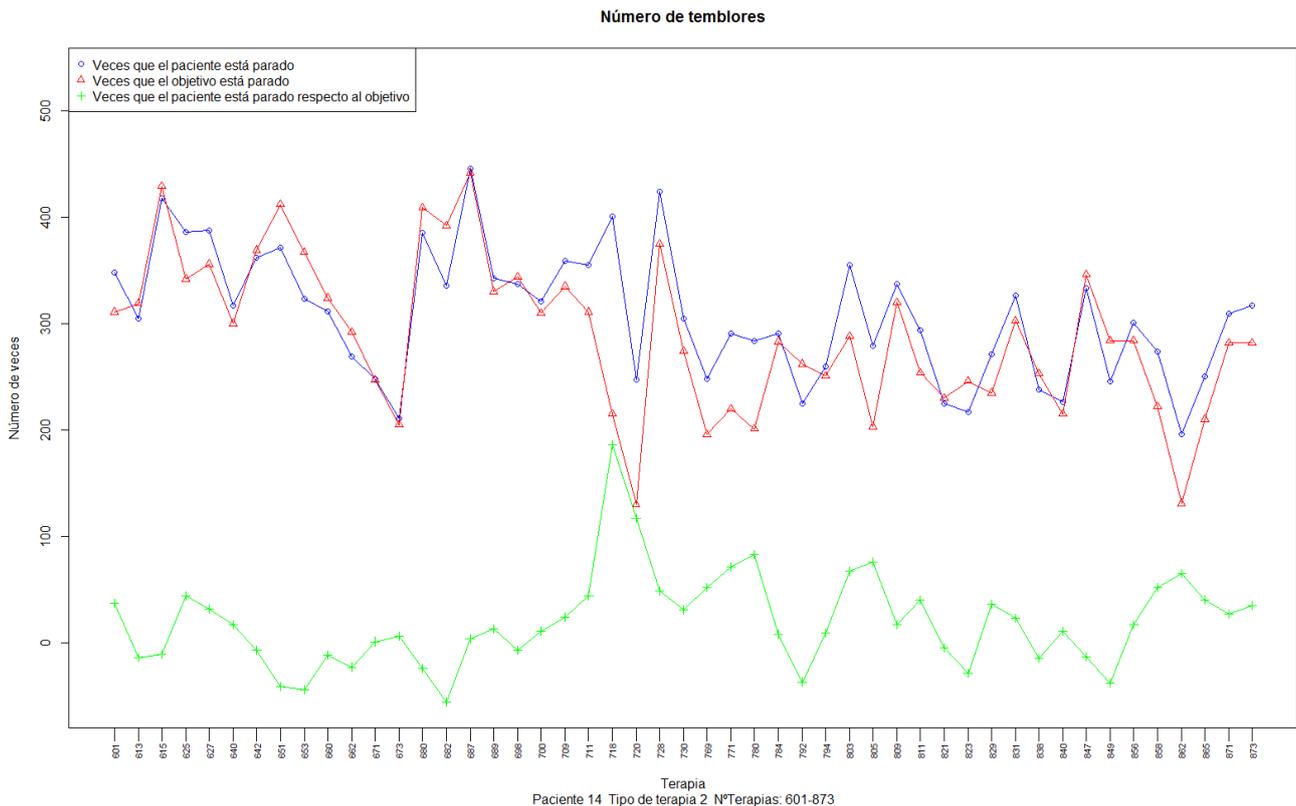
Fig. 30 Error cometido en el eje “x” y en el eje “y”

- **Métrica 5: Número de temblores.**

Vemos cuantas veces ha estado parado el robot cuando era manejado por el fisioterapeuta o cuantas veces el ratón en la pantalla ha estado en una posición fija, y cuantas veces ha estado parado el paciente, así como la comparación de las mismas. Esto lo calculamos restando una posición con su anterior, si da 0 es que no se ha movido, si hay varios ceros seguidos lo registramos como sólo un cero porque ha estado parado una sola vez, y ese número de ceros es las veces que ha estado parado, que las contamos y las llamamos número de veces.

Esta gráfica se puede emplear para ambas terapias pero el número de temblores de la terapia activo asistiva es mucho menor que la pasiva guiada debido a que el seguir el ratón produce hacer muchos más giros y por lo tanto al seguir el ratón será menor el número de veces que el objetivo y el paciente estén parados.

El índice de es el número de veces que el robot está parado.



31. Número de temblores.



- **Métrica 6: Simetría direccional en el eje “x” y en el eje “y”.**

En el análisis estadístico de una serie temporal y en el procesamiento de la señal, la simetría direccional es una medida estadística de rendimiento de un modelo para predecir el cambio, positivo o negativo, de una serie temporal de un período de tiempo al siguiente.

Para calcular la simetría direccional a partir de los puntos deseados y los puntos que ha realizado el paciente, en nuestro caso utilizamos el algoritmo definido en [20].

Vamos a analizar el código que empleamos y vamos a ver cómo funciona en R para poder explicar mejor la simetría direccional.

```
m<-(length(px))
dummyx <-diff(px)*diff(pxf)
dx  <- (1:m)*0
jx  = which(dummyx >= 0)
dx[jx] = 1
dx[is.na(dx)] <- 0
dsx[i]<-sum(dx)*100/m
```

Ese código está aplicado sólo en el eje de las x pero el código para el eje de las y será igual, primero miramos la longitud del vector de datos que vamos a crear y que necesitaremos a posterior, luego en la variable dummyx aplicamos la operación diff() tanto en los puntos realizados en x como a los puntos deseados en x y lo multiplicamos, como ya explicamos anteriormente el comando diff(), resta los números del propio vector de forma consecutiva, es decir, si tenemos un vector de 8000 números restamos el primer número con el segundo, el segundo con el tercero y así sucesivamente. Inicializamos el vector dx, y en jx introducimos un vector donde tendremos valores “NA”, que son valores indefinidos y valores mayores o iguales que cero, a continuación realizamos el sumario de todos los valores dx y les dividimos entre el número de valores total y para obtenerlo de forma porcentual lo multiplicamos por cien, ese “[i]” que aparece en dsx es debido a que todo este código está introducido dentro de un bucle for para poder tener los valores de todas las terapias.

Esta métrica de evolución se puede emplear tanto para una terapia activa asistiva como para una pasiva guiada.

El índice de la simetría direccional es un porcentaje(%).

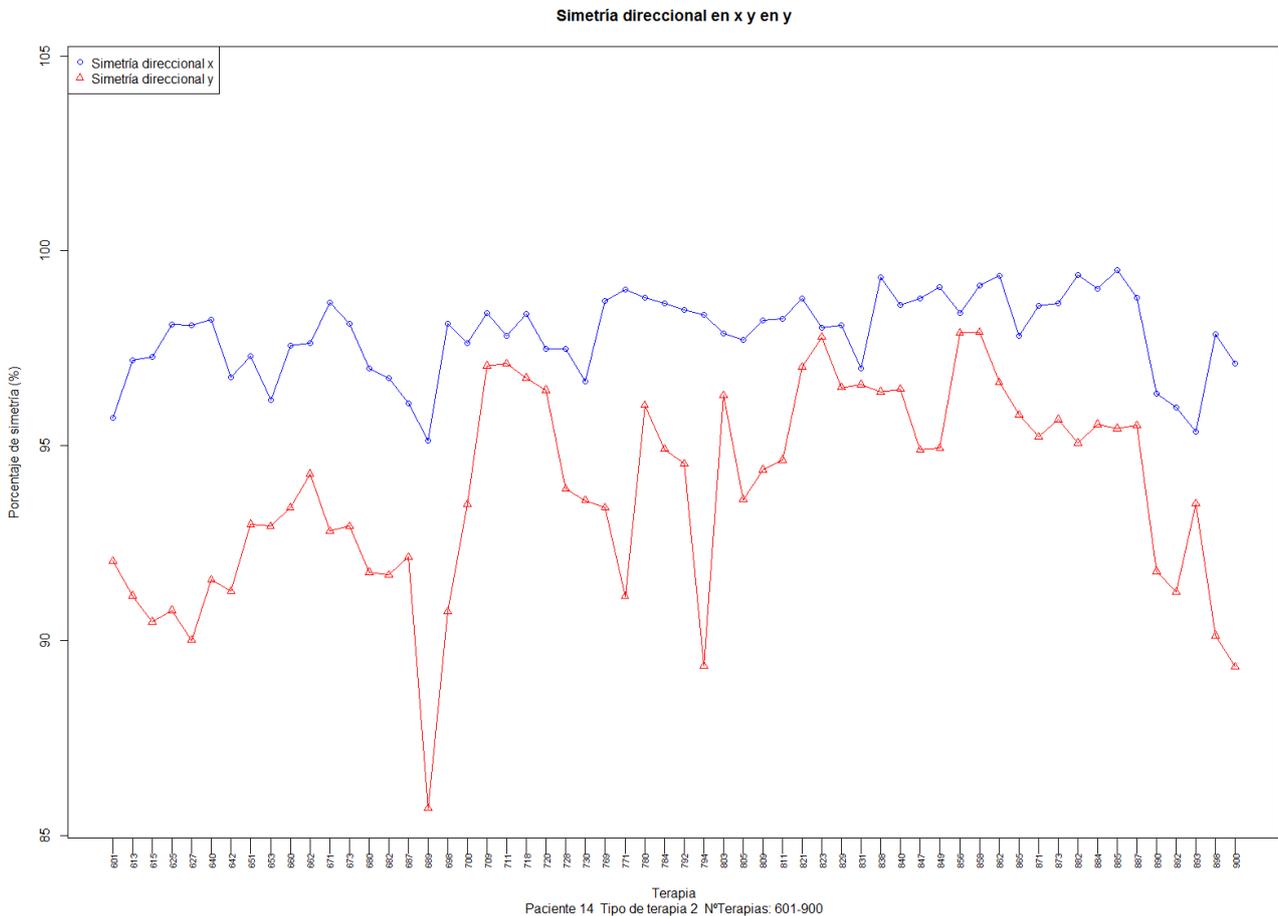


Fig 31 Simetría direccional en “x” y en “y”.

En esta gráfica podemos ver como en la simetría direccional en x está en prácticamente en todos los casos por encima del 95% esto significa que el paciente con la ayuda asistiva ha seguido muy bien los puntos previstos que se esperaba que siguiese mientras vemos que en el caso de la simetría direccional en y llega a bajar en ocasiones hasta el 85% esto se debe a que en esa terapia la persona no llegó a alcanzar tantos puntos que se había predicho que iba a alcanzar, aun así, si que ha llegado a alcanzar un gran número de puntos que se habían predicho.



- **Métrica 7: Simetría direccional de la velocidad, de la magnitud y del ángulo.**

Igual que en el caso anterior solo que en lugar de con la posición ahora vamos a emplear velocidad en x y en y, la magnitud de la velocidad es la semisuma al cuadrado de las velocidades x e y el ángulo es el arcoseno de la velocidad con respecto a la magnitud y lo que vemos en la pantalla es el porcentaje de que ángulo sea el mismo del paciente y de la trayectoria objetivo.

Los índices son la simetría direccional de la velocidad, la simetría direccional de la magnitud de la velocidad y la simetría direccional del ángulo, se mide en porcentajes de simetría (%).

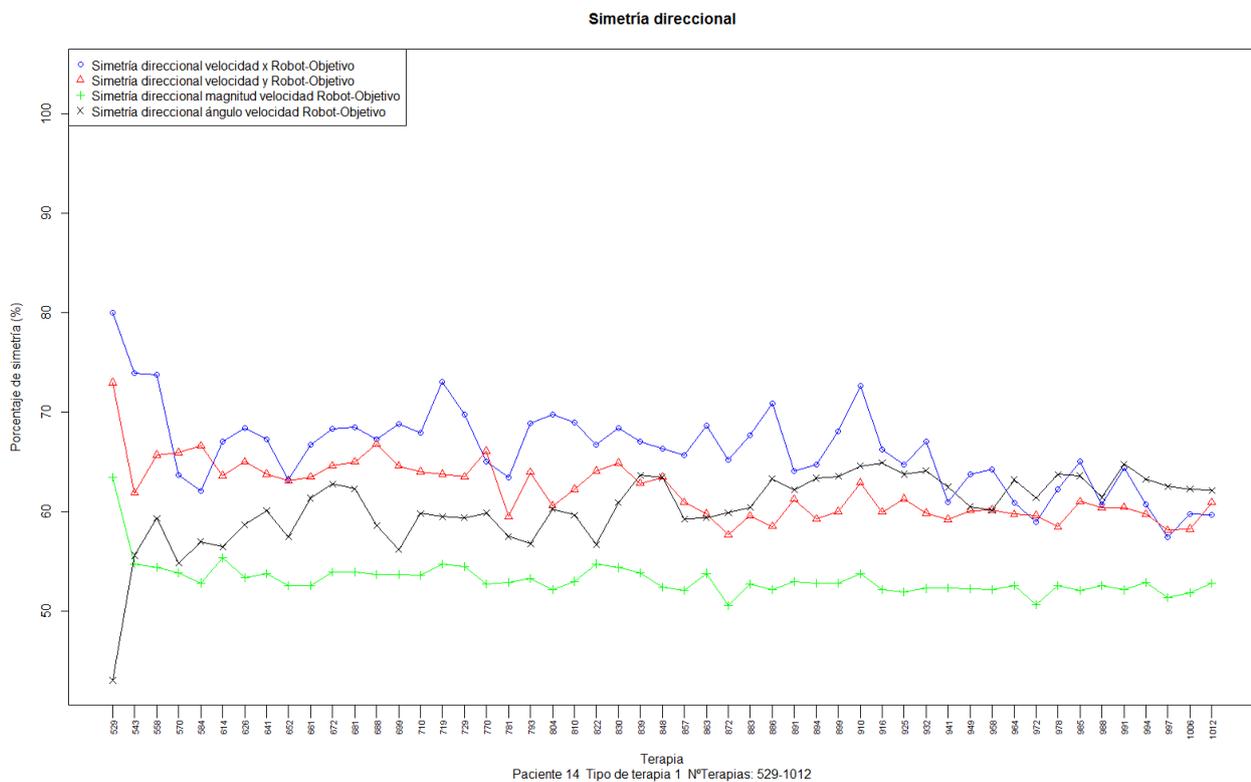


Fig. 32. Simetría direccional de la velocidad, de la magnitud y del ángulo.



• Métrica 8: Evolución de la fuerza en el eje “x” y en el eje “y”

Esta medida va a marcar la evolución de la fuerza que tenga el paciente, para ello lo que valoramos es no solo la fuerza que aplica el paciente sino también la fuerza con la que le ayuda el robot. Para calcular la evolución vamos a tomar la fuerza que aplica el paciente y la vamos a dividir entre la suma de la fuerza que aplica el robot y la fuerza que aplica el paciente.

Esta métrica funciona tanto para la terapia pasiva guiada como para la terapia activa asistida.

La ecuación que empleamos es la siguiente:

$$E. fuerza = 100 \cdot \frac{\frac{1}{n} \sum_{i=1}^n |Fuerza del paciente|}{\frac{1}{n} \sum_{i=1}^n |Fuerza del paciente| + \frac{1}{n} \sum_{i=1}^n |Fuerza asistiva|} \quad (6)$$

El índice será un porcentaje de la fuerza en x y en y que realiza el paciente con respecto a la fuerza que hace el paciente y la fuerza asistida que recibe, se medirá con un porcentaje (%).

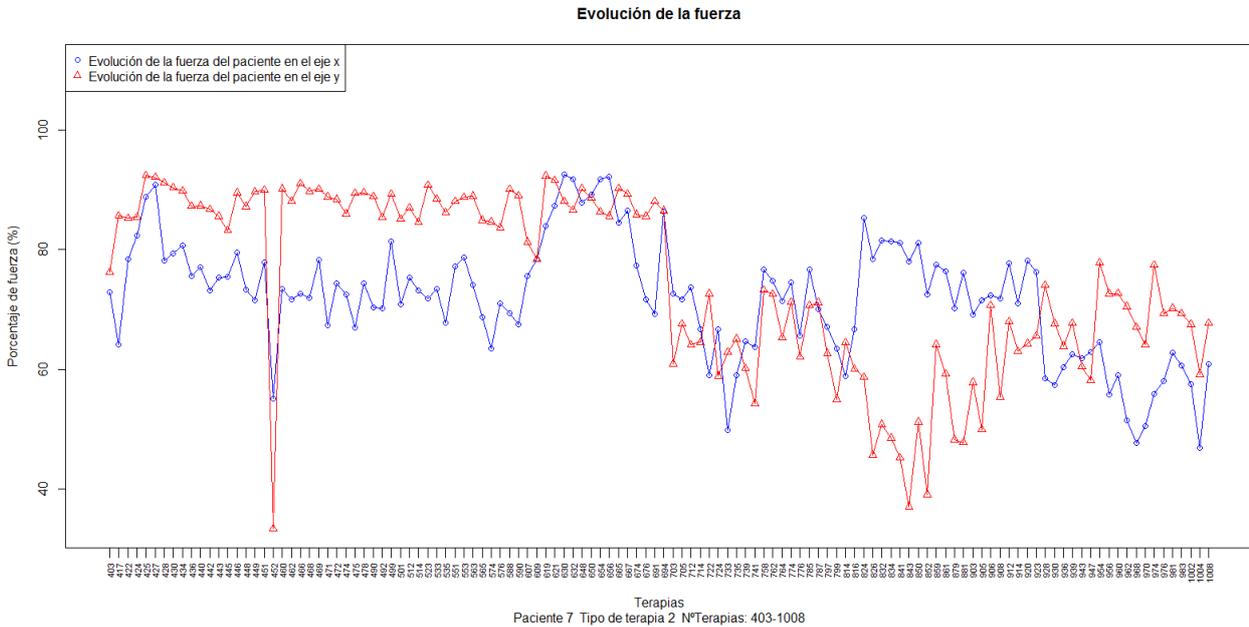


Fig. 33. Evolución de la fuerza en “x” y en “y”



- **Métrica 9: Marcador**

En esta métrica vamos a ver la cantidad de puntos que consigue el paciente en el juego del ratón, cada vez que el paciente atrapa al ratón recibe puntos de forma que cuantas más veces haya atrapado al ratón, mayor puntuación tendrá, la puntuación puede variar bastante dependiendo de la dificultad.

Este resultado está guardado como el último resultado de la columna marcador de cada terapia, el resto de números de la terapia son ceros, así que lo que hacemos es utilizamos la función *max()* en R que se encarga de buscar el valor máximo y obtenemos de esta forma el resultado.

Esta métrica sólo funciona con la terapia activa asistida dado que es la única que genera puntos de juego.

El índice de esta métrica son los puntos del juego.

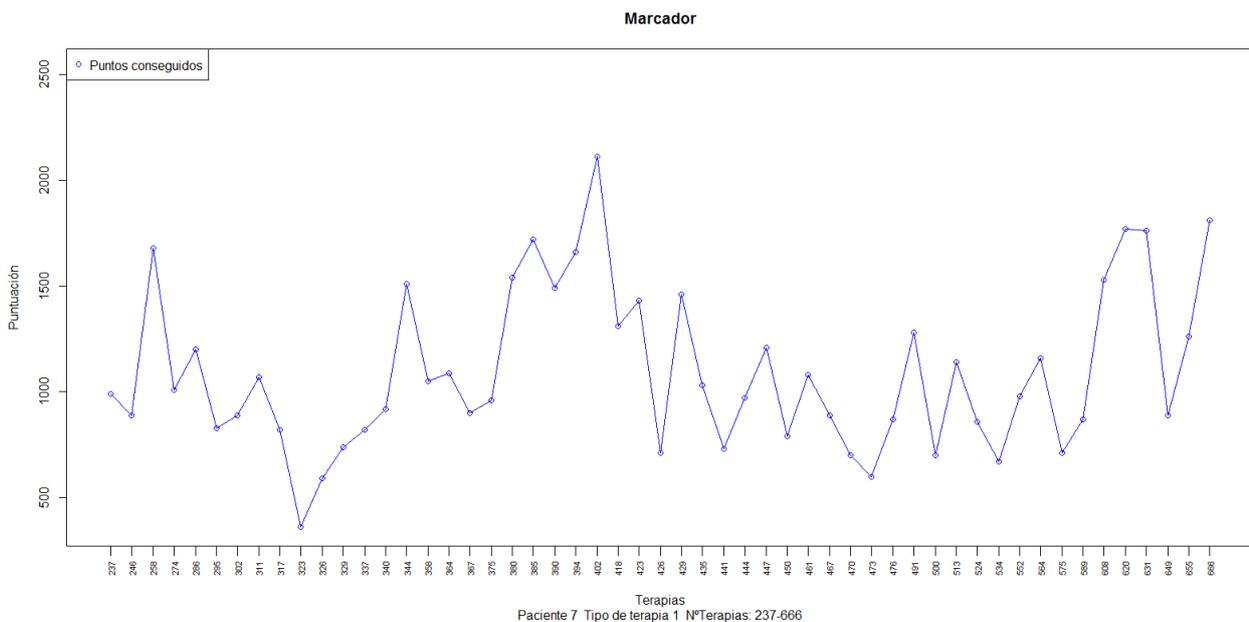


Fig. 34. Marcador



2.3. INTERFAZ Y SOFTWARE DE GESTIÓN DE INFORMACIÓN

Debido a la necesidad de que un usuario pueda acceder a las métricas sin tener conocimiento técnico ha sido necesaria crear una interfaz y para hacer su uso más transparente hemos creado un software de actualización y que así las gráficas se calculen más rápido, además para que pueda tener todas las gráficas se va a dar la opción de generar un PDF con las métricas

Para realizar el software vamos a emplear WPF (Windows Presentation Foundation) y R, WPF es un subsistema gráfico que se emplea para crear interfaces para aplicaciones creadas para Windows. Debido a que los scripts de R, que nos generan las gráficas vistas previamente no se pueden ejecutar, lo que vamos a realizar es una llamada desde WPF a los scripts de R, y desde WPF le vamos a enviar unas variables como pueden ser IdPaciente, el número de terapia, el tipo de terapias...

También vamos a obtener datos con R y se los vamos a enviar de vuelta a WPF.

Primero vamos a analizar los programas realizados con R que no se emplean para métricas pero si para el software después vamos a analizar el desarrollo y la funcionalidad de la interfaz gráfica.

2.3.1 ACTUALIZACIÓN DE FICHEROS

Debido a la gran cantidad de datos que tenemos para hacer ciertos cálculos puede tardar hasta medio minuto, es decir si quieres ver la evolución de un paciente a lo largo de un año, por ejemplo el paciente 7 tienes 200 terapias, que a su vez tienen 8000 filas cada una de esas terapias, y cada una de esas filas tiene 13 columnas, esto te acaba generando más de 27 millones de datos, que no solo lees y almacenas sino que también operas con ellos.

Para hacer transparente al usuario esta operación lo que se ha creado es un archivo de actualización que funciona de la siguiente forma:

Dado que los datos son demasiados no podemos trabajar con la tabla completa, porque la memoria RAM no puede con ello, por ello iremos cogiendo de doscientas en doscientas filas de la tabla de muestras, lo primero que realiza el programa es ver si existe un documento de texto tipo "txt" llamado Resumen si no existe, lo crea, y si existe registra el último número de terapia para seguir grabando resultados después a partir de esta, y si está llena no graba nada.

Si funciona realiza las operaciones necesarias para obtener todas las gráficas, y guarda los resultados en Resumen.txt con las siguientes columnas:



Número de terapia, Identidad del paciente, tipo de terapia, media, desviación típica, velocidad, tiempo de reacción, error en x, error en y, veces que el paciente está parado, veces que el robot está parado, diferencia entre las veces que el paciente y el robot está parado, simetría direccional en x, simetría direccional en y, simetría direccional de la velocidad en x, simetría direccional de la velocidad en y, simetría direccional de la magnitud de la velocidad, simetría direccional del ángulo, evolución de la fuerza en x, evolución de la fuerza en y y marcador.

La actualización lleva un tiempo pero luego el tiempo para la realización de las gráficas se reduce considerablemente, y da la sensación al usuario de calcularla en el instante.

Aun así la actualización es opcional, y nos puede llevar a tres casos:

- El software está actualizado: A la hora de hacer la gráfica, el script de R no se conecta con el servidor y obtiene los datos que necesita de Resumen.txt
- El software está desactualizado: A la hora de hacer la gráfica, el script de R obtiene los datos que necesita de Resumen.txt, y después se conecta con la base de datos para obtener el resto.
- El software no está actualizado: A la hora de hacer la gráfica, el script de R no encuentra el fichero Resumen.txt y se conecta a la base de datos para obtener todos los datos que necesita.

Para conseguir esto necesitamos dos scripts de R uno que llamaremos “actualización” que cree un documento llamado Resumen.txt y que se encargue de rellenarlo con los datos correspondientes, y otro llamado “comprobar actualización” que sea capaz de leer o verificar que existe Resumen.txt y comprobar que está actualizado hasta la última terapia o cuantas terapias le quedan para informar al usuario y que este pueda tomar una decisión.

Actualización:

Vamos analizar el código del script, nada más empezar empleamos el comando “gc()” que se encarga de liberar espacio en la RAM. Después empleamos el siguiente código:

```
if(file.exists(direccion)){
  existe<-read.table(direccion, sep="\t",col.names=mydata)
  attach(existe)
  nmin<-max(nterapia)
}else{
  file.create(direccion); nmin<-0}
```



Este fragmento de código que tenemos al inicio lo que hace es buscar si existe el archivo Resumen.txt, en el caso de no existir crea un archivo Resumen.txt nuevo, en el caso de que existe lee hasta que terapia hay escrita y la almacena en la memoria.

Como tener la tabla entera con todos los valores ocupa demasiado espacio en la RAM y ralentiza mucho las operaciones lo que hacemos es coger los valores de doscientas en doscientas terapias hasta llegar a la final, pero si tenemos por ejemplo trescientas terapias analizaríamos primero doscientas y luego cien y para ello empleamos este código:

```
if(((max(vector.terapiagrande))-nmin)<200){  
    nmax<-max(vector.terapiagrande)  
}else{  
    nmax<-nmin+200  
}
```

De esta forma en el caso de que el valor sea menor de doscientos al inicio de la actualización tomará como valor máximo el correspondiente, y de no ser así tomará los doscientos que queríamos.

Dentro de este script también se harán todos los cálculos de las métricas de evolución de la forma que hemos explicado en el apartado 2.2.2.

Vamos a analizar un problema que nos ha surgido, cuando con el fichero ya está actualizado y no hay ninguna terapia más en la base de datos y pulsamos actualizar, esto nos produce una línea adicional de valor "NA", es decir indeterminado, y que a la hora de buscar la terapia máxima nos da error, para arreglarlo hemos empleado el siguiente código.

```
if(is.na(Paciente[length(i)])==FALSE){  
    write(datos, file = direccion,sep = "\t",ncolumns=21,append=TRUE)  
}else{  
    break  
}
```

Esta es la instrucción que sirve para escribir en el documento Resumen.txt escribe las 21 columnas en un orden determinado cada una de esas columnas pertenece a una de las



rectas que hemos visto en el apartado 2.2.2., no a cada una de las métricas, sino a cada una de las rectas que hay en cada métrica, además añadiendo ese if else, lo que nos dice es que si va a escribir algún dato de tipo “NA” en lugar de escribirlo que haga un break.

Además al hacer esta actualización todos los scripts de métricas deben de añadir un encabezado para poder acceder a Resumen.txt que se realiza con el siguiente código:

```
if(file.exists("Resumen.txt")){
  existe<-read.table("Resumen.txt", sep="\t",col.names=mydata)
  attach(existe)
  if(max(nterapia)>nmin){
    nmmax<-max(nterapia)
    m<-subset(existe, tipoterapia == terapiatipo & nterapia>nmin & nterapia<=nmax
& paciente==idpac)
    attach(m)
    Evolutionxact<-Evolucionx
    Evolutionyact<-Evoluciony
    nmin<-max(nterapia)
    vector.terapiaact<-nterapia
    rm(m,existe)
  }
  }else{ Evolutionxact<-0
    nmmax<-nmax
  }
```

En este caso hemos empleado la evolución de la fuerza en el eje x y en el eje y, podemos ver que lo primero que hace es comprobar si el archivo Resumen.txt existe si existe extrae los datos que necesita de él y los adjunta y guarda el valor máximo de forma que si el valor máximo es menor que el que te piden a la hora de calcular la métrica sólo actuaría con los datos del fichero de actualización, si no existe fichero de actualización entonces tomaría los datos de la base de datos, y se podría producir también un caso mixto donde tomase parte de los datos del fichero de actualización y parte de la base de datos.

Comprobación de actualización

Este archivo se encarga de comparar la cantidad de terapias sin actualizar que tenemos para mandarle al usuario un mensaje diciendo si necesita actualizar y en el caso de



tener muchas decirle que debe actualizar lo más rápido posible para ello lo que se realiza es una consulta a la base de datos dónde se pregunta cuál es la última terapia que ha sido realizada y se compara con la última terapia del fichero Resumen.txt

Para ver mejor como funciona analizamos el código:

```
BBD<-max(ldTerapia)
...
if(file.exists(paste(args[1],"Resumen.txt", sep = ""))){
    existe<-read.table((paste(args[1],"Resumen.txt", sep = "")),
    sep="\t",col.names=mydata)
    attach(existe)
    actualiza<-max(nterapia)
}else{
    actualiza<-0
}
resultado<-BBD-actualiza
print(resultado)
```

Podemos ver como obtiene de la base de datos la máxima identidad de terapia, luego comprueba que el archivo Resumen.txt existe de existir guarde el mvalor de la máxima terapia en la variable actualiza, y si no existe guarda un cero en actualiza, luego imprime el resultado, que consiste en la máxima identidad de terapia de la base de datos menos el variable actualiza.

2.3.2 GENERACIÓN DE INFORMES EN PDF

Se va a dar la opción de imprimir en formato PDF la evolución de las terapias, para ello les vamos a dar a elegir al usuario que gráficas de evolución quiere ver y para qué tipo de terapia de terapia, puede elegir varias gráficas de su elección y si quiere que sea de una terapia tipo de 1 o una terapia de tipo 2 o si quiere ambas, en anexos se incluye un informe PDF.

El problema que tuvimos en la creación del PDF es que no teníamos ningún editor de texto que nos pudiese servir de forma automática, podríamos haber empleado LaTeX, pero tendríamos que bajarnos he instalarnos el programa en cada uno de los ordenadores del robot y configurarlos por lo que podría ser un coste muy pesado de instalación. Otra opción era emplear WPF como editor de textos pero deberíamos de crearnos el editor de texto al completo,



y finalmente la opción que hemos empleado que ha consistido en fabricarnos nuestro propio editor de texto en R, esto nos ha supuesto una serie de complicaciones y los informes impresos de esta forma no tienen un encuadre perfecto, además hemos tenido que incrustar imágenes y más cosas que explicaremos más adelante.

Dado que imprimimos una cantidad elevada de datos y gráficas necesitamos tener el archivo de actualización completo porque si no tardaría mucho tiempo en crear el PDF, podría tardar más de 20 minutos si el archivo está sin actualizar.

A continuación vamos a analizar el código que empleamos:

```
mydata<-
c('nterapia','paciente','terapiatipo','ECMact','STDact','ve','t_reaccion','error_x','error_y','v_pac_par
ado','v_rob_parado','v_pac_rob','sd_x','sd_y','sd_vx','sd_vy','sd_magnitudvel','sd_angulo','Evoluci
onx','Evoluciony','marca')
if(file.exists(paste(args[1],"Resumen.txt", sep = ""))){
  existe<-read.table((paste(args[1],"Resumen.txt", sep = "")), sep="\t",col.names=mydata)
  attach(existe)
  if(max(nterapia)>nmin){
    nmmax<-max(nterapia)
    m<-subset(existe, tipoterapia == terapiatipo & nterapia>nmin & nterapia<=nmmax
& paciente==idpac)
    attach(m)
    mediaMar<-marca
    mediaMar[is.na(mediaMar)] <- 0
    errorx<-error_x
    errory<-error_y
    ECM<-ECMact
    ...
    v<-ve
    MagPosObjiCero<-v_rob_parado
    MagPosCeroCompa<-v_pac_rob
    dsx<-sd_x
    Evolutiony<-Evoluciony
    rm(m,existe)
  }
  }else{ print("La base de datos no está actualizada")
}
rm(mydata)
```



En este archivo vemos como comprobamos si el archivo Resumen.txt existe, y de ser así lo que realiza es guarda todos los datos del archivo que estén entre las terapias máximas y mínimas que buscamos en distintos vectores, en el caso de no existir nos dice que la base de datos no está actualizada.

Vamos a ver el código para poder editar el texto, tomamos por ejemplo la parte en la que imprimimos el marcador.

```
#Graficamos
layout(matrix(c(1,1,2,2), 2, 2, byrow = TRUE),
widths=c(2,1), heights=c(0.9,1.2))
plot(0:20, type = "n", xaxt="n", yaxt="n", bty="n", xlab = "", ylab = "")
rasterImage(img, 8, 17, 14, 20, interpolate=FALSE)
text(11,16,"-----")
text(11,14,paste("Paciente",idpac,"Marcador Terapia",tipoterapia),font = 30,cex=2)
text(8,12,"Fecha de impresión:")
text(13,12,Sys.time())
text(11,11,"-----")
text(11,10,"En esta métrica vamos a ver la cantidad de puntos que consigue el paciente en el
juego del ratón,")

text(11,2,paste("Última puntuación      Fecha:",fecha[length(mediaMar)],"Terapia
número:",vector.terapia[length(mediaMar)],"Puntuación:",mediaMar[length(mediaMar)]))
text(11,1,paste("Puntuación Media:",ceiling(Marmed)),cex=1.2)
x <- c(1,max(length(vector.terapia)))
y <- c(min(mediaMar),1.2*max(mediaMar))
plot(x, y,type="b", pch=21, col="white",xaxt="n", lty=3, xlab="Terapias", ylab="Puntuación")
axis(1,at=c(1:max(length(vector.terapia))),labels=round(vector.terapia,digits=2), las=2,
cex.axis=0.7, tck=-.02)
lines(c(1:max(length(vector.terapia))),mediaMar[1:length(vector.terapia)],
type="overplotted",pch=1, col="blue")
main<-"Marcador"
sub<-paste("Paciente ",idpac," Tipo de terapia ",tipoterapia," N°Terapias: ",vector.terapia[1],"-
",vector.terapia[length(vector.terapia)],sep="")
title(main, sub)
legend("topleft",legend=c("Puntos conseguidos"),box.lty=1,pch=c(1),col=c("blue"))
rm(Marmax,Marmin,Marmed,aa,maxaa,minaa,main,sub,x,y)
```



}

Para escribir el documento lo que hacemos es dividimos un A4 en dos matrices, una superior y otra inferior mediante los dos primeros comandos, luego empleamos un comando `plot()` para poder dividir la matriz que hemos creado en un número de espacios determinado en nuestro caso de cero a veinte, la siguiente instrucción `rasterImage()`, lo que hace es introducir la imagen en los márgenes que les damos es decir entre la línea 20 y la 17 en el eje de las y y entre la 8 y la 14 del eje de las x, de esta forma podemos cambiar el tamaño a una imagen que ya tenemos ensanchándola o estirándola.

Con el comando `text()` introducimos un texto en una línea, el primer número es para la posición en el eje de las x y el segundo en el eje de las y, la primera línea con texto nos vale como título donde nos dice que paciente es, que métrica estamos viendo y que tipo de terapia es, al tener como opciones en `text()` “font = 30”, hace que sea más grande, el comando `Sys.time()`, sirve para obtener la fecha en el momento de la impresión, después de eso acabamos el encabezado, y escribimos una breve descripción acerca de para que sirve la métrica, a continuación escribimos las terapias más significativas y la fecha en la que fueron hechas, y calculamos la media de todas las terapias seleccionadas.

En la siguiente matriz realizamos la gráfica, los comandos que vemos en el código son para escribir título, subtítulo, leyenda y la para pintar la gráfica.

Para poder realizado el tratado de imagen hemos requerido usar la librería PNG [21], para poder emplear el comando `rasterImage()`, y así poder introducir imágenes.

Tenemos otro script en R que se llama Imprimir que sirve para imprimir un pdf con un funcionamiento muy similar al que hemos visto en el informe, sólo que esta vez lo que se va a imprimir son dos imágenes, una que es una trayectoria y otra que es una métrica de análisis individual, en una sola hoja en un pdf, salvo en el caso de que queramos comparar dos terapias en cuyo caso nos imprimirá dos hojas en tamaño a4 cada una con una terapia y una métrica correspondiente a lo que se va a comparar.

2.3.3 INTERACCIÓN ENTRE R Y WPF

Para comunicarnos con R desde WPF necesitamos crear una función C# que se llama “Rscriptrunner” a la cual vamos a ir llamando durante todo el código para poder comunicarnos con los scripts de R de forma que cuando necesitemos una gráfica vamos a llamar desde WPF a C#, vamos a hacer que R imprima esa gráfica como un png, y después desde WPF vamos a incrustar esa gráfica donde corresponda, pero para ello deben de comunicarse entre ellos y especificarse el nombre de la gráfica y otros datos.



Vamos a analizar varias partes del código de comunicación entre WPF y R para poder entender mejor cómo funcionan:

```
public class RScriptRunner
{
    public string RunFromCmd(string rCodeFilePath, string rScriptExecutablePath, string args)
    {
        string file = rCodeFilePath;
        string result = string.Empty;

        try
        {

            var info = new ProcessStartInfo();
            info.FileName = rScriptExecutablePath;
            info.WorkingDirectory = Path.GetDirectoryName(rScriptExecutablePath);
            info.Arguments = rCodeFilePath + " " + args;

            info.RedirectStandardInput = false;
            info.RedirectStandardOutput = true;
            info.UseShellExecute = false;
            info.CreateNoWindow = true;

            using (var proc = new Process())
            {
                proc.StartInfo = info;
                proc.Start();
                result = proc.StandardOutput.ReadToEnd();
                proc.Close();
            }

            return result;
        }
        catch (Exception ex)
        {
            throw new Exception("R Script failed: " + result, ex);
        }
    }
}
```



}

Con esta función y a través del programa Rscript.exe se pueden ejecutar los scripts de R, los argumentos que introducimos en RunFromCmd son el documento donde se encuentra el script de R, el segundo es la localización de Rscript.exe, el tercer parámetro son los argumentos que vamos a utilizar, separamos varios argumentos mediante espacios, podemos ver como introducimos estos datos en la variable info, creamos la variable proc que es el proceso que se va a encargar de comunicarse con R dado que le manda todos los datos que queremos, y a mayores vemos como tiene un parámetro llamado result que nos va a devolver un resultado si en el script de R introducimos un print() o varios print, de ser así nos devolvería una vector con varios resultados, finalmente si fallase el script de R nos lo diría también.

Pero el documento de R no puede leer por si solo el vector que le manda necesitamos introducir un código para que pueda funcionar que es el siguiente:

```
args <- commandArgs(TRUE)

Analisis_Terapias<-as.numeric(args[1])
Comparacion_Terapias<-as.numeric(args[2])
Terapia<-as.numeric(args[5])
```

El primer comando lo que nos hace es separarnos las variables que nos manda desde WPF según los espacios de forma que ahora ocupan un espacio en un vector, pero hay otro problema que nos da los valores que mandemos como valores de tipo caracter, de forma que debemos convertir los números de nuevo en tipo numérico.

2.3.4 INTERFAZ GRÁFICA DEL SOFTWARE DEL ROBOT PHYSIOBOT

Como hemos dicho antes para la interfaz gráfica hemos empleado WPF, con ello vamos a tener un grupo de ventanas, partiendo de una ventana inicial vamos a poder acceder a varias ventanas más y en alguna de ellas vamos a poder llamar más ventanas a mayores, esta parte tiene que ser la más transparente para el usuario de forma que pueda introducir el todas las cosas que sean necesarias para obtener las gráfica y no vea cómo funciona el programa y vaya de la forma más rápida posible, por ello vamos a explicar partes del código que el usuario no ve y que son muy importantes.

Por ejemplo como antes de abrir una ventana carga datos o como se conecta con la base de datos con WPF.

Primero conectamos la base de datos con nuestra Solución en WPF de forma que nos queda la siguiente imagen:

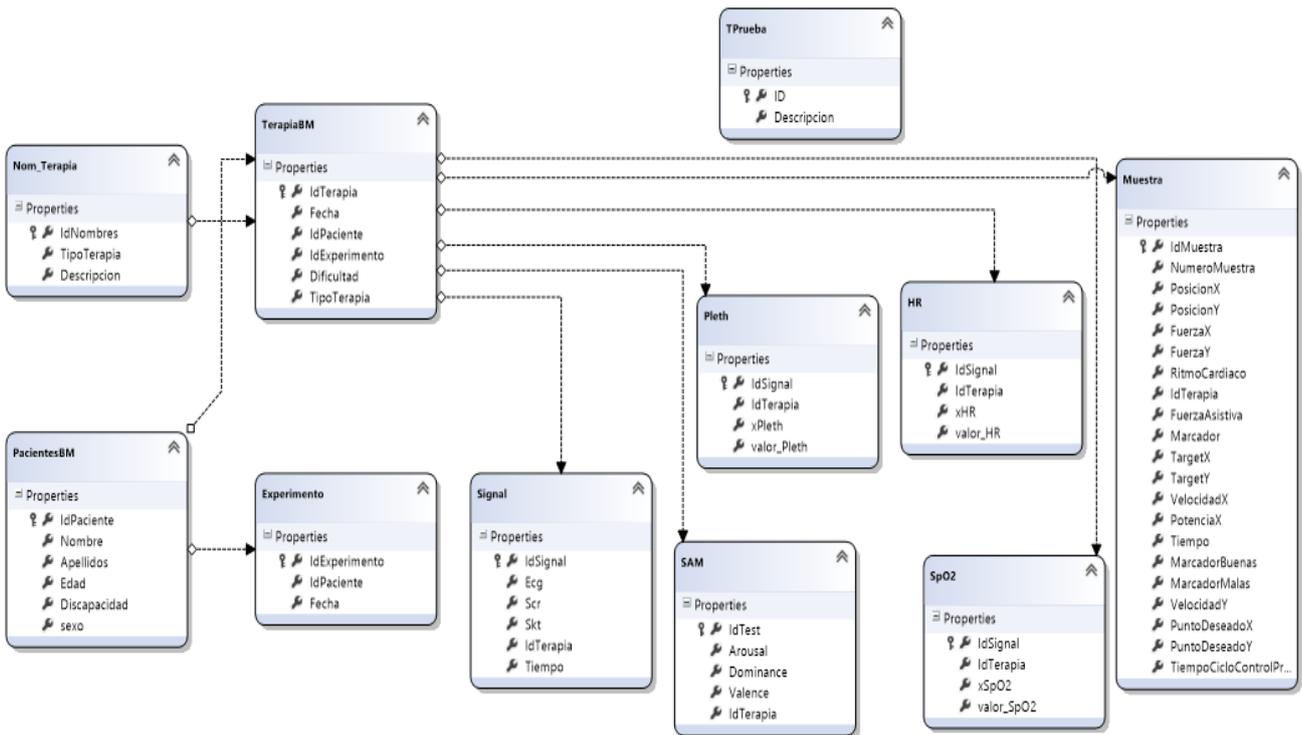


Fig. 35 Estructura de la base de datos en WPF

En el esquema de la Figura 35 podemos ver cómo están relacionadas todas las tablas de la bases de datos WPF crea un esquema interno para poder tener la estructura de la base de datos y así poder acceder a ella de una forma mucho más sencillas a la hora de hacer consultas.

Para acceder a la base de datos hemos creado una función en C# llamada funciones. Esta función es la encargada de hacer las consultas SQL para poder obtener los datos de la base de datos vamos a ver una de las consultas que tiene:



```
public List<TerapiaBM> CargarCbTerapias(string idPaciente, string TerapiaTipo)
{
    List<TerapiaBM> ListaTerapia = new List<TerapiaBM>();
    try
    {
        var resultado = (from consulta in bbdd.TerapiaBMs
                        where consulta.IdPaciente == int.Parse(idPaciente) &&
consulta.TipoTerapia == int.Parse(TerapiaTipo)
                        orderby consulta.IdTerapia
                        select consulta ).Distinct()
;
        ListaTerapia = resultado.ToList();
        return ListaTerapia;
    }
    catch (Exception ex)
    {
        return ListaTerapia;
    }
}
```

En este código es la función que nos muestra todas las terapias de un paciente y de un tipo de terapia para ello debemos de enviar los datos de que paciente es y de qué tipo de terapia vamos a realizar la consulta para que nos devuelva una lista dónde el usuario pueda elegir lo que desea.

Es una Lista lo que vamos a crear, podemos ver en la consulta como pedimos que nos de la lista de terapias que a realizado un paciente con el número que le hemos introducido y para el tipo de terapia que hemos seleccionado, hemos introducido la orden “orderby” para que nos de las terapias ordenadas de menor a mayor en lugar de que nos de las terapias ordenadas aleatoriamente, y después nos va a devolver la lista de terapias e n un argumento tipo Lista para que la podamos utilizar.

A continuación vamos a ver todas las ventanas de la interfaz gráfica de forma que podamos ver que aspecto tiene el programa y cuál es su funcionalidad.

- **Ventana de inicio**

Cuando iniciamos el programa es la primera ventana que nos aparece y nos permite acceder al resto de pantallas, este es su aspecto:



Fig. 36 Ventana de inicio

Hemos incluido el logo del proyecto y una breve descripción del robot y lo que hace, tenemos 5 botones que nos llevarán a otra ventana, el botón Actualizar que nos servirá para actualizar Resumen.txt y Salir que emplearemos para salir del programa, un detalle clave de la programación de esta ventana la vamos a ver analizando su código:



```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    RScriptRunner R = new RScriptRunner();

    string directorioScriptR = directorioImágenes + "Comprobar_Actualizacion.R";
    string parametros = directorioImágenes;
    string resultado = objetoR.RunFromCmd(directorioScriptR, rutaR, parametros);
    int Resultado_final;
    string[] Quitar_Espacios = resultado.Split(null);

    Resultado_final = int.Parse(Quitar_Espacios[1]);
}
```

Antes de que cargue la pantalla comprobamos a través del scrip de comprobar actualizaciones cuantas actualizaciones hay de diferencia entre Resumen.txt y la base de datos, y dependiendo del número de actualizaciones te irá apareciendo un mensaje u otro diciendo que deberías actualizar o que urge actualizar.

```
MessageBoxResult result = MessageBox.Show("Se tienen más de 50 terapias sin  
actualizar sería conveniente actualizar.¿Desea actualizar?", "Actualización del  
programa", MessageBoxButton.YesNo);
    switch (result)
    {
        case MessageBoxResult.Yes:
            directorioScriptR = directorioImágenes + "Actualizacion.R";
            parametros = directorioImágenes;
            resultado = objetoR.RunFromCmd(directorioScriptR, rutaR, parametros);
    }
```

En este código vemos como hemos realizado que aparezca una ventana de aviso diciendo que sería conveniente actualizar y nos da dos opciones sí o no, en el caso de que diéramos no simplemente cerraríamos la ventana, pero en el caso que diésemos sí iniciaría la actualización,



este mensaje es en el caso de que haya pocas terapias por lo que se dice sería conveniente, en caso de ser mayor de cien o de doscientas el mensaje cambia.

En esta pantalla también se encarga de mostrar un mensaje de texto cuando pulsas el botón de “Imprimir informe”, el mensaje texto te dice que si el fichero está desactualizado no vas poder tener acceso a todas las terapias sino hasta a la última que hayas accedido para imprimir el informe por el mismo motivo que hemos explicado anteriormente.

- **Analizar terapia**

La imagen que tenemos de inicio es la siguiente:

The screenshot shows a software window titled "Análisis de terapias". On the left side, there is a large, light gray button labeled "Analizar". Below this button are two smaller buttons: "Imprimir" and "Volver". To the right of the "Analizar" button, there are four dropdown menus arranged horizontally, labeled "Paciente:", "Terapia tipo:", "NºTerapia:", and "Gráfica:". Below these dropdowns, there are two radio buttons: "Valor" and "Tiempo". Under the "Valor" radio button, there is a dropdown menu labeled "Medida:". Under the "Tiempo" radio button, there are two input boxes for "De" and "a", followed by the text "segundos".

Fig. 37 Ventana de análisis de terapias 1

Mientras se cargaba la ventana lo que se ha producido ha sido una consulta a la base de datos para saber que pacientes tenemos, como las gráficas, el tipo de terapia y la medida son datos que no varían están también integrados mientras la pantalla carga los pacientes, está cargando estos valores también pero sin hacer ningún tipo de consulta.

El número de terapia no puede llegar a cargarlo hasta que no introduzcamos el Paciente y tipo de terapia.



```
private void CBTipoTerapia_SelectionChanged(object sender,
SelectionChangedEventArgs e) {
    TerapiaTipo = CBTipoTerapia.SelectedValue.ToString();
    List<TerapiaBM> L_Terapia = f.CargarCbTerapias(idPaciente, TerapiaTipo);
    CB_Terapia.Items.Clear();
    foreach (TerapiaBM item in L_Terapia) {
        CB_Terapia.Items.Add(item.IdTerapia);
    }
}
```

Existen dos tramos de código igual que este, uno que funciona para el cambio de paciente primero y luego el tipo de terapia y otro que funciona al revés, lo que hace este código es detectar que se ha introducido un valor en el tipo de terapia y como ya hemos tenido antes un cambio en el paciente, manda esos datos a hacer una consulta SQL, el comando “CB_Terapia.Items.Clear();” limpia el combo box, es decir la barra de las terapias y lo hace antes de añadir nuevos números, la borra aunque esté vacía para que si en algún caso cambias el paciente o el tipo de terapia te aparezcan los nuevos resultados.

En la siguiente gráfica vamos a ver como se cumple el código como hemos explicado:

The image displays two screenshots of a web application interface. The top screenshot shows a form with four dropdown menus: 'Paciente:', 'Terapia tipo:', 'NºTerapia:', and 'Gráfica:'. The 'NºTerapia:' dropdown is open, showing a search box and a radio button labeled 'Valor'. Below the search box, there is a text input labeled 'Medida:' and a range selector 'De [] a [] segundos' with a radio button labeled 'Tiempo'. The bottom screenshot shows the same form with 'Paciente:' set to '5' and 'Terapia tipo:' set to '1'. The 'NºTerapia:' dropdown is open, showing a list of items: 137, 138, and 218. The 'Medida:' and 'Tiempo' fields are also visible.



Fig. 38 Ventana de análisis de terapias 2

Como podemos ver salvo que no elijamos tipo de terapia y paciente el número de terapia no va a poseer ningún valor.

Una vez seleccionado Paciente, terapia tipo, número de paciente , gráfica y número de terapia vamos a tener que elegir Valor o Tiempo, en valor en las medidas tenemos máximo, mínimo y media, por valor solo podemos analizar terapias pasivo guiadas porque necesita que haya una serie de iteraciones, mientras que tiempo vale para los tipos de terapia.

El resto de botones que tenemos son Volver que sirve para cerrar esa ventana y volver a la anterior, imprimir que nos vale para imprimir las dos imágenes en un PDF, y el botón Analizar que vamos a ver cómo funciona a continuación.

Lo primero que hacemos es comprobar que se rellenan todos los campos, en el caso de no ser así mostramos un mensaje para que rellene el campo correspondiente, a mayores si el tiempo es superior a 300 segundos también nos saldrá un error, esto lo hacemos usando este código:

```
if (CB_Paciente.SelectedIndex < 0)
{
    MessageBox.Show("Por favor ,seleccione un paciente.");
}
else if (CBTipoTerapia.SelectedIndex < 0)
{
    MessageBox.Show("Por favor ,seleccione un tipo de terapia.");
}
```

Este código nos sirve para que salte un mensaje de error si falta algún combo box de seleccionar, al igual para con los radio buttons de valor y tiempo, y de forma similar al tiempo, el código es mucho más largo pero con ver este ejemplo nos hacemos una idea.

Cuando tenemos lo que ha seleccionado el usuario se implementa el código para ejecutar el script de R que necesitamos, parte del código que empleamos es el siguiente:



```
switch (CB_Graficas.SelectedIndex)
{
    //Segun el tipo de gráfica que seleccionemos llama a R y carga la
    imagen para tiempo
    case 0:

        RScriptRunner R0 = new RScriptRunner();
        string directorioScriptR0 = directorioImágenes + "Velocidadx.R";
        string parametros0 = System.AppDomain.CurrentDomain.BaseDirectory
+ " " + tiempo1 + " " + tiempo2 + " " + Terapia + " " + 0;
        string result0 = objetoR.RunFromCmd(directorioScriptR0, rutaR,
parametros0);
        CargarGrafica2(directorioImágenes + "VelocidadxTerapia" + Terapia +
".png");

        break;
    case 1:
        RScriptRunner R1 = new RScriptRunner();
        string directorioScriptR1 = directorioImágenes + "Aceleracionx.R";
        string parametros1 = System.AppDomain.CurrentDomain.BaseDirectory
+ " " + tiempo1 + " " + tiempo2 + " " + Terapia + " " + 0;
        string result1 = objetoR.RunFromCmd(directorioScriptR1, rutaR,
parametros1);
        CargarGrafica2(directorioImágenes + "AceleracionxTerapia" + Terapia +
".png");

        break;

        ...
}
}
```

Hacemos un comando de tipo switch, case , para seleccionar que tipo script de R empleamos, en este caso el script de Velocidadx.R que genera la gráfica de tiempo para ver la velocidad, una gráfica parecida a la de la figura 13, podemos ver cómo le mandamos de argumentos de entrada los dos tiempos el tipo de terapia, y el 0 es para indicar el tamaño en el cual debemos de imprimir la gráfica, el vector result0 nos da el resultado de print() que hemos introducido para R, luego la función de CargarGrafic2 es la que empleamos para poder incrustar la gráfica en el espacio correspondiente para ella, el código es el siguiente:



```
private void CargarGrafica2(string rutaGrafica)

{
    rutaanálisis = rutaGrafica;
    try
    {
        if (System.IO.File.Exists(rutaGrafica) == true)
        {

            BitmapImage _image = new BitmapImage();
            _image.BeginInit();
            _image.CacheOption = BitmapCacheOption.None;
            _image.UriCachePolicy = RequestCachePolicy(RequestCacheLevel.BypassCache);
            _image.CacheOption = BitmapCacheOption.OnLoad;
            _image.CreateOptions = BitmapCreateOptions.IgnoreImageCache;
            _image.UriSource = new Uri(rutaGrafica, UriKind.RelativeOrAbsolute);
            _image.EndInit();
            Im_análisis.Source = _image;
        }
        else
        {
            MessageBox.Show("La trayectoria en esta terapia es demasiado corta como
para poder calcular la traza realizada en una vuelta.");
            BitmapImage _image = new BitmapImage();

            _image.UriSource = new Uri(directorioImágenes + "Defecto.png",
UriKind.RelativeOrAbsolute);
            _image.EndInit();
            Im_análisis.Source = _image;
            rutaanálisis = directorioImágenes + "Defecto.png";
        }
    }
}
```



A la función le introducimos la dirección donde se encuentra la imagen, y la convertimos en un mapa de bits para poder introducirla en WPF, lo cual nos permite cargar la imagen, primero inicializamos la imagen, limpiamos e inicializamos la cache para poder introducir la URL donde está la imagen, cargamos la imagen en la caché, terminamos de hacer cálculos con la imagen e introducimos en la imagen que está vacía la imagen, en el else lo que pasa es que si falla el programa de R que suele ser porque la trayectoria en esta terapia es demasiado corta como para poder calcular la traza realizada en una vuelta, se emite un mensaje por pantalla, y se muestra una imagen blanca por defecto, que es Defecto.png.

Vamos a ver como se ve el análisis de la terapia, una vez introducimos los datos de forma correcta:

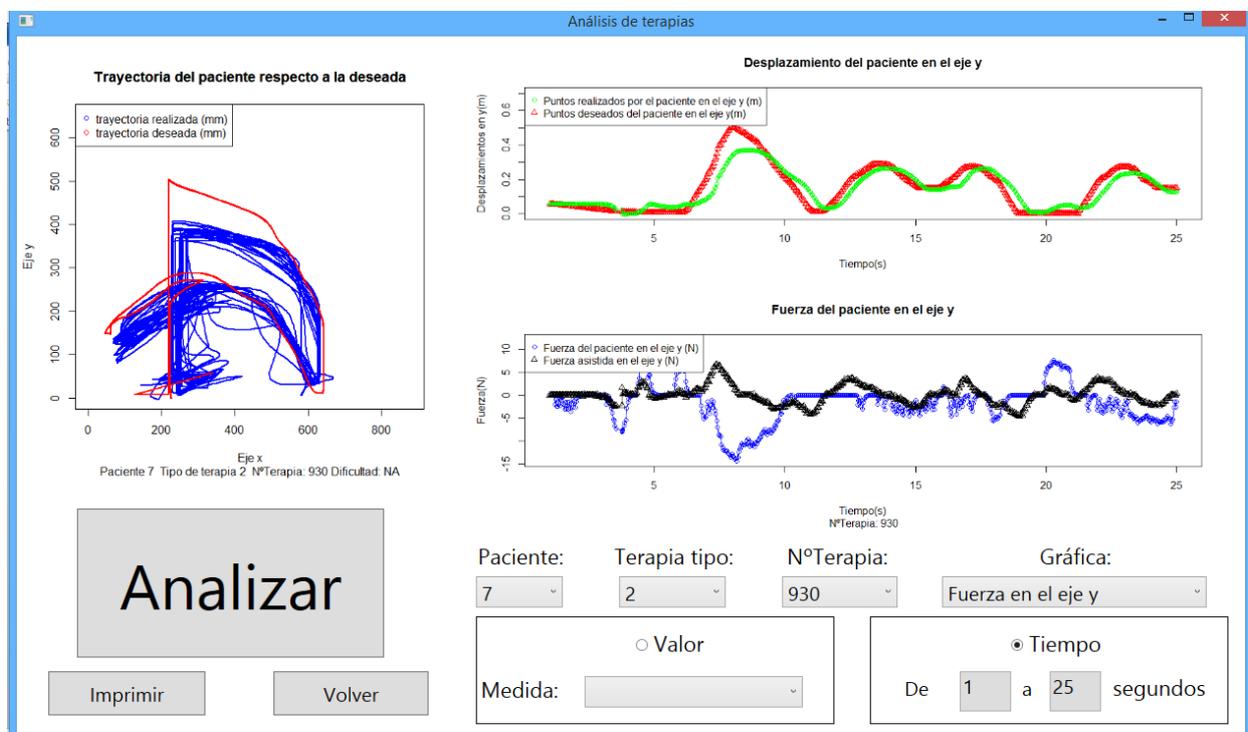


Fig. 39 Ventana de análisis de terapias 3

En la figura 39 podemos ver como se muestra una trayectoria a la izquierda y un análisis en función del tiempo de la fuerza del paciente en el eje y , como hemos explicado se han llamado a los programas de R que han creado estas imágenes y luego la hemos incrustado en el programa. Nos queda por explicar que en el caso de querer agrandar las imágenes podemos hacer click sobre ellas. Y obtendríamos las siguientes imágenes:

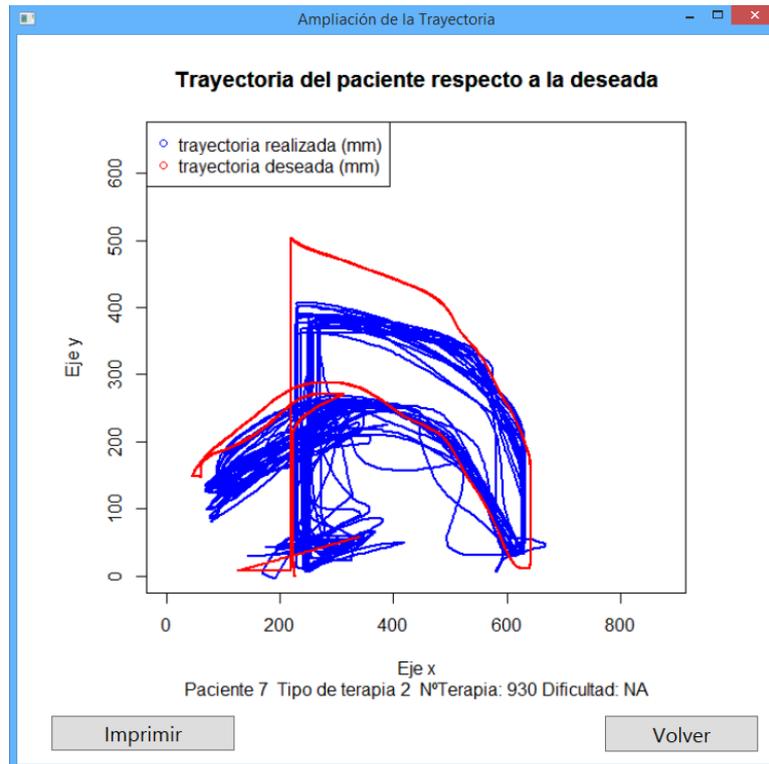


Fig. 40 Ampliación de la trayectoria

Si pulsamos el botón imprimir nos daría un png con la imagen y si damos a volver saldríamos de esta ventana.

Para llamar a la imagen con un click empleamos este código:

```
private void Im_analisis_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
{
    W_Ampliar analisis V_Ampliar analisis = new W_Ampliar analisis(ruta analisis);
    V_Ampliar analisis.Owner = this;
    V_Ampliar analisis.ShowDialog();
}
```

Podemos ver que cuando el ratón clickea la imagen con el botón izquierdo, crea una nueva ventana a la que manda una ruta de imagen donde se encuentra la URL de dónde se encuentra la imagen.

Podemos mirar el código de la nueva ventana para ver como recibe la nueva ventana la imagen:

```
string analisis;  
public W_Ampliar analisis(string ruta analisis)  
{  
    analisis = ruta analisis;  
    InitializeComponent();  
}  
private void Window_Loaded(object sender, RoutedEventArgs e)  
{  
    ImageSource imgSource = new BitmapImage(new Uri(analisis));  
    Im_Analisis1.Source = imgSource;  
}
```

Vemos como recibe la ruta analisis que habíamos mandado la guarda en otra variable y carga la imagen con el comando BitmapImage, como podemos ver este comando es mucho más fácil que el explicado anteriormente, eso se debe a que solo puede haber una imagen, si fuésemos a modificar la imagen necesitaríamos limpiar la caché de Im_Analisis1 o no se cambiaría la imagen.

En la imagen del análisis de la fuerza funciona exactamente igual que en esta si la haces click, el resultado es el siguiente:

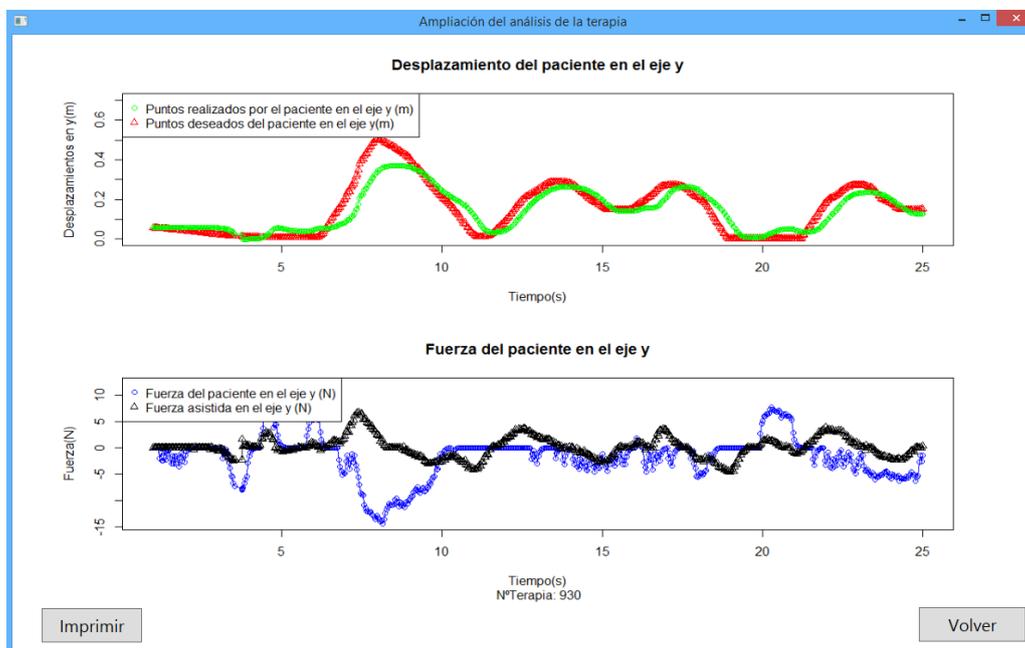


Fig. 41. Ampliación del análisis de la terapia

- Comparar terapias

Esta ventana nos va a servir para comparar dos terapias, de forma que tendremos dos gráficas de trayectoria y en paralelo la gráfica de análisis que hayamos seleccionado. La ventana que vemos cuando abrimos es la siguiente:

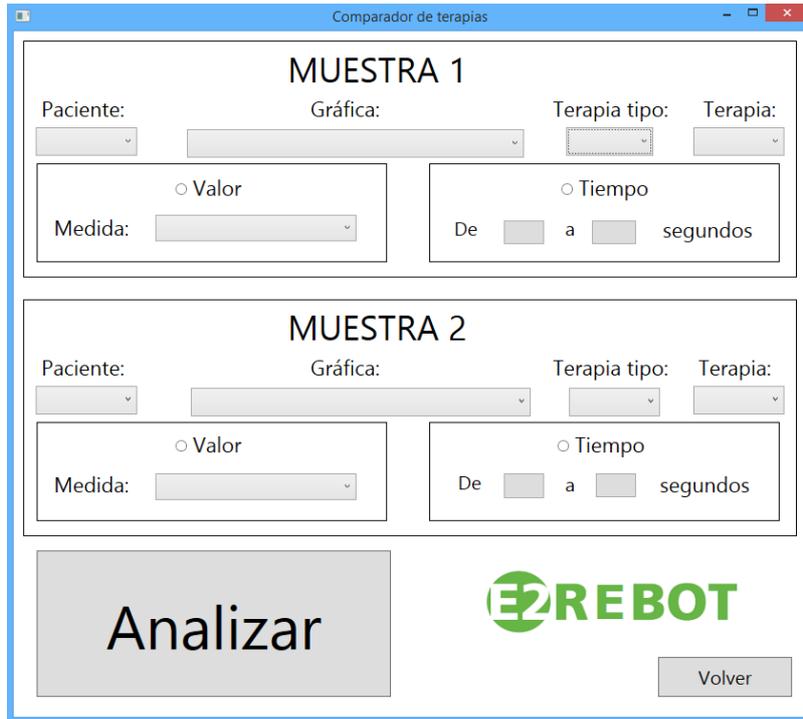


Fig. 42. Comparador de terapias.

Podemos ver cómo nos pide dos muestras y los mismos datos que en la ventana de analizar terapia. Vamos a explicar cómo funciona el tiempo, analizamos el código:

```
private void TB_tiempo inicial1_KeyDown(object sender, KeyEventArgs e)
{
    //Solo permitimos numeros en el primer text box porque no permitimos el resto
    de teclas del teclado
    if (e.Key >= Key.D0 && e.Key <= Key.D9 || e.Key >= Key.NumPad0 && e.Key <=
    Key.NumPad9)
        e.Handled = false;
    else
        e.Handled = true;
}
```

Lo que hemos realizado es solo permitir entrar números en el text box de tiempo, en el caso de que intentes introducir otra cosa, no te lo va a permitir, con el comando de Key.DO y Key.D9 permitimos los números de arriba del teclado y con Key.NumPad, permitimos la entrada de números por la parte derecha del teclado.

Vamos a rellenar de forma incorrecta los datos que nos pide para ver un mensaje de error que son los mismos que obteníamos en la ventana de análisis de terapias:



Fig. 43 Comparador de terapias con mensaje de error.

Vemos como nos da un mensaje de error debido a que no hemos elegido entre valor o terapia en la muestra dos, vamos a seleccionar un tiempo de 1 a 20 segundos también en la muestra 2 y vamos a analizar los resultados:

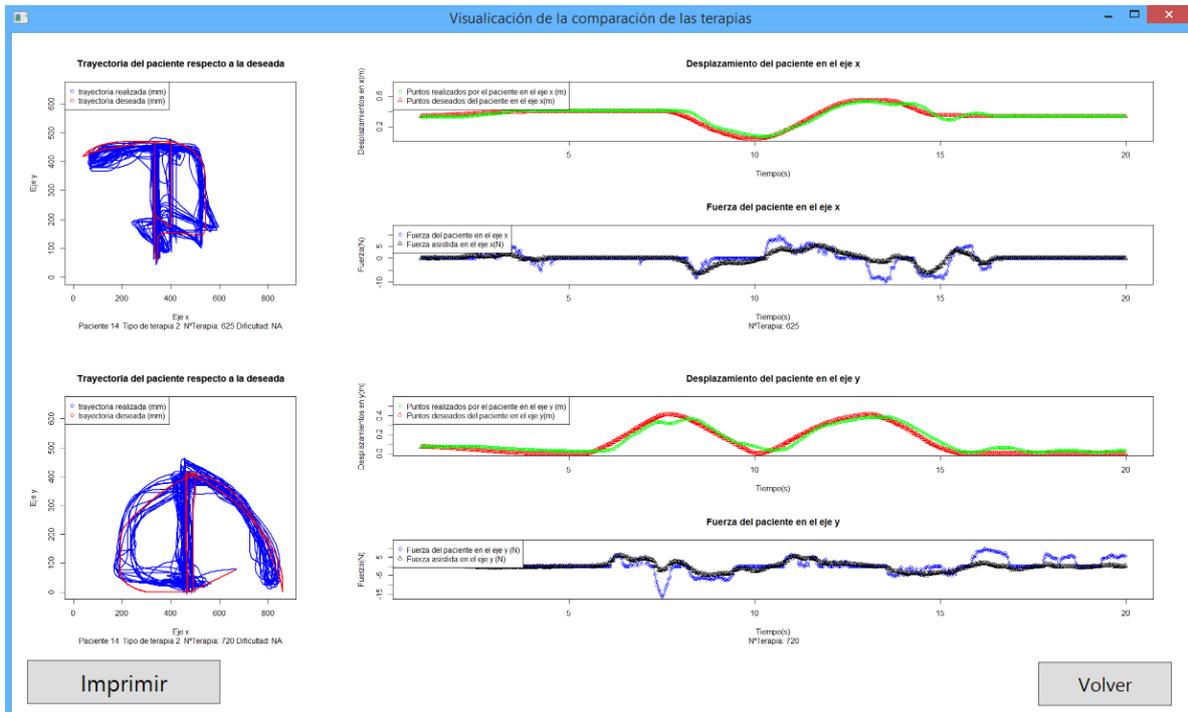


Fig. 44 Visualización de la comparación de las terapias.

Podemos ver cómo está formada esta ventana por cuatro gráficas la muestra 1 son las dos gráficas superiores y la muestra 2 son las dos gráficas inferiores.

Si pulsamos el botón imprimir se generará un PDF de dos páginas de tamaño a4 mostrando en la primera página la muestra 1 y en la segunda página la muestra 2, si pulsamos volver volveríamos a la pantalla de la figura 44 y podríamos introducir nuevos datos.

Si nos fijamos estas gráficas están más achatadas que en el análisis de terapia, pero aún así no pierden la calidad, esto se debe que en el código cuando nos comunicamos con R:

```
RScriptRunner R0 = new RScriptRunner();
string directorioScriptR0 = directorioImágenes + "Velocidadvx.R";
string parametros0 =
System.AppDomain.CurrentDomain.BaseDirectory + " " + maxima + " " + minima + " " +
media + " " + Terapia2 + " " + 2;
string result0 = objetoR.RunFromCmd(directorioScriptR0, rutaR,
parametros0);
```



Podemos ver como en el código cuando mandamos los argumentos, al final hay un 2, esto va actuar de la siguiente manera en el código de R:

```
if(as.numeric(args[6])==0){  
  png(paste(args[1],"R\\VelocidadvxTerapia",n,".png",sep=""),width = 1147, height = 756,  
  units = "px", pointsize = 16)  
}  
if(as.numeric(args[6])==2){  
  png(paste(args[1],"R\\Muestras2.png",sep=""),width = 1275, height = 450, units = "px",  
  pointsize = 16)  
}
```

Vemos como tenemos que si el argumento que tenemos es un 0, la anchura de la imagen debe de ser 1147 y su altura debe de ser 756, mientras que si es dos, su anchura y altura cambian para adaptarse mejor.

Para finalizar esta venta si clicamos en una imagen la veremos más grande y tendrá dos botones, si damos al de imprimir nos dará el png, y si damos a salir volveremos a la ventana anterior.

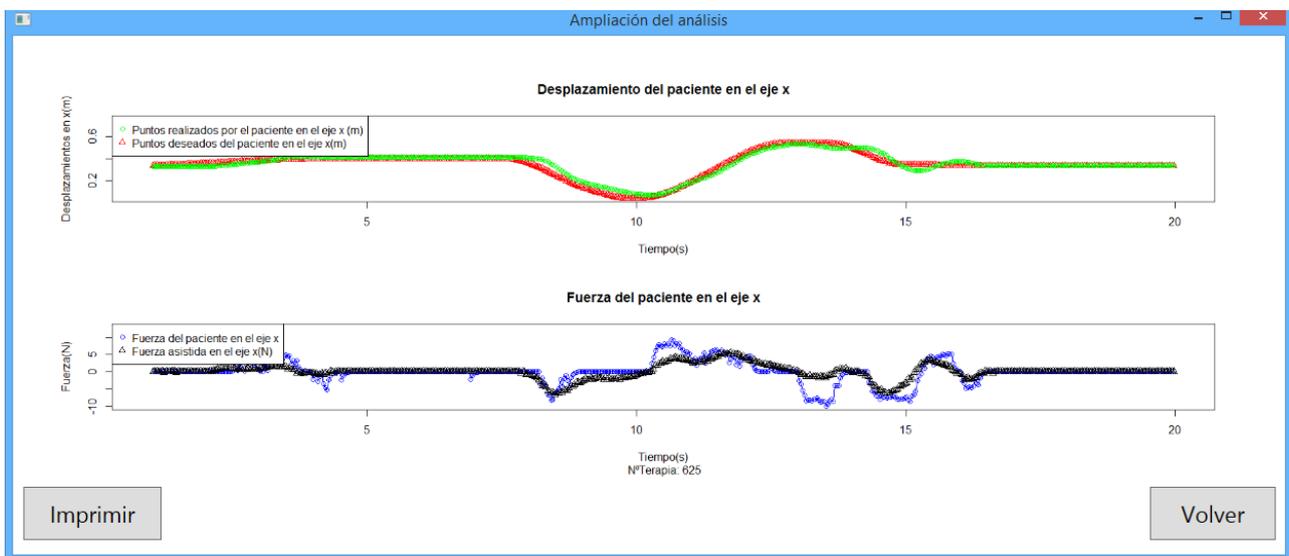


Fig. 45 Visualización de la comparación de las terapias.



- Ver evolución del paciente

En esta gráfica vamos a poder ver las métricas de evolución, analizamos la ventana para saber su funcionamiento.

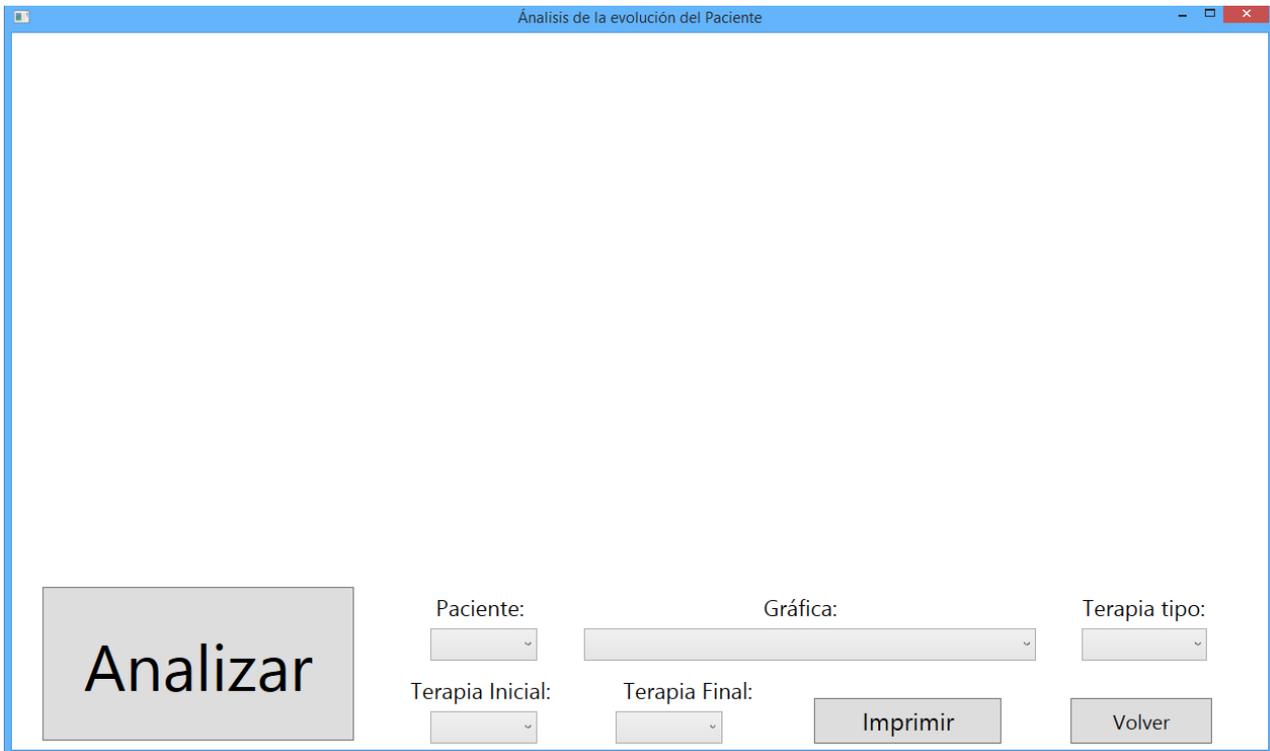


Fig. 46 Análisis de la evolución del Paciente 1.

Podemos observar como tiene una estructura similar al de análisis de terapias pero te pide que introduzcas datos distintos, ahora las gráficas son las métricas de evolución, paciente y terapia tipo son igual que en el anterior, pero al elegir terapia inicial y final cambia, la terapia inicial debe ser inferior a la terapia final, estas dos terapias se cargan una vez se haya seleccionado el paciente y la terapia tipo, al igual que vimos en análisis de terapia con el número de terapia.

Si pulsamos el botón de Imprimir nos imprimirá la gráfica en formato png, si pulsamos el botón de volver y si damos al botón de analizar se comunica con R para obtener la gráfica que pedimos.

Vamos a provocar el error al seleccionar la terapia inicial y la final para comprobar que nos da un mensaje de error:

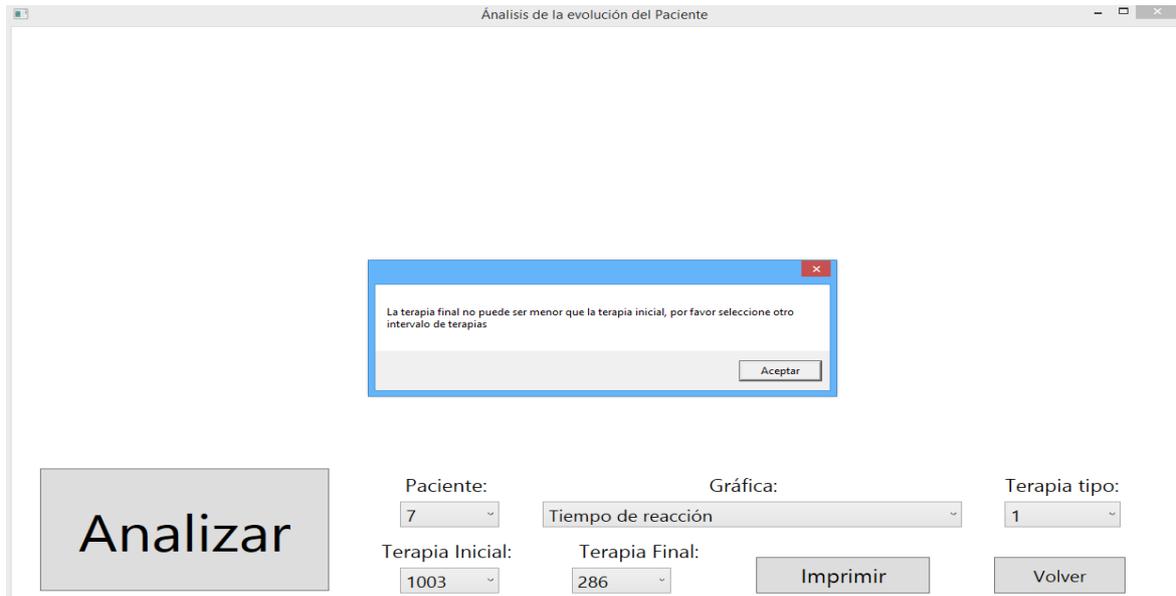


Fig. 47 Mensaje de error de análisis de la evolución del Paciente.

Si introducimos los datos de forma correcta obtenemos la siguiente gráfica:

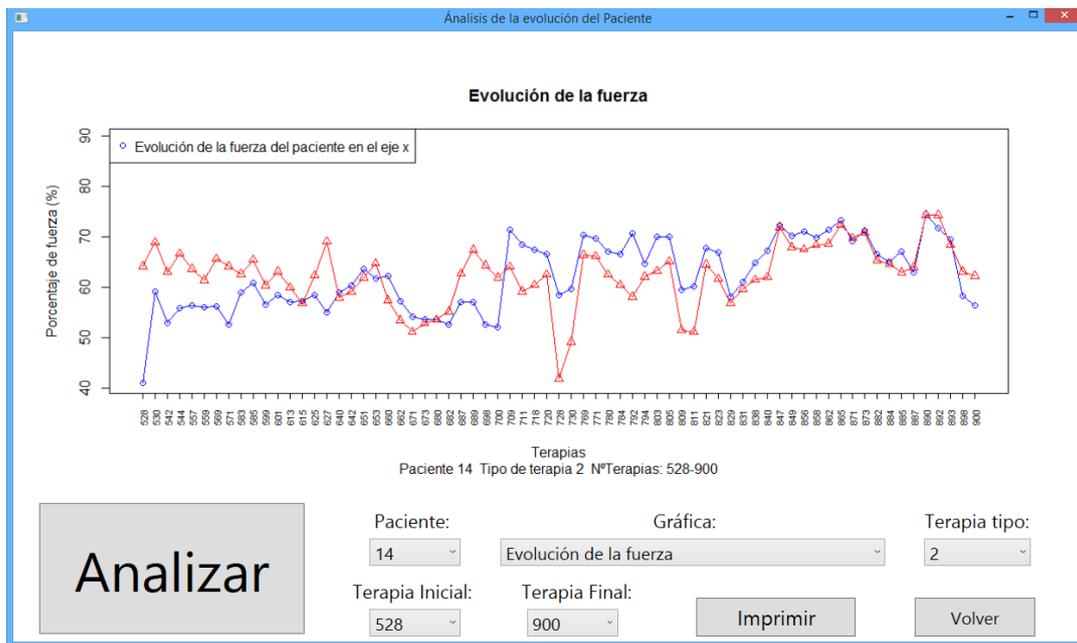


Fig. 48. A análisis de la evolución del Paciente 2.

Si clicamos las imágenes se nos ampliarán y nos dará la opción de imprimirlas al igual que vimos en el caso del análisis de terapias.



- **Comparar evoluciones**

En esta parte del programa va a ser muy parecida a la parte de comparar análisis vamos a volver a tener dos muestras y se puede ver que los datos que pide son los mismos que se piden en evolución de paciente, solo que se piden dos veces uno por cada muestra, en la primera pantalla que tenemos vemos lo siguiente:

The screenshot shows a software window titled "Comparación de evoluciones del paciente". It contains two identical sections, "MUESTRA 1" and "MUESTRA 2". Each section has five dropdown menus: "Paciente:", "Gráfica:", "Terapia tipo:", "Terapia Inicial:", and "Terapia Final:". At the bottom of the window, there is a large "Analizar" button and a smaller "Volver" button.

Fig. 49 Comparación de evoluciones del Paciente.

Tenemos también implementado todos los mensajes de errores y en esta venta al dar a Analizar lo que hará será mandarle los datos al programa de R para que este cree las gráficas y mandarle a la siguiente venta la ubicación de las imágenes para que pueda incrustarlas y así mostrárselas al usuario.

Al igual que en la ventana anterior la terapia inicial introducida debe de ser menor a la terapia final.



Si pulsamos el botón de analizar e introducimos los datos correctamente las gráficas que obtenemos son las siguientes:

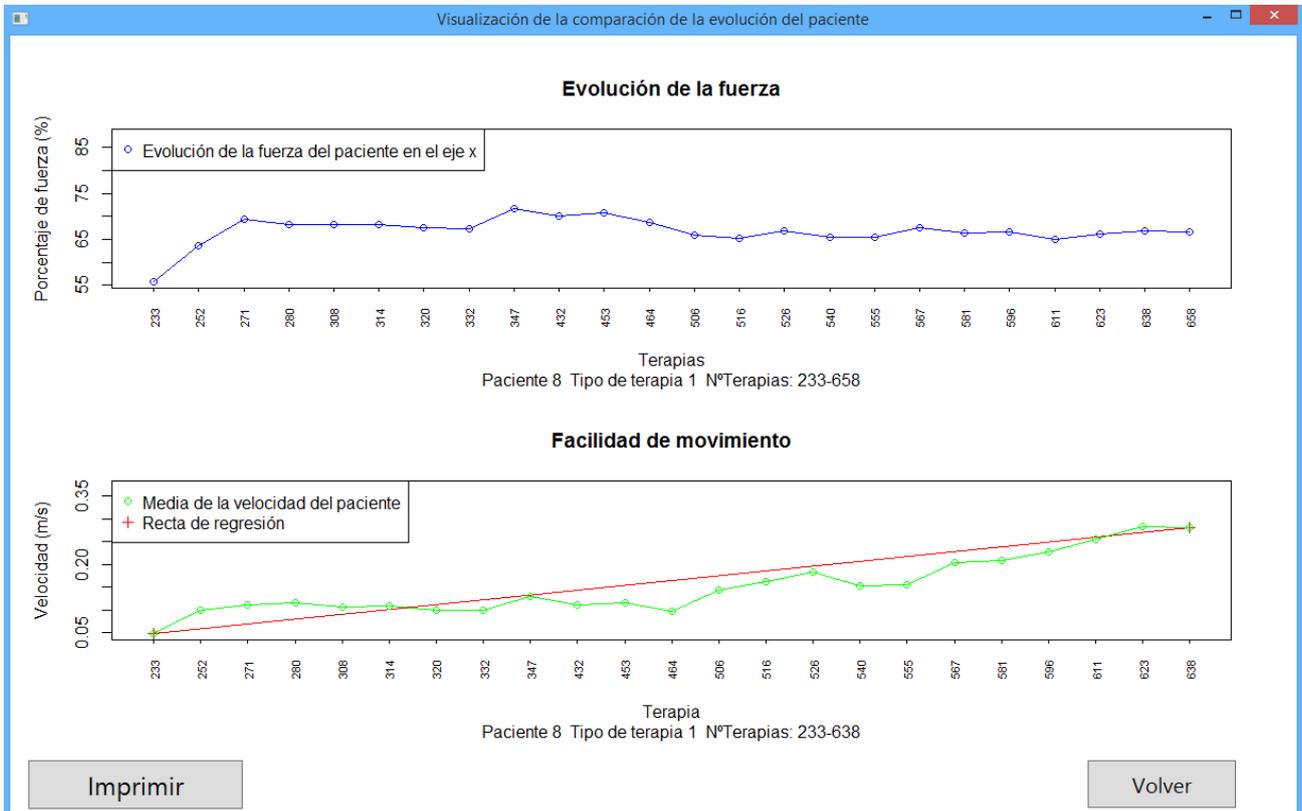


Fig. 50. Visualización de la comparación de la evolución del paciente.

En la gráfica superior tenemos los resultados obtenidos de la muestra 1 mientras que en la gráfica inferior tenemos la muestra 2, esta comparativa permite analizar entre dos gráficas de un mismo paciente o comparar cómo evolucionan dos pacientes distintos entre sí

Si pulsamos el botón imprimir nos devolverá dos imágenes .png con las métricas, y si pulsamos el botón volver saldrá de esta ventana llevándonos a la anterior.

- **Generación de pdf**

Lo que tenemos es una ventana con la posibilidad de elegir muchos campos y dependiendo que campos escojamos mandará la información al script de R donde

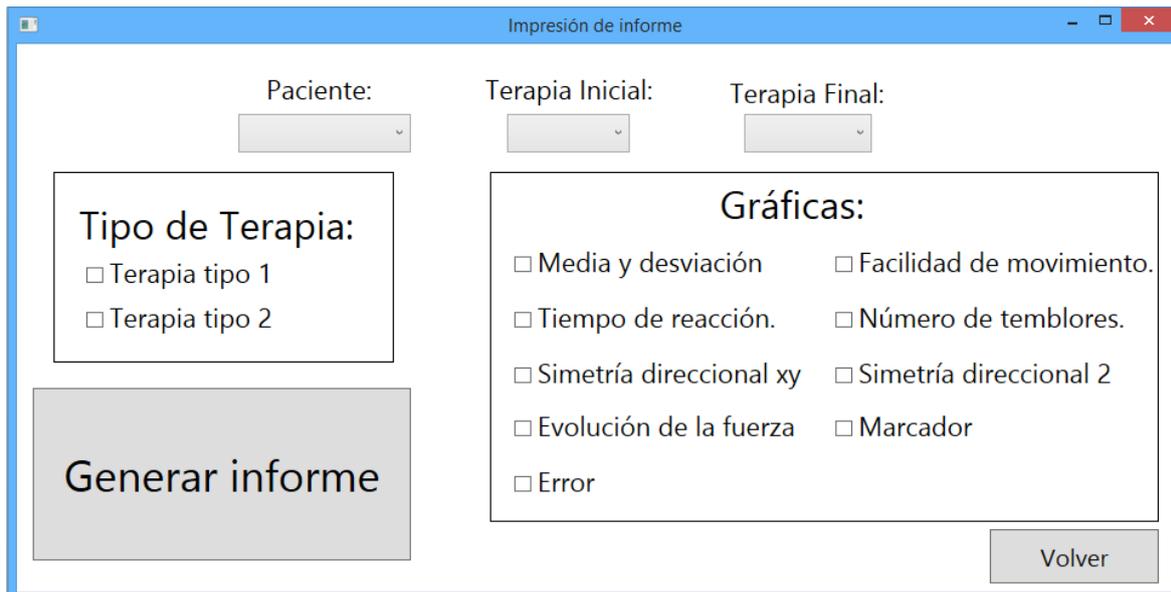


Fig. 51. Visualización de la comparación de la evolución del paciente.

Podemos elegir tanto terapia tipo 1 como tipo 2 o ambas y cualquiera de las gráficas de evolución que queramos elegimos las gráfica, el paciente y la terapia inicial y final y nos generará el pdf que queremos, que esta adjunto en los anexos.

- **Actualizar**

Pulsando el botón de actualizar no se crea una ventana como en los casos anteriores, sino que se ejecuta el archivo de actualización de R para que abra Resumen.txt y actualice el documento o lo cree en el caso de que el documento no exista.

En el caso de que se dé por error y el documento Resumen.txt ya está actualizado, no pasaría nada porque como dijimos cuando describimos el script de R tenía una





3. CONCLUSIONES





Debido a que el robot háptico para la rehabilitación Physiobot genera una inmensa cantidad de datos que almacenamos en una base de datos, necesitamos una forma para poder procesarlos, y hemos encontrado que lenguaje de programación R es un lenguaje muy completo y muy recomendable a la hora del análisis estadístico de grandes cantidades de datos.

Además para el análisis de estos datos lo más útil es emplear métricas y ha sido muy importante el estudio de las métricas previas de robots similares y de artículos de diferentes autores para poder hacer gráficas por con mayor fiabilidad y aceptación. El lenguaje de programación R es muy útil para graficar estas métricas.

Como el robot Physiobot tiene aspectos diferentes a muchos otros robots de rehabilitación ha sido necesaria la creación de nuevas métricas, tanto para analizar las sesiones de forma individual, como para ver la evolución de los pacientes a lo largo del tiempo analizando diversos parámetros.

Se ha realizado una interfaz gráfica para que cuando un médico o un fisioterapeuta consulte las métricas con los datos generados por un paciente al utilizar el robot Physiobot, pueda hacerlo sin la necesidad de ejecutar un script.

Esta interfaz gráfica se ha hecho lo más transparente posible, mediante la reducción del tiempo de ejecución de los scripts de R, como con la creación de un fichero de actualización para que sea posible calcular ciertos datos previamente a ejecutar el cálculo de las métricas. Esto es clave para dar una sensación de mayor continuidad mientras se emplea el software creado.

Además el incluir la posibilidad de imprimir un informe en PDF en el informe ayuda a tener un software más completo y permite la capacidad de poder tener los resultados guardados y no volver a calcularlos.

Ha sido importante la creación de filtros para adaptarnos a las necesidades específicas del robot a la hora de obtener los datos de la base de datos con R y esto nos ha llevado a tener que descartar algunas sesiones por diversos motivos.

Para finalizar decir que emplear C# con WPF para realizar la interfaz ha sido un acierto, dado que nos ha permitido comunicarnos con facilidad tanto con R como con la base de datos.





4. FUTURAS LÍNEAS DE TRABAJO





Debido a que tratamos con una primera versión del software necesita ser probado por varios usuarios para saber que partes se pueden mejorar del mismo.

Las nuevas métricas creadas necesitan ser valoradas por más especialistas médicos para saber si pueden ser mejoradas o para la creación de nuevas métricas a mayores valorando otros puntos de interés clínico.

También los usuarios del software deberán analizar más en profundidad la interfaz gráfica para ver si existe la necesidad de la creación de nuevas ventanas o de añadir nuevas funciones en el programa así como si es necesaria la modificación de algunas de las funciones ya existentes.

Como mejoras para una segunda versión del software existe la posibilidad de en el tiempo en el que estén funcionando los scripts de R en la interfaz gráfica aparezca una barra de proceso o que de alguna forma el usuario sea consciente de que el programa esté funcionando para que no intente ejecutar las mismas acciones pensando que el software no lo ha recibido.

También se puede implementar que en lugar de incrustar imágenes con las métricas correspondientes, en las métricas de evolución se puede implementar un javascript de forma que se pueda interactuar con las métricas y por ejemplo poner encima de cada punto generado en estas gráficas el cursor y que nos dé el valor exacto, número de terapias y varios parámetros más.

En la generación del pdf se puede reducir el número de decimales que se muestra cuando calculamos los valores.





5. BIBLIOGRAFÍA





- [1] C. Rodriguez Guerrero, EFTO Hospital Beata Maria Ana, P. Oliva Navarrete, J.C Fraile and P. Rivera Farina "Preliminary Results From the Use of the SOFTROBOT Platform in Stroke Patients"
- [2] T. Platz, "Evidence-based arm rehabilitation a systematic review of the literature," *Der Nervenarzt*, vol. 74, pp. 841-9, Oct. 2003.
- [3] D. J. Reinkensmeyer, L. Kahn, M. Averbuch, A. McKena-Cole, B. D. Schmit, and W. Z. Rymer, "Understanding and treating arm movement impairment after chronic brain injury: Progress with the ARM guide," *Rehab. Res. Dev.*, vol. 37, no. 6, pp. 653-662, 2000.
- [4] C. D. Rodriguez Guerrero, "Bio Cooperative Robotic Platform for Motor Function Recovery of The Upper Limb After Stroke," in *IEEE EMBS 2010*, no. Figure 1, pp. 4472-4475, 2010.
- [5] P. Lum, D. Reinkensmeyer, R. Mahoney, W. Z. Rymer, and C. Burgar, "Robotic devices for movement therapy after stroke: current status and challenges to clinical acceptance," *Hand, The*, vol. 8, no. 4, pp. 40-53, 2002.
- [6] H. Krebs, N. Hogan, M. Aisen, and B. Volpe, "Robot-aided neurorehabilitation," *IEEE Transactions on Rehabilitation Engineering*, vol. 6, pp. 75-87, Mar. 1998.
- [7] G. Kwakkel, R. van Peppen, R. Wagenaar, S. Wood, and C. Richards, "Effects of augmented exercise therapy time after stroke: a meta-analysis," *Stroke*, vol. 35, pp. 25-39, 2004.
- [8] J. Hidler, D. Nichols, M. Pelliccio, and K. Brady, "Advances in the understanding and treatment of stroke impairment using robotic devices.," *Topics in stroke rehabilitation*, vol. 12, pp. 22-35, Jan. 2005.
- [9] N. Hogan and H. I. Krebs, "Interactive robots for neurorehabilitation.," *Restorative neurology and neuroscience*, vol. 22, pp. 349-58, Jan. 2004.
- [10] G. B. Prange, M. J. A. Jannink, C. G. M. Grootuis-Oudshoorn, H. J. Hermens, and M. J. IJzerman, "Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke," *The Journal of Rehabilitation Research and Development*, vol. 43, no. 2, p. 171, 2006.
- [11] J. Mehrholz, T. Platz, J. Kugler, and M. Pohl, "Electromechanical and robot-assisted arm training for improving arm function and activities of daily living after stroke.," *Cochrane database of systematic reviews (Online)*, p. CD006876, Jan. 2008.



[12]Descripción: Noticia de Cartif acerca del Physiobot,

URL: <http://www.cartif.com/comunicacion/historico/2012/item/197-cartif-presenta-su-nuevo-robot-de-neurorehabilitaci%C3%B3n-en-el-icnr-2012.html>

Última fecha de consulta: 01/09/2012

[13] W. N. Venables, D. M. Smith and the R Core Team: An Introduction to R Notes on R: A Programming Environment for Data Analysis and Graphics Version 3.2.2 (2015-08-14),

[14] Andrés González, Silvia González y Development Core Team: Introducción a R, Notas sobre R: Un entorno de programación para Análisis de Datos y Gráficos Versión 1.0.1 (2000-05-16)

[15]Descripción: Explicación del funcionamiento del paquete RODBC

URL: <http://cran.r-project.org/web/packages/RODBC/RODBC.pdf>

Última fecha de consulta: 01/09/2012

[16] Rohrer et al: Movement Smoothness Changes during Stroke Recovery. J. Neurosci., September 15, 2002, 22(18):8297–8304

[17]Descripción: Explicación del funcionamiento del paquete FKF

URL: <http://cran.r-project.org/web/packages/FKF/FKF.pdf>

Última fecha de consulta: 01/09/2012

[18] Ingeniería de control moderna, 5ª Edición. Katsuhiko Ogata , PRENTICE-HALL, 2010

[19] Nordin et al.: Assessment of movement quality in robot- assisted upper limb rehabilitation after stroke: a review. Journal of NeuroEngineering and Rehabilitation 2014 11:137.

[20] Support Vector Machine With Adaptive Parameters in Financial Time Series Forecasting L. J. Cao y Francis E. H. Tay IEEE Transactions on Neural Networks, vol. 14, n. 6 pp. 1506-1518. Noviembre 2003

[17]Descripción: Explicación del funcionamiento del paquete PNG

URL: <https://cran.r-project.org/web/packages/png/png.pdf>

Última fecha de consulta: 01/09/2012



6. ANEXOS





ANEXO 1: INFORME PDF



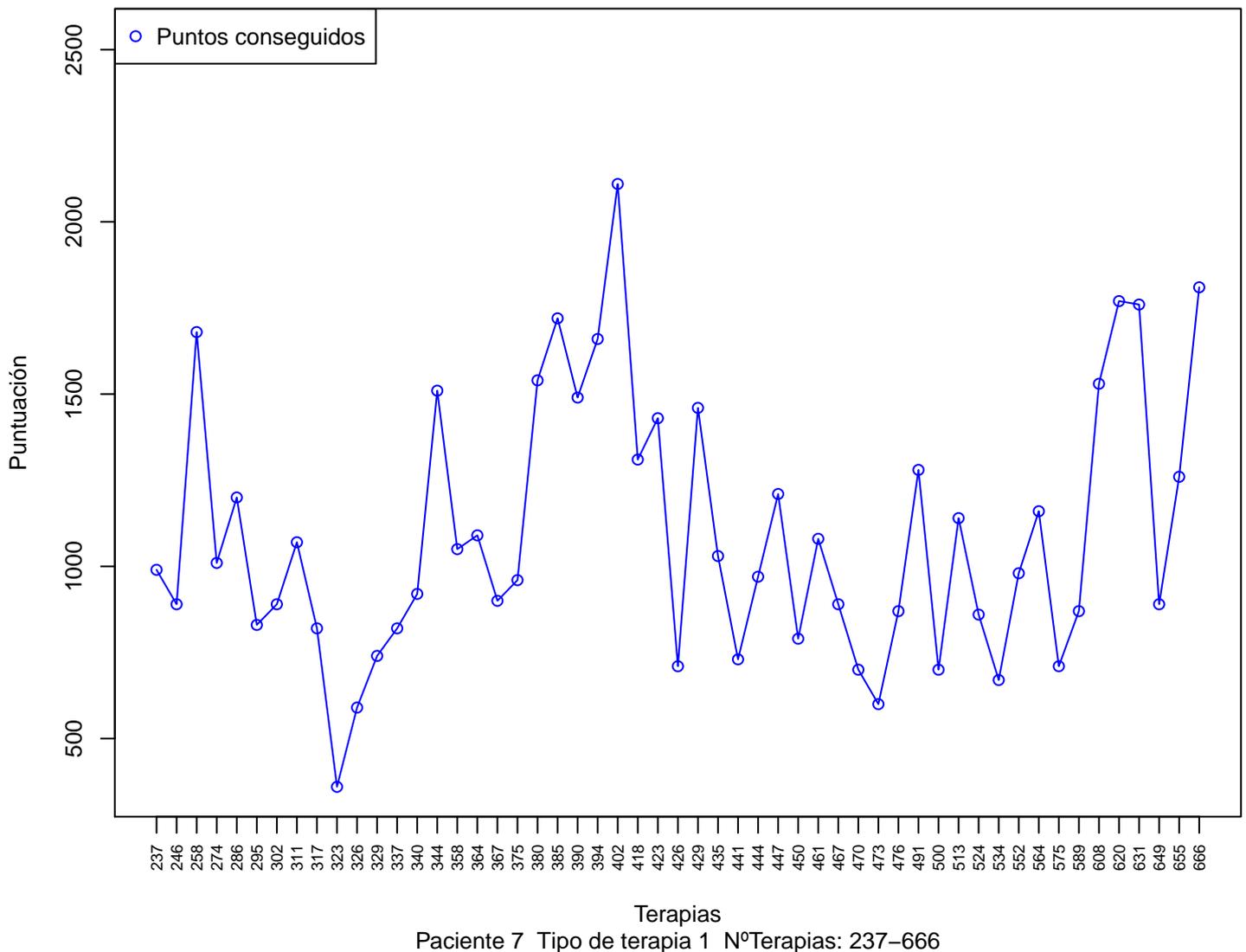
Paciente 7 Marcador Terapia 1

Fecha de impresión: 2015-09-01 23:30:31

En esta métrica vamos a ver la cantidad de puntos que consigue el paciente en el juego del ratón, cada vez que el paciente atrapa al ratón recibe puntos de forma que cuantas más veces haya atrapado al ratón, mayor puntuación tendrá, la puntuación puede variar bastante dependiendo de la dificultad. El índice de esta métrica son los puntos del juego.

Máxima puntuación Fecha: 2014-05-21 14:14:03 Terapia número: 402 Puntuación: 2110
 Mínima puntuación Fecha: 2014-05-07 17:12:36 Terapia número: 323 Puntuación: 360
 Primera puntuación Fecha: 2014-04-28 10:49:33 Terapia número: 237 Puntuación: 990
 Última puntuación Fecha: 2014-06-02 16:16:47 Terapia número: 666 Puntuación: 1810
 Puntuación Media: 1095

Marcador





Paciente 7 Porcentaje de error Terapia 1

Fecha de impresión: 2015-09-01 23:30:31

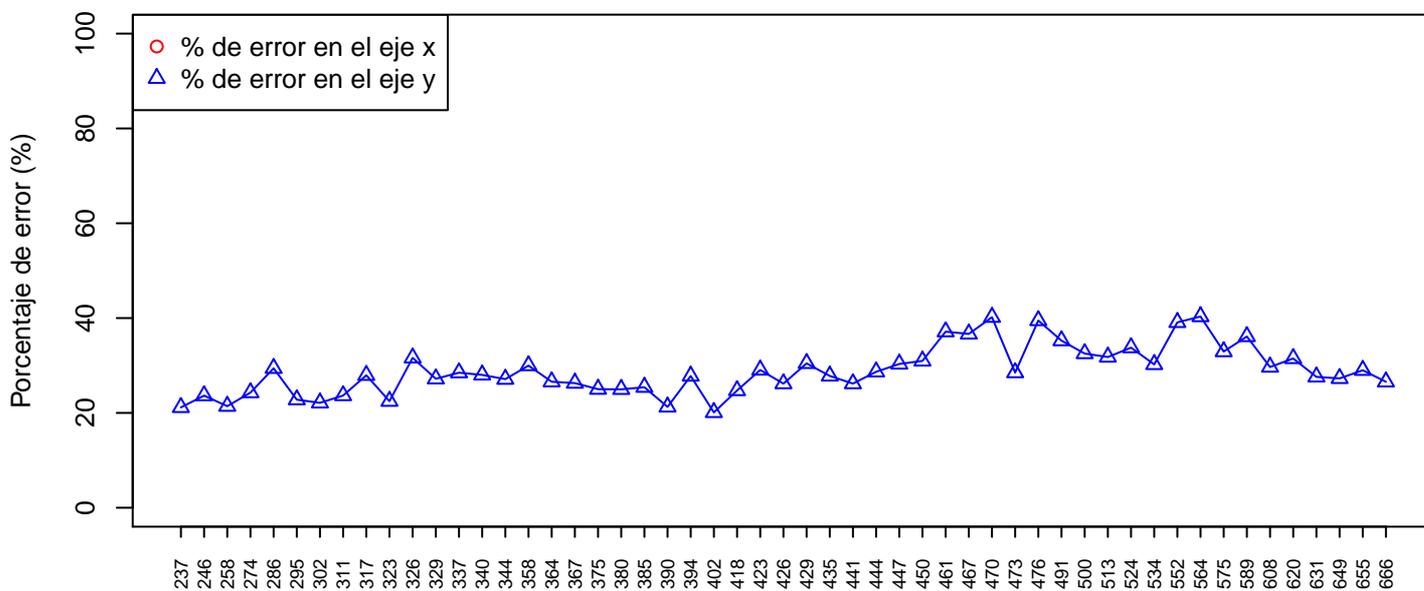
Vamos a visualizar la diferencia que hay en un punto entre x e y realizados por el paciente, con respecto a los puntos deseados x e y, para poder ver que movimiento realiza mejor, para ello se hace la media del punto que ha movido el paciente menos el punto deseado, entre el punto deseado tanto en x como en y.

El índice es el error relativo en "x" y en "y", y se medirá en %.

Máximo error en x Fecha: 2014-05-21 13:42:47 Terapia número: 364 Porcentaje de error: 8.03246 %
Mínimo error en x Fecha: 2014-06-02 12:32:10 Terapia número: 447 Porcentaje de error: 3.134385 %
Primer error en x Fecha: 2014-04-28 10:49:33 Terapia número: 237 Porcentaje de error: 3.38524 %
Último error en x Fecha: 2014-06-02 16:16:47 Terapia número: 666 Porcentaje de error: 4.117816 %
Error Medio en x : 5 %

Máximo error en y Fecha: 2014-06-02 15:59:45 Terapia número: 564 Porcentaje de error: 40.32364 %
Mínimo error en y Fecha: 2014-05-21 14:14:03 Terapia número: 402 Porcentaje de error: 20.10522 %
Primer error en y Fecha: 2014-04-28 10:49:33 Terapia número: 237 Porcentaje de error: 21.1574 %
Último error en y Fecha: 2014-06-02 16:16:47 Terapia número: 666 Porcentaje de error: 26.56489 %
Error Medio en y : 29 %

Error cometido



Paciente 7 Media y desviación Terapia 1

Fecha de impresión: 2015-09-01 23:30:31

Media: Es la suma de la diferencia de los puntos geométricos en el plano en x y en y recorridos por el paciente con respecto a los puntos deseados, dividido entre el numero de puntos recorridos.

Desviación típica: Es el sumatorio de la raíz cuadrada del cuadrado de la diferencia de los puntos geométrico en el plano en x y en y recorridos por el paciente con respecto a los puntos deseados, dividido entre el numero de puntos recorridos.

Los índices que tendremos serán la media y la desviación típica de los puntos de cada terapia.

Máxima media Fecha: 2014-06-02 13:57:39 Terapia número: 470 Media: 191.7076

Mínima media Fecha: 2014-04-28 12:25:10 Terapia número: 258 Media: 122.0406

Primera media Fecha: 2014-04-28 10:49:33 Terapia número: 237 Media: 131.692

Última media Fecha: 2014-06-02 16:16:47 Terapia número: 666 Media: 165.8592

Media de la media : 159

Máxima desviación Fecha: 2014-06-02 13:58:02 Terapia número: 473 Desviación típica: 75.18665

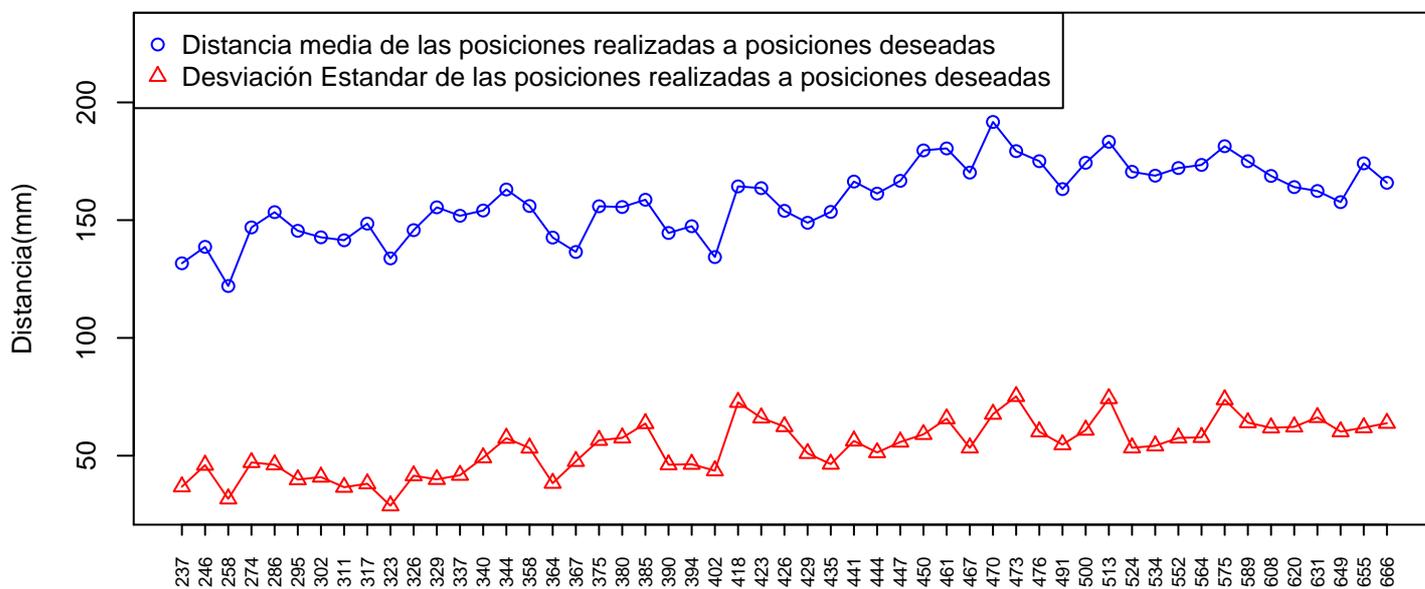
Mínima desviación Fecha: 2014-05-07 17:12:36 Terapia número: 323 Desviación típica: 28.75455

Primera desviación Fecha: 2014-04-28 10:49:33 Terapia número: 237 Desviación típica: 36.81957

Última desviación Fecha: 2014-06-02 16:16:47 Terapia número: 666 Desviación típica: 63.79402

Media de la desviación: 54

Media y desviación



Paciente 7 Facilidad de movimiento Terapia 1

Fecha de impresión: 2015-09-01 23:30:31

Según varios estudios, dicen que la facilidad de movimiento se puede relacionar con a velocidad, de forma que cuanto más rápido tengas las capacidad de moverte, menos esfuerzo te costará realizar la tarea y por lo tanto tendrás una mayor facilidad de movimiento.

El índice será la velocidad media medida en m/s.

Máxima velocidad Fecha: 2014-06-02 13:57:39 Terapia número: 470 Velocidad: 0.1446469 m/s
 Mínima velocidad Fecha: 2014-04-28 10:49:33 Terapia número: 237 Velocidad: 0.01515689 m/s
 Primera velocidad Fecha: 2014-04-28 10:49:33 Terapia número: 237 Velocidad: 0.01515689 m/s
 Última velocidad Fecha: 2014-06-02 16:16:47 Terapia número: 666 Velocidad: 0.08669902 m/s
 Velocidad media: 0.0879399301886792 m/s

Facilidad de movimiento





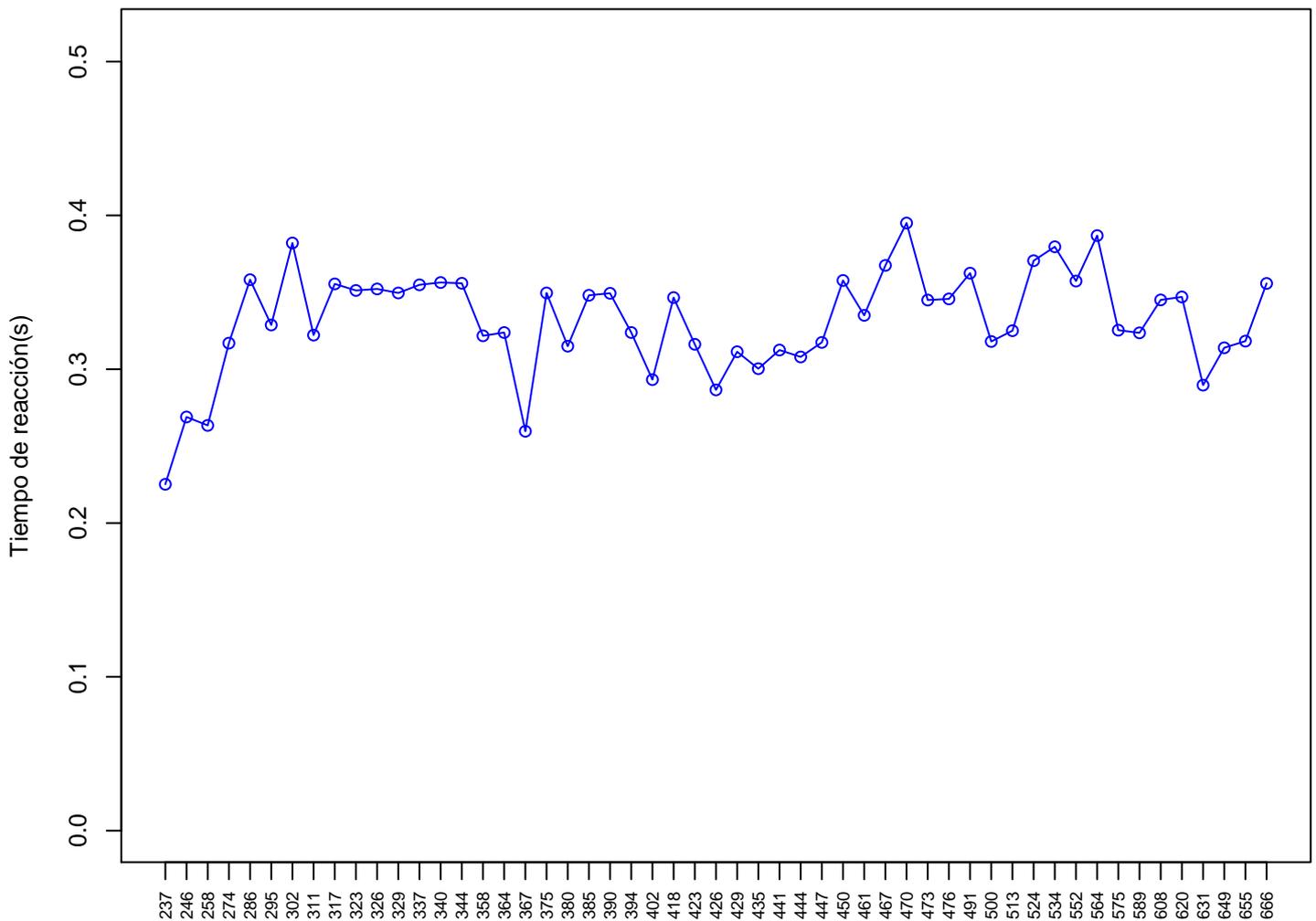
Paciente 7 Tiempo de Reacción Terapia 1

Fecha de impresión: 2015-09-01 23:30:31

Llamamos tiempo de reacción al tiempo que pasa desde que en la trayectoria deseada se produce un cambio de sentido hasta que en la trayectoria del robot se produce ese cambio de sentido también. Para ello restamos el tiempo en el que se produjo el cambio en el robot menos el tiempo en el que se produjo el cambio en el sentido deseado. El índice que tendremos será el tiempo de reacción y se medirá en segundos.

Máximo tiempo de reacción Fecha: 2014-06-02 13:57:39 Terapia número: 470 Tiempo: 0.3950496 s
Mínimo tiempo de reacción Fecha: 2014-04-28 10:49:33 Terapia número: 237 Tiempo: 0.2252034 s
Primer tiempo de reacción Fecha: 2014-04-28 10:49:33 Terapia número: 237 Tiempo: 0.2252034 s
Último tiempo de reacción Fecha: 2014-06-02 16:16:47 Terapia número: 666 Tiempo: 0.35578 s
Tiempo de reacción medio: 0.331878790566038 s

Tiempo de Reacción



Terapia

Paciente 7 Tipo de terapia 1 N°Terapias: 237-666

Paciente 7 Número de temblores Terapia 1

Fecha de impresión: 2015-09-01 23:30:31

Vemos cuantas veces ha estado parado el robot cuando era manejado por el fisioterapeuta o cuantas veces el ratón en la pantalla ha estado en una posición fija, y cuantas veces a estado parado el paciente, así como la comparación de la mismas.

El índice de es el numero de veces que el robot esta parado.

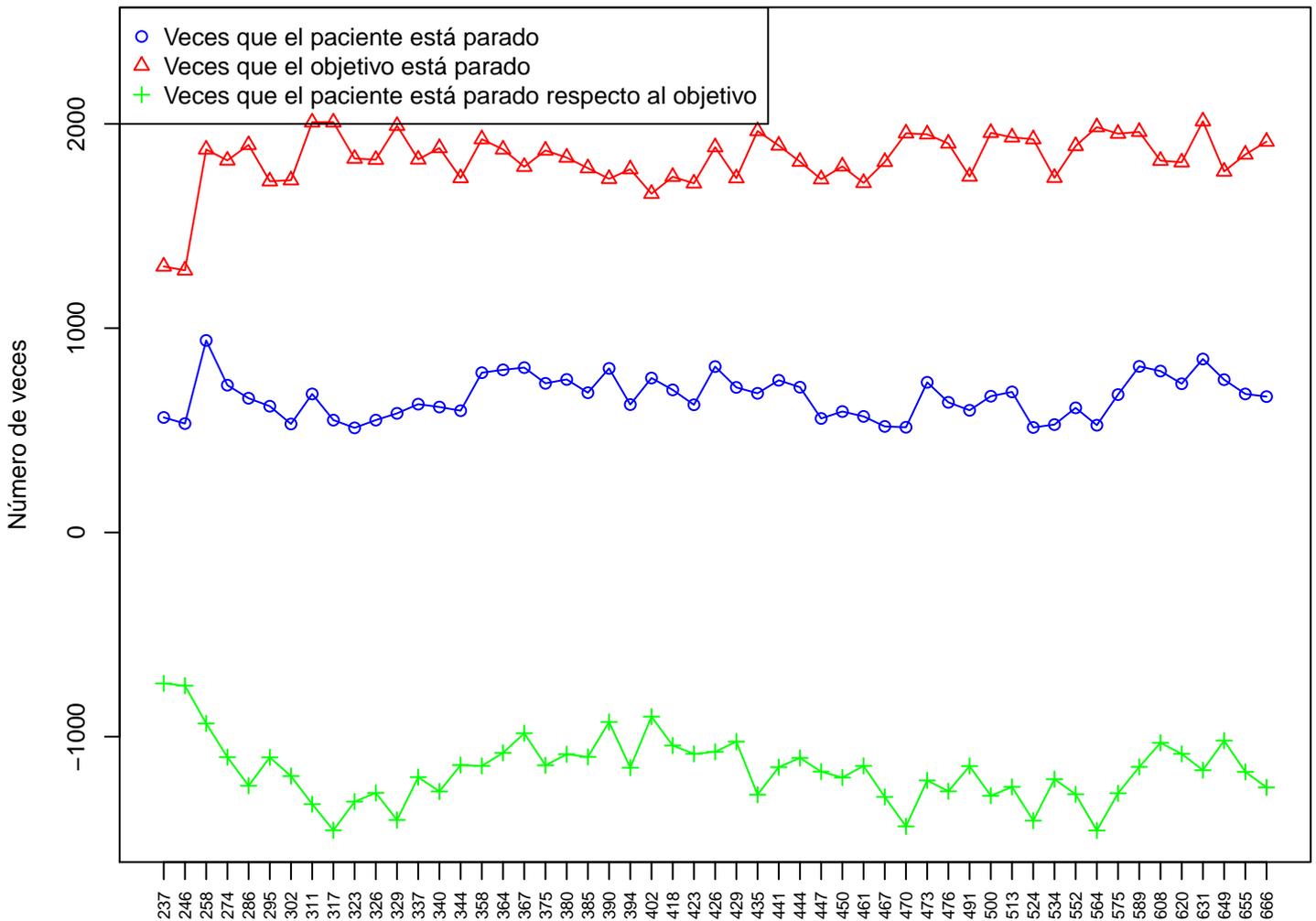
Máxima realción Robot/Paciente Fecha: 2014-04-28 10:49:33 n° Terapia : 237 Diferencia de veces: -739

Mínima realción Robot/Paciente Fecha: 2014-06-02 15:59:45 n° Terapia : 564 Diferencia de veces: -1459

Primera realción Robot/Paciente Fecha: 2014-04-28 10:49:33 n° Terapia : 237 Diferencia de veces: -739

Última realción Robot/Paciente Fecha: 2014-06-02 16:16:47 n° Terapia: 666 Diferencia de veces: -1248
 Media de veces que el paciente esta parado respecto al robot: -1162

Número de temblores



Paciente 7 Simetría direccional x e y Terapia 1

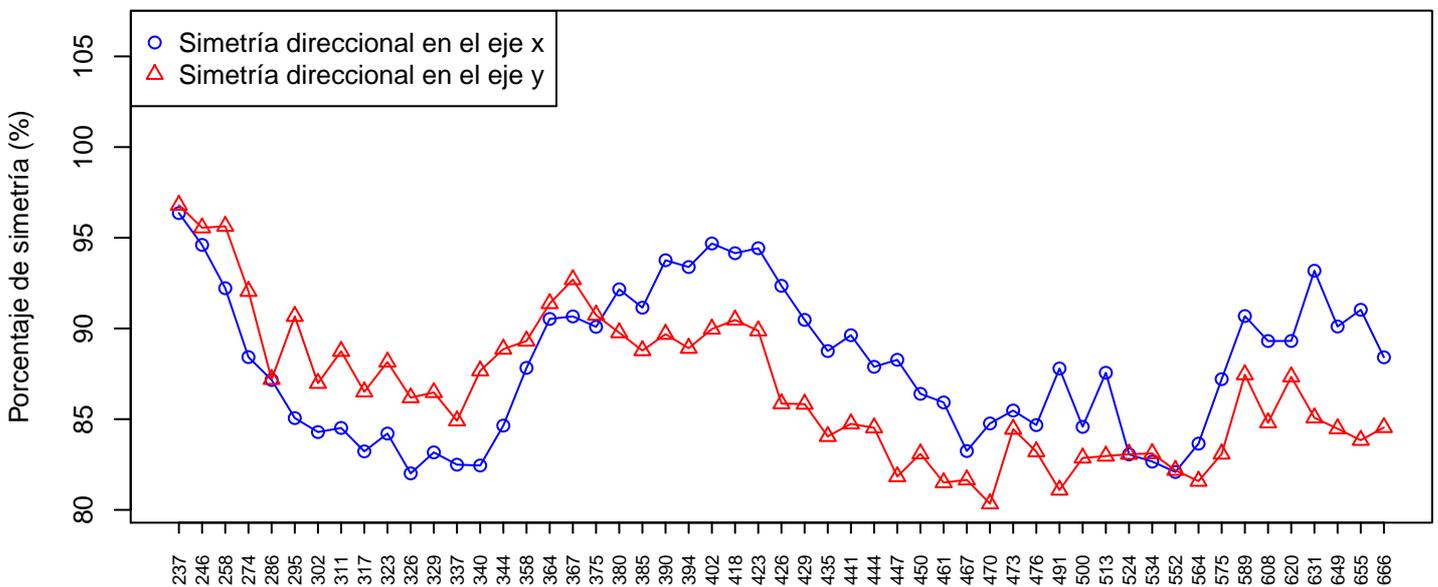
Fecha de impresión: 2015-09-01 23:30:31

En el análisis estadístico de una serie temporal y en el procesamiento de la señal, la simetría direccional es una medida estadística de rendimiento de un modelo para predecir el cambio, positivo o negativo, de una serie temporal de un período de tiempo al siguiente. El índice de la simetría direccionales es el porcentaje de simetría direccional(%).

Máxima Simetría direccional x Fecha: 2014-04-28 10:49:33 Terapia número: 237 Simetría direccional: 97 %
 Mínima Simetría direccional x Fecha: 2014-05-07 17:14:34 Terapia número: 326 Simetría direccional: 83 %
 Primera Simetría direccional x Fecha: 2014-04-28 10:49:33 Terapia número: 237 Simetría direccional: 97 %
 Última Simetría direccional x Fecha: 2014-06-02 16:16:47 Terapia número: 666 Simetría direccional: 89 %
 Media de la Simetría direccional x : 89 %

Máxima Simetría direccional y Fecha: 2014-04-28 10:49:33 Terapia número: 237 Simetría direccional: 97 %
 Mínima Simetría direccional y Fecha: 2014-06-02 13:57:39 Terapia número: 470 Simetría direccional: 81 %
 Primera Simetría direccional y Fecha: 2014-04-28 10:49:33 Terapia número: 237 Simetría direccional: 97 %
 Última Simetría direccional y Fecha: 2014-06-02 16:16:47 Terapia número: 666 Simetría direccional: 85 %
 Media de la Simetría direccional y : 87 %

Simetría direccional en el eje x y en el eje y



Paciente 7 Simetría velocidad y ángulo Terapia 1

Fecha de impresión: 2015-09-01 23:30:31

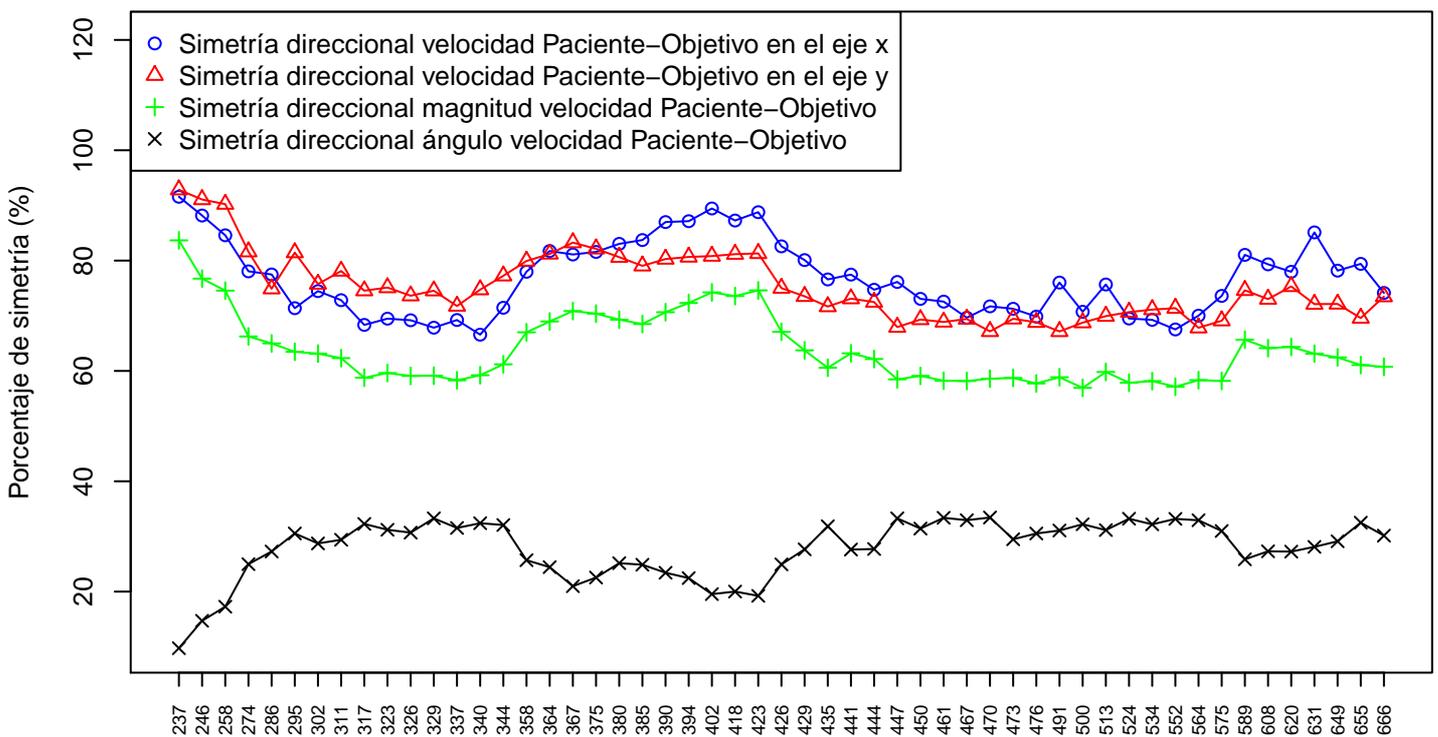
Simetría direccional de la velocidad en x y en y, la magnitud de la velocidad es la semisuma al cuadrado de las velocidades x e y y el ángulo es el arcoseno de la velocidad con respecto a la magnitud lo que vemos en la pantalla es el porcentaje de que ángulo sea el mismo del paciente y de la trayectoria objetivo.

Los índices son porcentajes de simetría(%).

Máxima simetría angular Fecha: 2014-06-02 13:57:39 nº Terapia: 470 Porcentaje de simetría: 34 %
 Mínima simetría angular Fecha: 2014-04-28 10:49:33 nº Terapia: 237 Porcentaje de simetría: 10 %
 Primera simetría angular Fecha: 2014-04-28 10:49:33 nº Terapia: 237 Porcentaje de simetría: 10 %
 Última simetría angular Fecha: 2014-06-02 16:16:47 nº Terapia: 666 Porcentaje de simetría: 31 %
 Media de la simetría angular : 28 %

Máxima magnitud de velocidad Fecha: 2014-04-28 10:49:33 nº Terapia: 237 Porcentaje de simetría: 84 %
 Mínima magnitud de velocidad Fecha: 2014-06-02 15:54:58 nº Terapia: 500 Porcentaje de simetría: 57 %
 Primera magnitud de velocidad Fecha: 2014-04-28 10:49:33 nº Terapia: 237 Porcentaje de simetría: 84 %
 Última magnitud de velocidad Fecha: 2014-06-02 16:16:47 nº Terapia: 666 Porcentaje de simetría: 61 %
 Media de la simetría de la magnitud de velocidad: 64 %

Simetría direccional



Paciente 7 Evolucion de la fuerza Terapia 1

Fecha de impresión: 2015-09-01 23:30:32

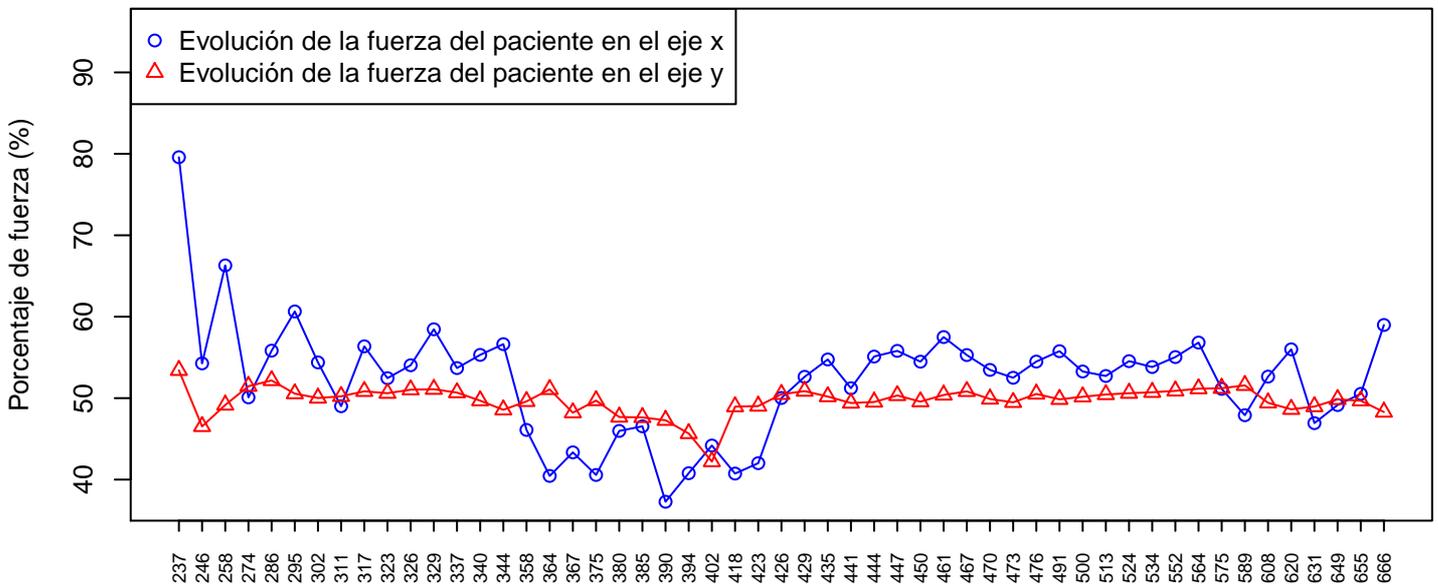
Esta medida va a marcar la evolución de la fuerza que tenga el paciente, para ello lo que valoramos es no solo la fuerza que aplica el paciente sino también la fuerza con la que le ayuda el robot. Para calcular la evolución vamos a tomar la fuerza que aplica el paciente y la vamos a dividir entre la suma de la fuerza que aplica el robot y la fuerza que aplica el paciente.

El índice será un porcentaje de la fuerza en x y en y que realiza el paciente con respecto a la fuerza que hace el paciente y la fuerza asistida que recibe, se medirá con un porcentaje (%).

Máxima evolución en x Fecha: 2014-04-28 10:49:33 Terapia número: 237 Evolución: 79.5817 %
 Mínima evolución en x Fecha: 2014-05-21 13:57:33 Terapia número: 390 Evolución: 37.27911 %
 Primera evolución en x Fecha: 2014-04-28 10:49:33 Terapia número: 237 Evolución: 79.5817 %
 Última evolución en x Fecha: 2014-06-02 16:16:47 Terapia número: 666 Evolución: 58.98654 %
Media de la evolución en x : 53 %

Máxima evolución en y Fecha: 2014-04-28 10:49:33 Terapia número: 237 Evolución: 53.41449 %
 Mínima evolución en y Fecha: 2014-05-21 14:14:03 Terapia número: 402 Evolución: 42.20542 %
 Primera evolución en y Fecha: 2014-04-28 10:49:33 Terapia número: 237 Evolución: 53.41449 %
 Última evolución en y Fecha: 2014-06-02 16:16:47 Terapia número: 666 Evolución: 48.29314 %
Media de la evolución en y : 50 %

Evolución de la fuerza





Paciente 7 Porcentaje de error Terapia 2

Fecha de impresión: 2015-09-01 23:30:32

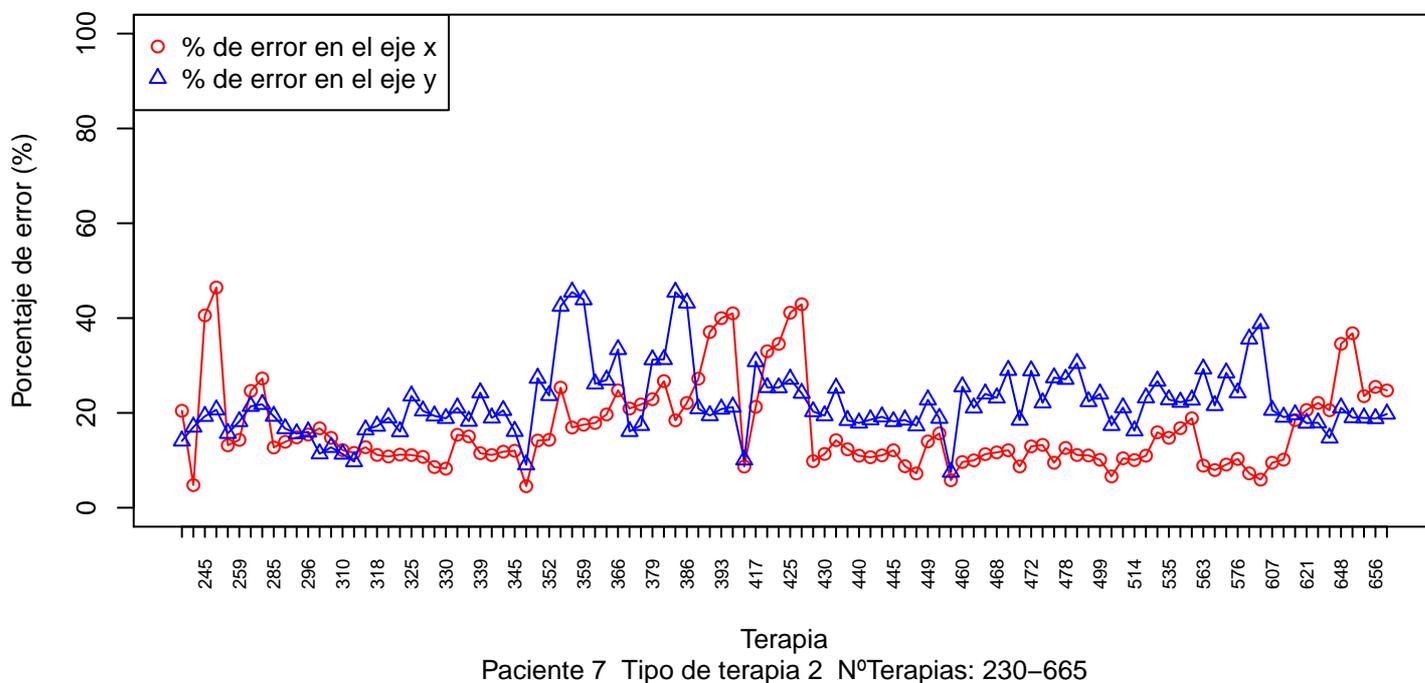
Vamos a visualizar la diferencia que hay en un punto entre x e y realizados por el paciente, con respecto a los puntos deseados x e y, para poder ver que movimiento realiza mejor, para ello se hace la media del punto que ha movido el paciente menos el punto deseado, entre el punto deseado tanto en x como en y.

El índice es el error relativo en "x" y en "y", y se medirá en %.

Máximo error en x Fecha: 2014-04-28 12:29:54 Terapia número: 247 Porcentaje de error: 46.45232 %
Mínimo error en x Fecha: 2014-06-02 12:20:34 Terapia número: 346 Porcentaje de error: 4.545975 %
Primer error en x Fecha: 2014-04-28 10:49:33 Terapia número: 230 Porcentaje de error: 20.45376 %
Último error en x Fecha: 2014-06-03 16:58:08 Terapia número: 665 Porcentaje de error: 24.74628 %
Error Medio en x : 17 %

Máximo error en y Fecha: 2014-06-02 13:57:04 Terapia número: 357 Porcentaje de error: 45.49654 %
Mínimo error en y Fecha: 2014-06-02 17:08:17 Terapia número: 452 Porcentaje de error: 7.502062 %
Primer error en y Fecha: 2014-04-28 10:49:33 Terapia número: 230 Porcentaje de error: 14.08571 %
Último error en y Fecha: 2014-06-03 16:58:08 Terapia número: 665 Porcentaje de error: 19.76944 %
Error Medio en y : 23 %

Error cometido



Paciente 7 Media y desviación Terapia 2

Fecha de impresión: 2015-09-01 23:30:32

Media: Es la suma de la diferencia de los puntos geométricos en el plano en x y en y recorridos por el paciente con respecto a los puntos deseados, dividido entre el numero de puntos recorridos.

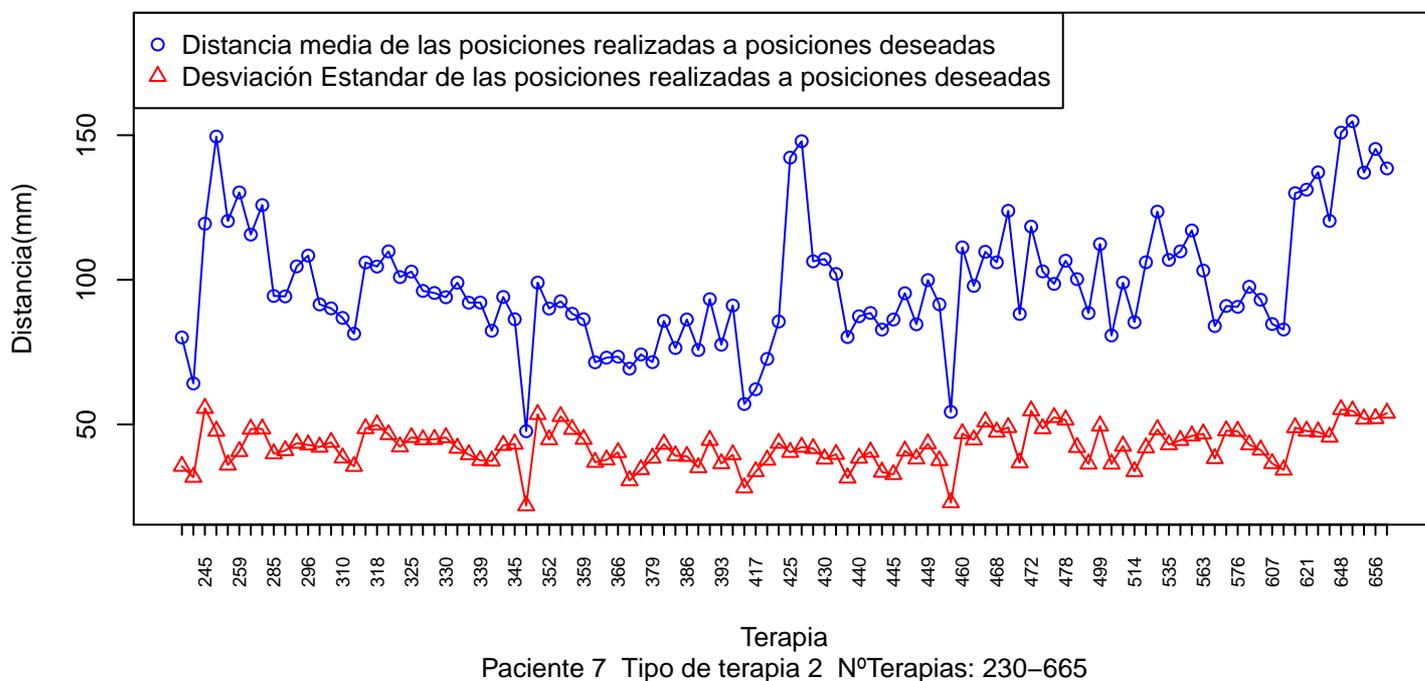
Desviación típica: Es el sumatorio de la raíz cuadrada del cuadrado de la diferencia de los puntos geométrico en el plano en x y en y recorridos por el paciente con respecto a los puntos deseados, dividido entre el numero de puntos recorridos.

Los índices que tendremos serán la media y la desviación típica de los puntos de cada terapia.

Máxima media Fecha: 2014-06-03 16:57:48 Terapia número: 650 Media: 154.8496
 Mínima media Fecha: 2014-06-02 12:20:34 Terapia número: 346 Media: 47.67385
 Primera media Fecha: 2014-04-28 10:49:33 Terapia número: 230 Media: 80.08818
 Última media Fecha: 2014-06-03 16:58:08 Terapia número: 665 Media: 138.5327
Media de la media : 99

Máxima desviación Fecha: 2014-04-28 12:25:10 Terapia número: 245 Desviación típica: 55.59799
 Mínima desviación Fecha: 2014-06-02 12:20:34 Terapia número: 346 Desviación típica: 21.90164
 Primera desviación Fecha: 2014-04-28 10:49:33 Terapia número: 230 Desviación típica: 35.56957
 Última desviación Fecha: 2014-06-03 16:58:08 Terapia número: 665 Desviación típica: 53.93688
Media de la desviación: 43

Media y desviación



Paciente 7 Facilidad de movimiento Terapia 2

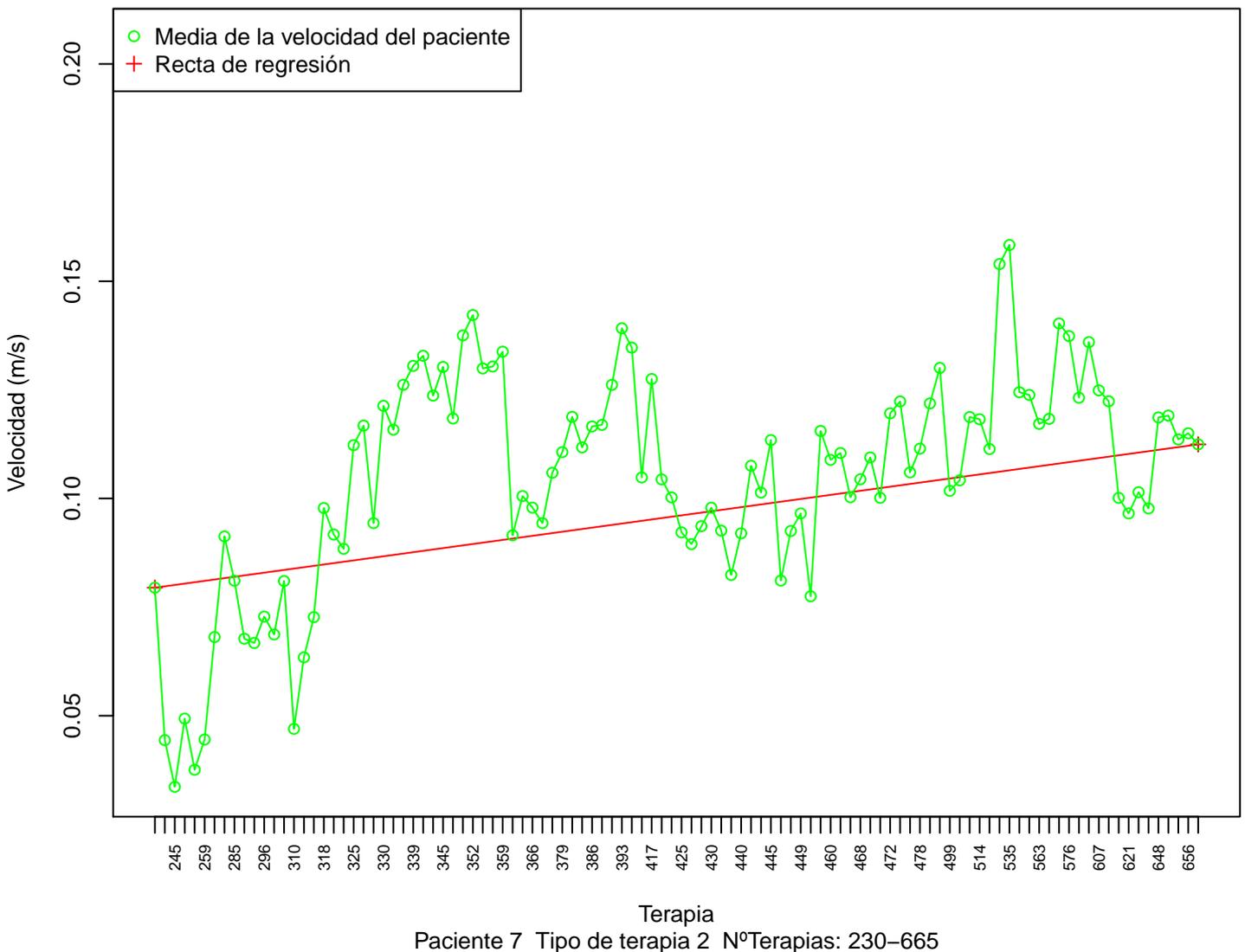
Fecha de impresión: 2015-09-01 23:30:32

Según varios estudios, dicen que la facilidad de movimiento se puede relacionar con a velocidad, de forma que cuanto más rápido tengas las capacidad de moverte, menos esfuerzo te costará realizar la tarea y por lo tanto tendrás una mayor facilidad de movimiento.

El índice será la velocidad media medida en m/s.

Máxima velocidad Fecha: 2014-06-03 16:30:28 Terapia número: 535 Velocidad: 0.1583421 m/s
 Mínima velocidad Fecha: 2014-04-28 12:25:10 Terapia número: 245 Velocidad: 0.03364103 m/s
 Primera velocidad Fecha: 2014-04-28 10:49:33 Terapia número: 230 Velocidad: 0.07946977 m/s
 Última velocidad Fecha: 2014-06-03 16:58:08 Terapia número: 665 Velocidad: 0.1124527 m/s
 Velocidad media: 0.104961510377358 m/s

Facilidad de movimiento





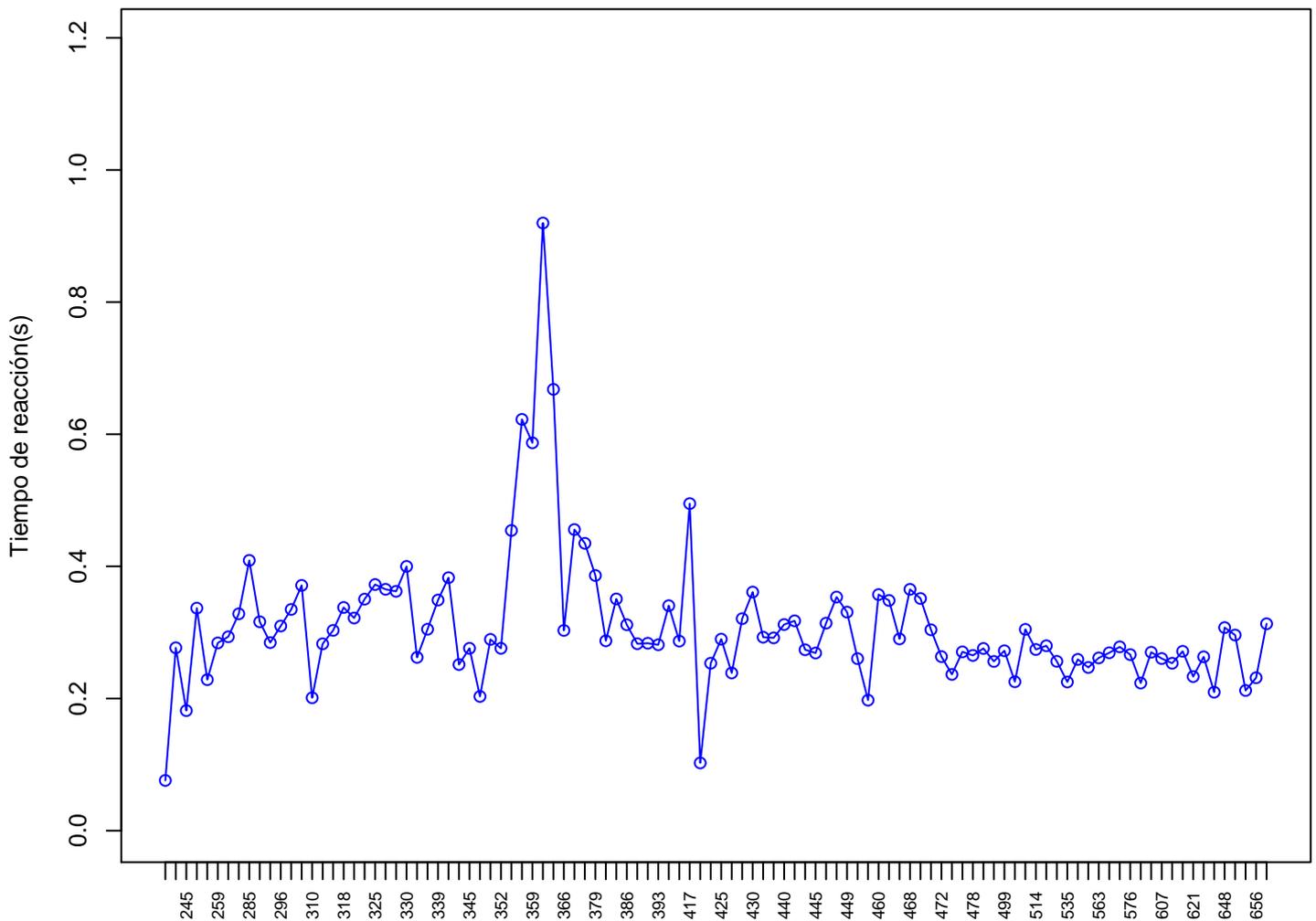
Paciente 7 Tiempo de Reacción Terapia 2

Fecha de impresión: 2015-09-01 23:30:32

Llamamos tiempo de reacción al tiempo que pasa desde que en la trayectoria deseada se produce un cambio de sentido hasta que en la trayectoria del robot se produce ese cambio de sentido también. Para ello restamos el tiempo en el que se produjo el cambio en el robot menos el tiempo en el que se produjo el cambio en el sentido deseado. El índice que tendremos será el tiempo de reacción y se medirá en segundos.

Máximo tiempo de reacción Fecha: 2014-06-02 13:58:02 Terapia número: 363 Tiempo: 0.9196526 s
Mínimo tiempo de reacción Fecha: 2014-04-28 10:49:33 Terapia número: 230 Tiempo: 0.0760044 s
Primer tiempo de reacción Fecha: 2014-04-28 10:49:33 Terapia número: 230 Tiempo: 0.0760044 s
Último tiempo de reacción Fecha: 2014-06-03 16:58:08 Terapia número: 665 Tiempo: 0.3131278 s
Tiempo de reacción medio: 0.309475483018868 s

Tiempo de Reacción



Paciente 7 Tipo de terapia 2 N°Terapias: 230-665

Paciente 7 Número de temblores Terapia 2

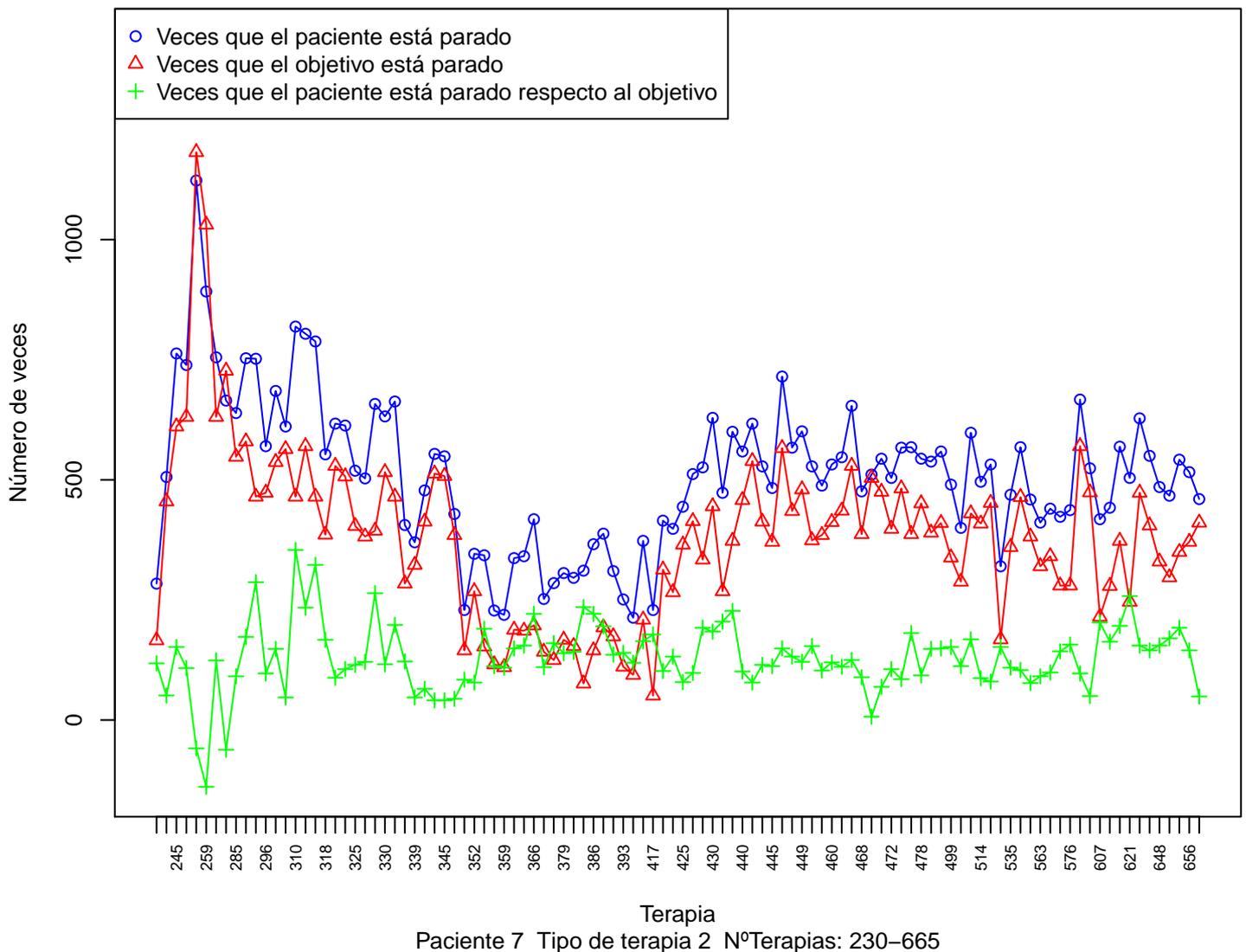
Fecha de impresión: 2015-09-01 23:30:32

Vemos cuantas veces ha estado parado el robot cuando era manejado por el fisioterapeuta o cuantas veces el ratón en la pantalla ha estado en una posición fija, y cuantas veces a estado parado el paciente, así como la comparación de la mismas.

El índice de es el numero de veces que el robot esta parado.

Máxima realción Robot/Paciente Fecha: 2014-05-21 13:34:52 n° Terapia : 310 Diferencia de veces: 354
 Mínima realción Robot/Paciente Fecha: 2014-04-28 12:48:39 n° Terapia : 259 Diferencia de veces: -139
 Primera realción Robot/Paciente Fecha: 2014-04-28 10:49:33 n° Terapia : 230 Diferencia de veces: 118
 Última realción Robot/Paciente Fecha: 2014-06-03 16:58:08 n° Terapia: 665 Diferencia de veces: 49
 Media de veces que el paciente esta parado respecto al robot: 130

Número de temblores



Paciente 7 Simetría direccional x e y Terapia 2

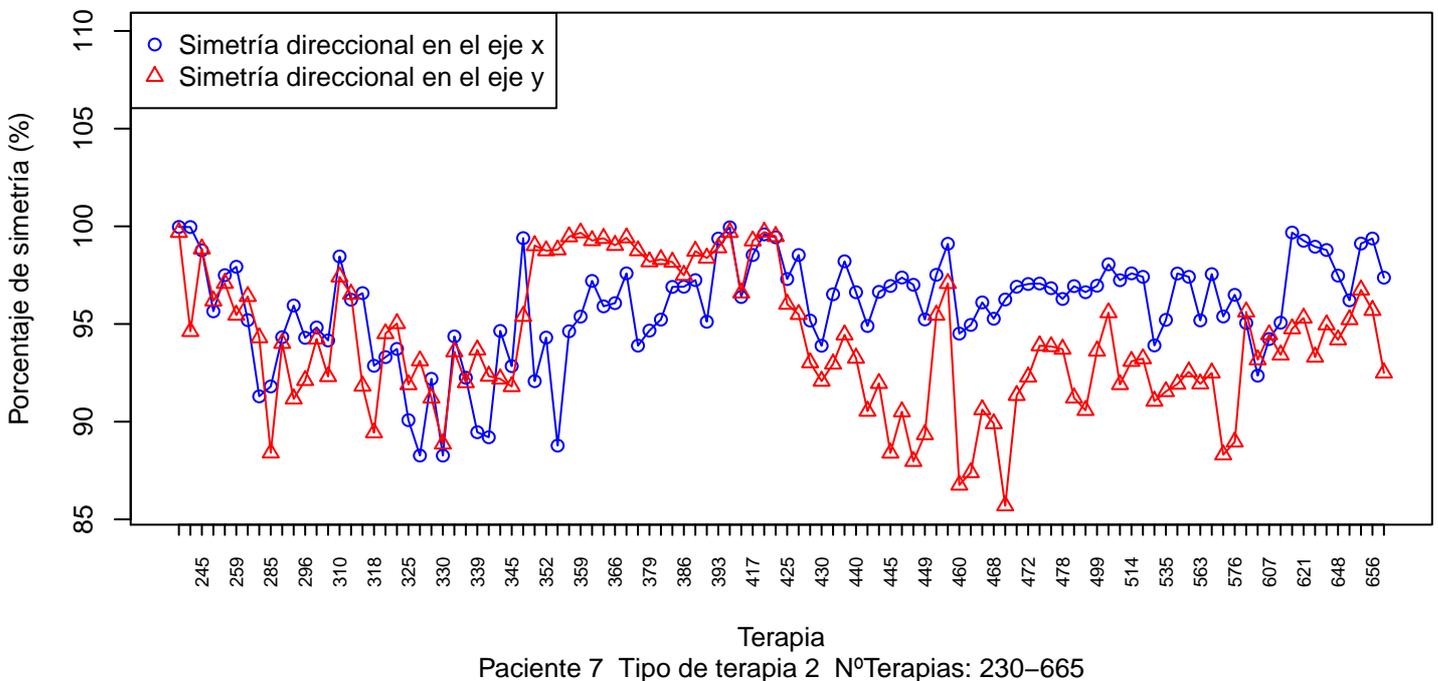
Fecha de impresión: 2015-09-01 23:30:32

En el análisis estadístico de una serie temporal y en el procesamiento de la señal, la simetría direccional es una medida estadística de rendimiento de un modelo para predecir el cambio, positivo o negativo, de una serie temporal de un período de tiempo al siguiente. El índice de la simetría direccionales es el porcentaje de simetría direccional(%).

Máxima Simetría direccional x Fecha: 2014-04-28 10:49:33 Terapia número: 230 Simetría direccional: 100 %
 Mínima Simetría direccional x Fecha: 2014-05-21 14:14:03 Terapia número: 330 Simetría direccional: 89 %
 Primera Simetría direccional x Fecha: 2014-04-28 10:49:33 Terapia número: 230 Simetría direccional: 100 %
 Última Simetría direccional x Fecha: 2014-06-03 16:58:08 Terapia número: 665 Simetría direccional: 98 %
 Media de la Simetría direccional x : 96 %

Máxima Simetría direccional y Fecha: 2014-06-02 16:16:40 Terapia número: 422 Simetría direccional: 100 %
 Mínima Simetría direccional y Fecha: 2014-06-03 16:23:04 Terapia número: 469 Simetría direccional: 86 %
 Primera Simetría direccional y Fecha: 2014-04-28 10:49:33 Terapia número: 230 Simetría direccional: 100 %
 Última Simetría direccional y Fecha: 2014-06-03 16:58:08 Terapia número: 665 Simetría direccional: 93 %
 Media de la Simetría direccional y : 95 %

Simetría direccional en el eje x y en el eje y



Paciente 7 Simetría velocidad y ángulo Terapia 2

Fecha de impresión: 2015-09-01 23:30:32

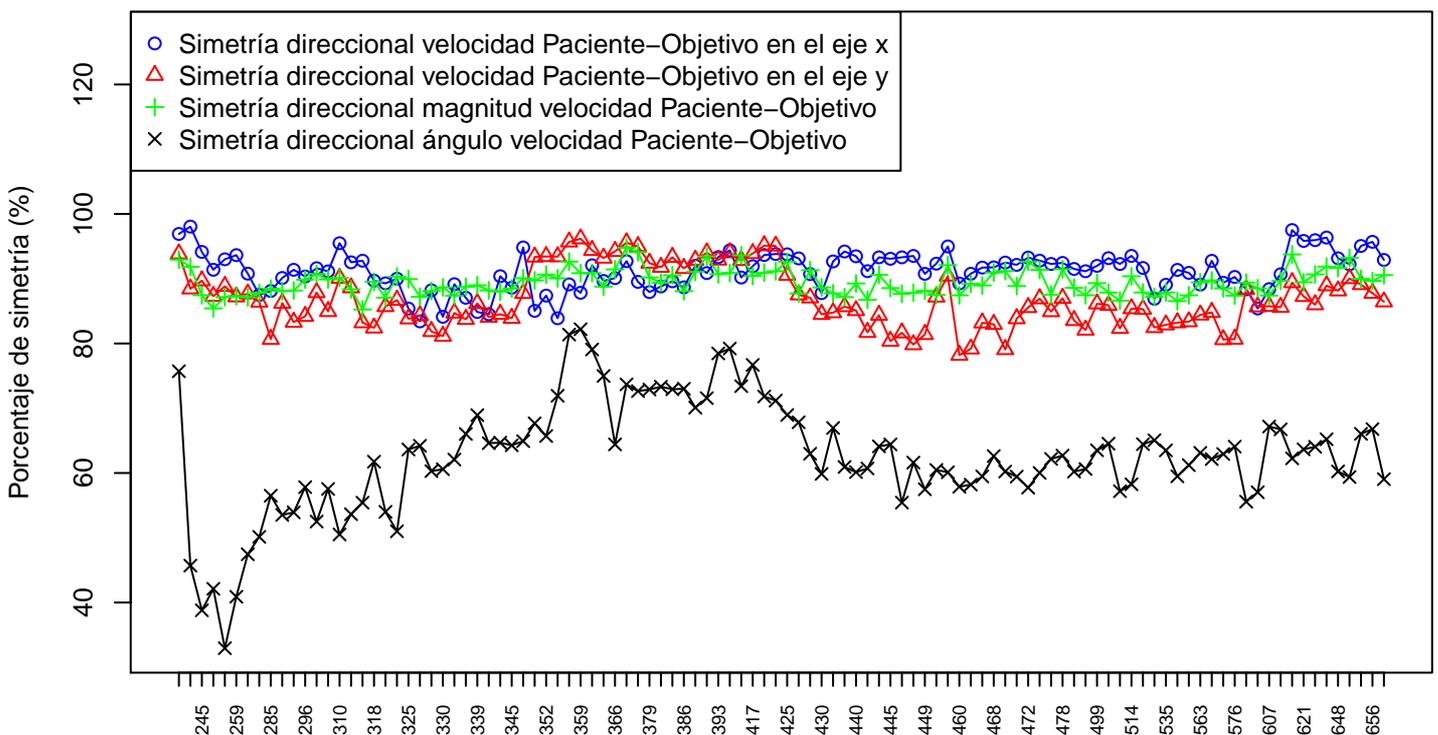
Simetría direccional de la velocidad en x y en y, la magnitud de la velocidad es la semisuma al cuadrado de las velocidades x e y y el ángulo es el arcoseno de la velocidad con respecto a la magnitud lo que vemos en la pantalla es el porcentaje de que ángulo sea el mismo del paciente y de la trayectoria objetivo.

Los índices son porcentajes de simetría(%).

Máxima simetría angular Fecha: 2014-06-02 13:57:39 nº Terapia: 359 Porcentaje de simetría: 83 %
 Mínima simetría angular Fecha: 2014-04-28 12:35:46 nº Terapia: 257 Porcentaje de simetría: 33 %
 Primera simetría angular Fecha: 2014-04-28 10:49:33 nº Terapia: 230 Porcentaje de simetría: 76 %
 Última simetría angular Fecha: 2014-06-03 16:58:08 nº Terapia: 665 Porcentaje de simetría: 60 %
 Media de la simetría angular : 63 %

Máxima magnitud de velocidad Fecha: 2014-06-02 15:54:58 nº Terapia: 374 Porcentaje de simetría: 95 %
 Mínima magnitud de velocidad Fecha: 2014-05-21 13:42:47 nº Terapia: 316 Porcentaje de simetría: 86 %
 Primera magnitud de velocidad Fecha: 2014-04-28 10:49:33 nº Terapia: 230 Porcentaje de simetría: 93 %
 Última magnitud de velocidad Fecha: 2014-06-03 16:58:08 nº Terapia: 665 Porcentaje de simetría: 91 %
 Media de la simetría de la magnitud de velocidad: 90 %

Simetría direccional



Paciente 7 Evolucion de la fuerza Terapia 2

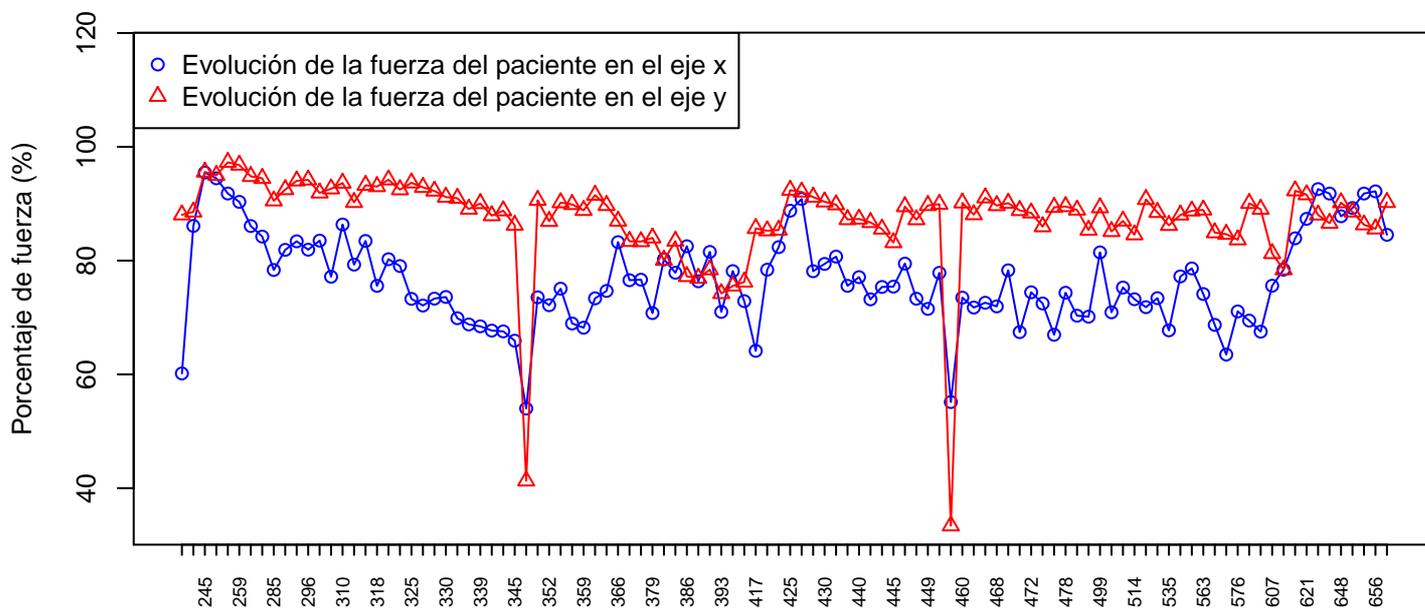
Fecha de impresión: 2015-09-01 23:30:32

Esta medida va a marcar la evolución de la fuerza que tenga el paciente, para ello lo que valoramos es no solo la fuerza que aplica el paciente sino también la fuerza con la que le ayuda el robot. Para calcular la evolución vamos a tomar la fuerza que aplica el paciente y la vamos a dividir entre la suma de la fuerza que aplica el robot y la fuerza que aplica el paciente. El índice será un porcentaje de la fuerza en x y en y que realiza el paciente con respecto a la fuerza que hace el paciente y la fuerza asistida que recibe, se medirá con un porcentaje (%).

Máxima evolución en x Fecha: 2014-04-28 12:25:10 Terapia número: 245 Evolución: 95.48547 %
 Mínima evolución en x Fecha: 2014-06-02 12:20:34 Terapia número: 346 Evolución: 53.99935 %
 Primera evolución en x Fecha: 2014-04-28 10:49:33 Terapia número: 230 Evolución: 60.16772 %
 Última evolución en x Fecha: 2014-06-03 16:58:08 Terapia número: 665 Evolución: 84.53417 %
Media de la evolución en x : 77 %

Máxima evolución en y Fecha: 2014-04-28 12:35:46 Terapia número: 257 Evolución: 97.279 %
 Mínima evolución en y Fecha: 2014-06-02 17:08:17 Terapia número: 452 Evolución: 33.39253 %
 Primera evolución en y Fecha: 2014-04-28 10:49:33 Terapia número: 230 Evolución: 88.04851 %
 Última evolución en y Fecha: 2014-06-03 16:58:08 Terapia número: 665 Evolución: 90.23464 %
Media de la evolución en y : 88 %

Evolución de la fuerza





ANEXO 2: IMPRESIÓN PDF



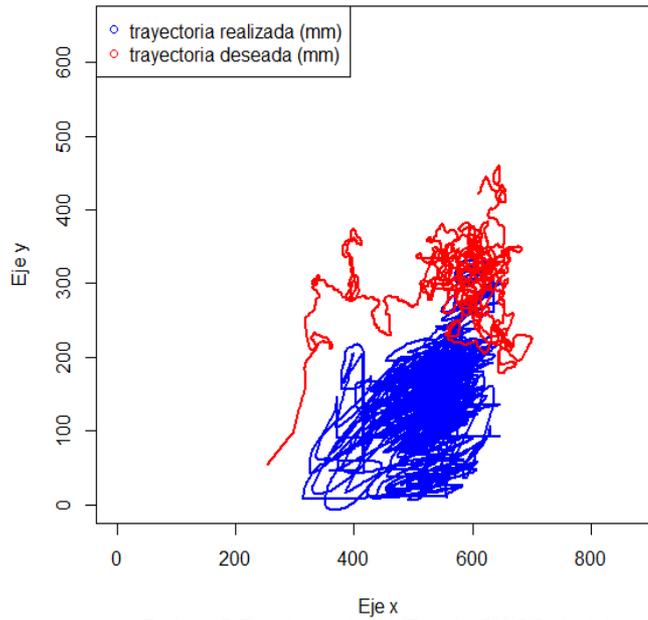
ESCUELA DE INGENIERÍAS
INDUSTRIALES

Miguel Zatarain Valles



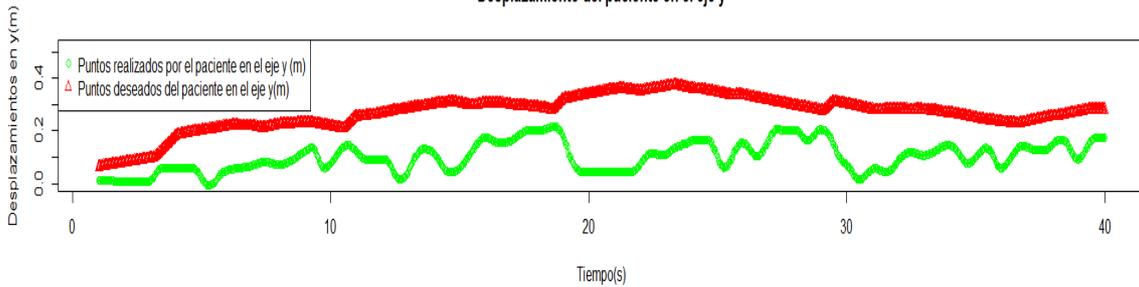
Fecha de impresión: 2015-09-01 23:32:24

Trayectoria del paciente respecto a la deseada

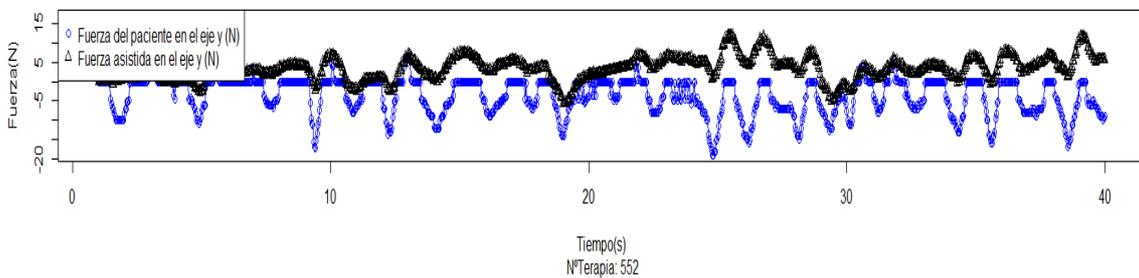


Paciente 7 Tipo de terapia 1 N°Terapia: 552 Dificultad: 1

Desplazamiento del paciente en el eje y

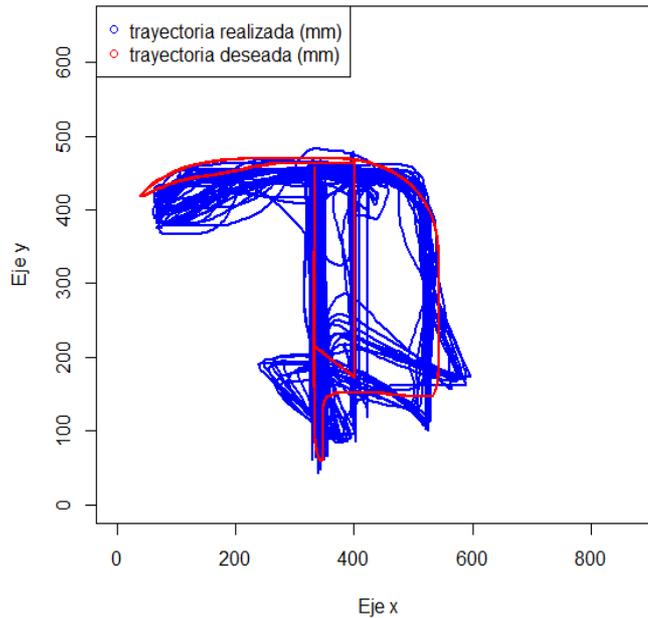


Fuerza del paciente en el eje y





Trayectoria del paciente respecto a la deseada



Paciente 14 Tipo de terapia 2 N°Terapia: 625 Dificultad: NA

Velocidad del paciente en el eje x

