

Trabajo Fin de Máster

Reconocimiento de Gestos Mediante
Acelerómetro y Giróscopo para Implementación de
Driver de Teclado en Aula Virtual

Gesture Recognition whit Gyroscope and
Accelerometer for Implementing a Keyboard Driver
in a Virtual Classroom

Autor

Javier Atrián Blasco

Director/

D. José Elías Herrero Jaraba

Escuela de Ingeniería y Arquitectura

2016

1. Resumen

En el presente proyecto se ha realizado un estudio sobre distintas técnicas en el reconocimiento de gestos para la realización de un “teclado” inercial. Este trabajo pretende generar una aplicación capaz de reconocer, mediante gestos, los distintos caracteres alfanuméricos.

El trabajo afronta el reto de disponer única y exclusivamente de dos sensores inerciales, acelerómetro y giróscopo. Por este motivo, el procesamiento de las señales es de vital importancia y es diferencial a la hora del reconocimiento.

El más importante reto que este proyecto afronta es el desarrollo y estudio de distintos métodos matemáticos, muy usados para el reconocimiento de patrones, que permita discernir cual es el más eficaz de todos ellos. Para poder realizar este estudio es necesario realizar una base de datos mínima, con todos los patrones correspondientes a los 10 dígitos numéricos 0-9 y a las 26 letras del abecedario anglosajón con los que realizar los entrenamientos y estudios necesarios.

Una vez realizado dicho estudio se realiza una aplicación que, recibiendo los datos de entrada con el mismo sistema que en las capturas realizadas para la base de datos y apoyándonos en las soluciones extraídas de los estudios, toma una decisión sobre el carácter realizado.

Para lograr estos objetivos se ha optimizado un algoritmo de compensación de la gravedad y se han buscado los mejores algoritmos para el reconocimiento de patrones, tratando de maximizar su eficacia dentro de nuestra base de datos. Para ello se ha usado la extendida herramienta informática, Matlab, que permite el diseño de algoritmos de manera ágil y eficiente para posteriormente, y también con dicha herramienta, se realiza una APP mediante su utilidad para el diseño de GUI (Interfaces Gráficas de Usuario).

En el primer capítulo incluimos una introducción, repasando el estado del arte del reconocimiento de caracteres y, más en concreto, en sistemas inerciales. Además, se incluye la motivación y objetivos del proyecto.

En el segundo capítulo se recogen los primeros desarrollos del proyecto, que comprenden desde el desarrollo HW hasta el procesamiento llevado a cabo para la generación de trayectorias fieles al gesto real. Pasando por el FW y el método de captura de señales. Además, se puede observar el reparto de señales en la captura y una guía de procedimiento.

En el tercer capítulo se realiza una breve explicación de las tres técnicas escogidas para el reconocimiento de patrones temporales obtenidos.

En los capítulos cuarto, quinto y sexto se exponen respectivamente resultados, conclusiones y una reflexión sobre futuros trabajos que podrían mejorar nuestro estudio o caminos alternativos que podrían enfocar el desarrollo final de una aplicación llevada a la vida real y cotidiana del usuario medio.

2. Índice

1.	Resumen	1
2.	Índice	4
3.	Índice de Figuras.....	6
4.	Índice de Tablas.....	8
5.	Introducción.....	9
5.1.	Preámbulo	9
5.2.	Objeto del Proyecto	9
5.2.1.	Motivación	9
5.2.2.	Objetivos.....	10
6.	Primeros Pasos	11
6.1.	Plataforma desarrollo.....	11
6.1.1.	Hardware	11
6.1.2.	Firmware	12
6.2.	Realización de Patrones	13
6.2.1.	Números:	14
6.2.2.	Letras:.....	14
6.3.	Procesado de Patrones.....	15
6.3.1.	Captura.....	15
6.3.2.	Segmentación	15
6.3.3.	Compensación Gravedad	16
6.3.4.	Normalización	17
7.	Técnicas Matemáticas Estudiadas	19
7.1.	DTW (Alineamiento temporal dinámico)	19
7.2.	HMM (Modelos Ocultos de Markov)	20
7.3.	NN (Redes Neuronales Aplicadas al Reconocimiento de Patrones)	22
8.	Resultados.....	23
8.1.	DTW (Alineamiento temporal dinámico)	23
8.1.1.	Resultados estudio de números (aceleración)	23
8.1.2.	Resultados estudio de números (trayectoria)	25
8.1.3.	Resultados estudio de números (aceleración + trayectorias)	26
8.1.4.	Resultados estudio de letras (aceleración).....	28

8.1.5.	Resultados estudio de letras (trayectoria).....	30
8.1.6.	Resultados del estudio de letras (aceleración + trayectorias).....	31
8.2.	HMM (Modelos Ocultos de Markov)	33
8.2.1.	Resultados estudio de números (aceleración)	33
8.2.2.	Resultados estudio de números (trayectoria)	35
8.2.3.	Resultados estudio de números (aceleración + trayectoria).....	36
8.2.4.	Resultados estudio de letras (aceleración).....	37
8.2.5.	Resultados estudio de letras (trayectoria).....	40
8.2.6.	Resultados estudio de letras (aceleración + trayectoria)	43
8.3.	NN (Redes Neuronales Aplicadas al Reconocimiento de Patrones)	45
8.3.1.	Resultados estudio de números (aceleración)	45
8.3.2.	Resultados estudio de números (trayectoria)	47
8.3.3.	Resultados estudio de números (aceleración + trayectoria).....	48
8.3.4.	Resultados estudio de letras (aceleración).....	49
8.3.5.	Resultados estudio de letras (trayectoria).....	52
8.3.6.	Resultados estudio de letras (trayectoria+aceleración)	54
9.	Conclusiones	56
10.	Futuros Desarrollos.....	57
11.	Bibliografía	58
12.	Anexo A.....	59
13.	Anexo B.....	61

3. Índice de Figuras

Fig. 1. Xbox Kinect	9
Fig. 2. Mando Nintendo Wii	9
Fig. 3. Mando a Distancia LG Smart TV	10
Fig. 4. Modelo Final Sistema Bolígrafo-Inercial.....	11
Fig. 5. Grafo de Estados Simplificado FW.....	13
Fig. 6. Sujeción Adecuada Dispositivo Inercial	13
Fig. 7. Guía de Trayectorias y Resultados para Patrones Numéricos.....	14
Fig. 8. Guía de Trayectorias y Resultados para Patrones Alfabéticos	15
Fig. 9. Evolución Algoritmo Compensación Gravedad	16
Fig. 10. Ejemplo de Compensación de Aceleración de la Gravedad	17
Fig. 11. Ecuaciones y Gráficos DTW	19
Fig. 12. Ejemplo reacondicionamiento para HMM	21
Fig. 13. Red Neuronal Tipo Seleccionada	22
Fig. 14. Evolución de aciertos.....	23
Fig. 15. Decisión DTW Números Aceleración	24
Fig. 16. Evolución de aciertos en función de ventana de adaptación temporal tomando como patrón las trayectorias.	25
Fig. 17. Decisión DTW Números Trayectorias.....	26
Fig. 18. Decisión DTW Números Aceleración + Trayectoria.....	27
Fig. 19. Evolución de aciertos.....	28
Fig. 20. Decisión DTW Letras Aceleración	29
Fig. 21. Evolución de aciertos en función de ventana de adaptación temporal tomando como patrón las trayectorias.	30
Fig. 22. Decisión DTW Letras Trayectorias	31
Fig. 23. Decisión DTW Letras Trayectorias+Aceleración	32
Fig. 24. Evolución de aciertos.....	33
Fig. 25. Decisión HMM Números Aceleración.....	34
Fig. 26. Evolución de aciertos en función de ventana de adaptación temporal tomando como patrón las trayectorias.	35
Fig. 27. Decisión HMM Números Trayectorias.....	36
Fig. 28. Decisión HMM Números con Aceleración Trayectoria.....	37
Fig. 29. Evolución de aciertos en función de ventana de adaptación temporal tomando la aceleración como patrón.	38
Fig. 30. Decisión HMM Letras Aceleración.....	39
Fig. 31. Evolución de aciertos.....	40
Fig. 32. Decisión HMM Letras Trayectorias.....	42
Fig. 33. Decisión HMM Letras Aceleración Trayectoria.....	44
Fig. 34. Evolución de aciertos en función de ventana de número de muestras de entrada y neuronas en la capa oculta.	45
Fig. 35. Decisión NN Números con Aceleración	46
Fig. 36. Evolución de aciertos.....	47
Fig. 37. Decisión DTW Números Trayectorias.....	48

Fig. 38. Decisión NN Números Trayectorias.....	49
Fig. 39. Evolución de aciertos.....	49
Fig. 40. Decisión NN Letras Aceleración.....	51
Fig. 41. Evolución de aciertos.....	52
Fig. 42. Decisión NN Letras Aceleración.....	53
Fig. 43. Evolución de aciertos.....	54
Fig. 44. Decisión NN Letras Aceleración.....	55
Fig. 45. Pantallazo de la Aplicación	56

4. Índice de Tablas

Tabla 1. Validación Cruzada DTW Números Aceleración.....	24
Tabla 2. Validación Cruzada DTW Números Trayectorias.....	25
Tabla 3. Validación Cruzada DTW Números Aceleración + Trayectoria.....	26
Tabla 4. Validación Cruzada DTW Letras Aceleración.....	28
Tabla 5. Validación Cruzada DTW Letras Trayectorias.....	30
Tabla 6. Validación Cruzada DTW Letras Aceleración + Trayectorias.....	31
Tabla 7. Validación Cruzada HMM Números Aceleración	34
Tabla 8. Validación Cruzada HMM Números Trayectorias	35
Tabla 9. Validación Cruzada HMM Números Aceleración + Trayectoria	36
Tabla 10. Validación Cruzada HMM Letras Aceleración	38
Tabla 11. Validación Cruzada HMM Letras Trayectorias	41
Tabla 12. Respuesta HMM Letras Aceleración Trayectoria	43
Tabla 13. Validación Cruzada NN Números Aceleración	46
Tabla 14. Validación Cruzada NN Números Trayectorias.....	47
Tabla 15. Validación Cruzada NN Números Aceleración Trayectoria	48
Tabla 16. Respuesta NN Números-Aceleración	50
Tabla 17. Respuesta NN Letras-Trayectorias	52
Tabla 18. Respuesta NN Letras (Trayectorias + Aceleración)	54
Tabla 19. Resumen Resultados	56

5. Introducción

5.1. Preámbulo

Con la aparición de los nuevos soportes informáticos, móviles, tabletas o televisores interactivos, cada día se hace más presente la necesidad de innovar en el ámbito de la interacción hombre maquina. Recientemente muchas investigaciones se centran en el estudio de IHM mediante gestos, tanto por reconocimiento visual como por sensores inerciales.

Hasta ahora la industria solo ha llegado al reconocimiento de gestos simples, Xbox kinect por método visual, o los conocidos mandos de la video consola de nintendo, Wii, aplicando sensores inerciales. Pero ninguna gran marca se ha lanzado al desarrollo de interfaces capaces de reproducir un teclado mediante gestos.



Fig. 1. Xbox Kinect



Fig. 2. Mando Nintendo Wii

En estudios de diversas universidades se ha tratado este tema con anterioridad. Aplicando distintas técnicas con resultados diversos, pero todos ellos actúan con bases de datos propias, generadas con sus sistemas propios de manera que unificando el estudio podremos sacar unas mejores conclusiones.

Algunas de las técnicas usadas son DTW, HMM, redes neuronales [1] – [6]. De las cuales se obtienen resultados notables en el reconocimiento individual de caracteres.

5.2. Objeto del Proyecto

5.2.1. Motivación

La idea de este TFM surgió en el verano del año 2015 al descubrir que el último televisor en llegar a mi casa tenía un sensor inercial que permitía manejar algunas de las funciones especiales que incorporaba.

Una de ellas es la navegación por internet, que siempre requiere la escritura de pequeñas direcciones. Esto se hace llevando el cursor hacia las teclas ficticias de un teclado impreso en la pantalla. Esto se convierte en un proceso tedioso y desagradable que no deja muy buena sensación del acabado sobre el usuario y que requiere constantes borrados mientras escribimos.



Fig. 3. Mando a Distancia LG Smart TV

Coincidiendo con mi experiencia laboral, pensé que sería capaz de desarrollar una aproximación a un sistema que sustituyese al anteriormente comentado mediante la escritura en el espacio y la captura de este gesto mediante sensores inerciales.

5.2.2. Objetivos

Los objetivos planteados inicialmente fueron los siguientes:

- Generar un HW que fuese capaz de capturar las señales necesarias para el proyecto.
- Generar una base de datos de caracteres alfanuméricos.
- Realizar un estudio sobre los mejores métodos matemáticos para el reconocimiento de patrones y los que más se adecuan a mi sistema de captura y procesado.
- Realizar una pequeña aplicación capaz de aprovechar de los resultados obtenidos para hacer una demostración del funcionamiento.

6. Primeros Pasos

6.1. Plataforma desarrollo

Se toma la realización de una plataforma de desarrollo como el punto inicial del proyecto. Esta plataforma constara de dos partes el HW propietario diseñado mediante Altium Designer y el software necesario, pudiendo dividir entre el FW del microcontrolador que controla el HW y que por otro lado se encarga de comunicarse con el PC.

El PC, mediante el programa informático Matlab, nos permite capturar y almacenar las señales para posteriormente realizar el estudio y hasta la aplicación comentada con anterioridad.

6.1.1. Hardware

El Hardware está formado principalmente por cinco bloques:

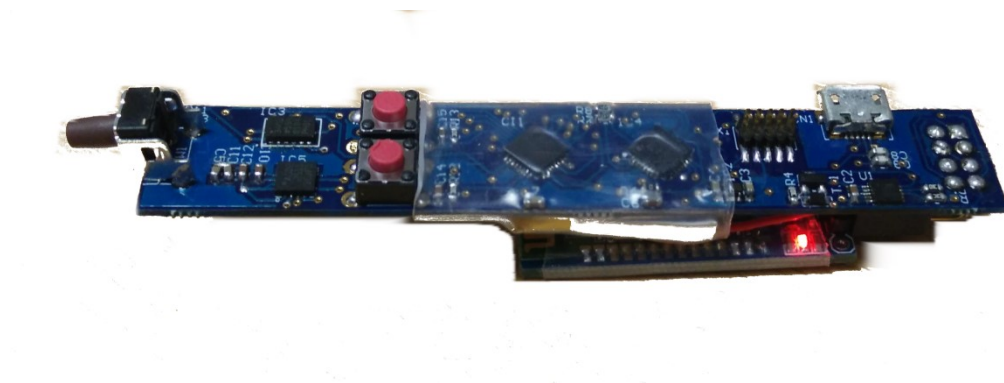


Fig. 4. Modelo Final Sistema Bolígrafo-Inercial

- Microcontrolador:

Se ha decidido escoger el microcontrolador kl16z128vfm4 del fabricante NXP, anteriormente Freescale, de la familia Kinetis.

Sus principales características son:

- Consumos muy contenidos, llegando a los 2uA en estado standby.
- 32 bits
- Memoria flash de 128kB y RAM de 32 kB.
- PinOut QFN32
- Un software para el desarrollo de FW muy intuitivo, basado en la extendida plataforma eclipse.

- ADXL345 (Acelerómetro)

Extendido acelerómetro del fabricante Analog Devices, se eligió este acelerómetro por las siguientes características.

- Ultra bajo consumo
- 10-bit de resolución
- Sensibilidad ± 2 , ± 4 , ± 8 , ± 16
- ODR desde 12.5 Hz hasta 3200 Hz

- L3GD20 (Giróscopo)

Un giróscopo muy extendido por muchos de los móviles que hoy en día se encuentran en el mercado, con unas buenas prestaciones su punto fuerte está en la parte económica, muy barato en comparación con otros de sus rivales directos. Sus principales características son las siguientes:

- Escala (250/500/2000 dps)
- 16 bit FS
- ODR 90/190/380/760

- HC-06 (Modulo de comunicaciones Bluetooth 2.0)

Es un modulo muy extendido entre los aficionados a la electrónica DIY, esto es debido a su simpleza. Es muy fácil de configurar y la puesta en marcha realmente rápida, la comunicación con el sistema se realiza mediante SCI, comúnmente llamado puerto serie, y en la conexión con el ordenador emula otro puerto serie. Esto es así porque trabaja en el perfil spp.

- El sistema ha sido dotado también de pulsadores como interfaz con la maquina.

En el Anexo A se encuentran los planos y todo el material referente al diseño HW.

6.1.2. Firmware

El firmware programado en el microcontrolador del dispositivo se ha desarrollado con la intención de ser invariante a lo largo del proyecto. Así, este software servirá sin ningún cambio, tanto para la captura, como para su posterior uso en la aplicación desarrollada.

Esto es debido a la necesidad de tener señales sin ninguna variación en su muestreo, ya sea de frecuencia, cambios en la comunicación u otros problemas que podrían suceder.

De tal manera se ha desarrollado un sistema basado en maquina de estados que permite actuar de manera sencilla sobre el dispositivo para obtener unos resultados rápidos y eficaces.

En la siguiente figura podemos ver un grafico con una morfología simplificada de la máquina de estados, mientras en el Anexo B se entra más en detalle a cerca del firmware del micro controlador.



Fig. 5. Grafo de Estados Simplificado FW

6.2. Realización de Patrones

Para realizar el estudio planteado necesitamos una base de datos mínima con la que trabajar. Así se recogen trayectorias realizadas por dos personas organizándose diferenciando entre letras y números.

La metodología para recoger dicha base de datos a de ser común con las posteriores pruebas que se quieran realizar. Así se plantea una metodología que incluye sujeción y trayectorias únicas por cada carácter. Así, y aunque la intención del proyecto es independizar el sistema de la orientación del dispositivo, la forma adecuada de sujetar el dispositivo tiene que seguir la postura de la mano que se puede observar en la siguiente imagen.

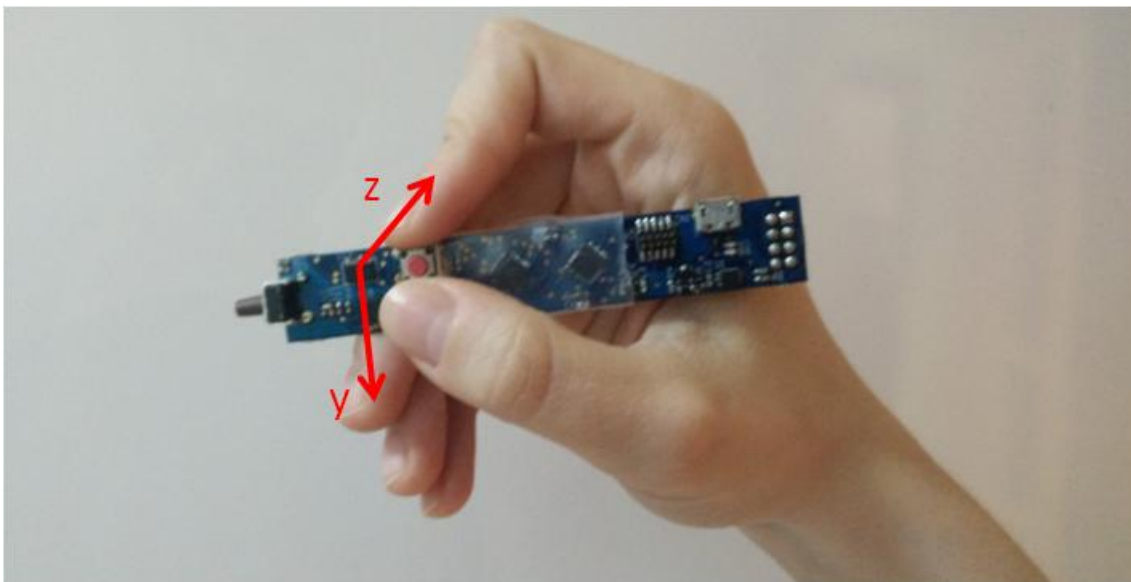


Fig. 6. Sujeción Adecuada Dispositivo Inercial

Esto tiene que ser así, debido a la necesidad de tener los ejes con principales aceleraciones contenidos en el plano sobre el que se dibuja el patrón. Si la postura de sujeción cambia el patrón final se verá modificado al pasar las aceleraciones principales a ejes no valorados por el algoritmo.

En los siguientes apartados se pueden observar las trayectorias planteadas para el proyecto. Como principal característica común podemos observar que todas se realizan mediante un solo trazo y que tienen un sentido de dibujado único y particularizado a cada carácter.

6.2.1. Números:

Para el caso particular de los números se recogen 15 dígitos por persona, alternando planos horizontales al suelo y verticales de manera que el número resultante de patrones es $2 \times 15 \times 10 = 300$. Quedando repartidos aleatoriamente en el 60% para entrenamiento y el 40% para test, 18 y 12 patrones respectivamente.

Para la realización de los patrones se utiliza la siguiente guía (a), que tras realizarse mediante el dispositivo y sometidos al procesamiento obtienen como resultante los caracteres situados en la imagen (b).

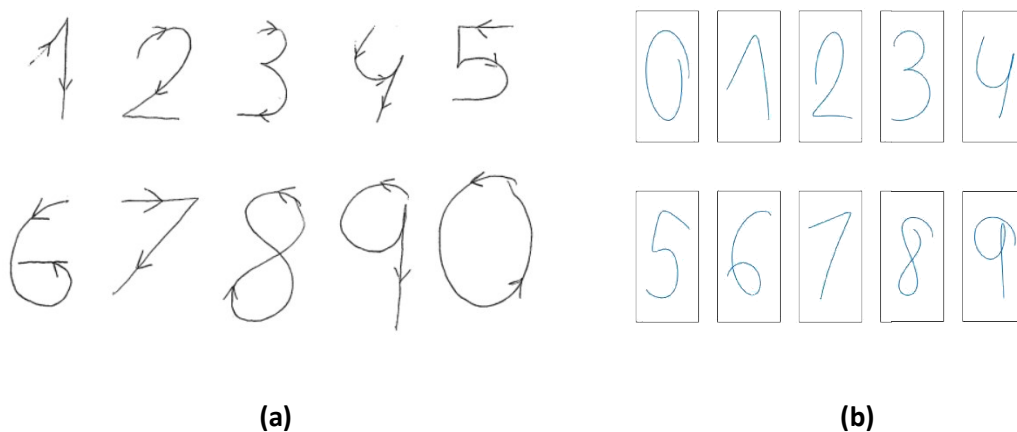


Fig. 7. Guía de Trayectorias y Resultados para Patrones Numéricos

6.2.2. Letras:

En el caso de las letras se han recogido 10 patrones por persona ($2 \times 26 \times 10 = 520$). Quedando repartidos con los mismos porcentajes que los números, 60-40% quedan 12 patrones de entrenamiento y 8 de test.

De igual manera que en el apartado correspondiente a los números, en la imagen (a) se puede observar la guía de las trayectorias para las letras. Mientras que en la (b) se pueden observar sus resultados.

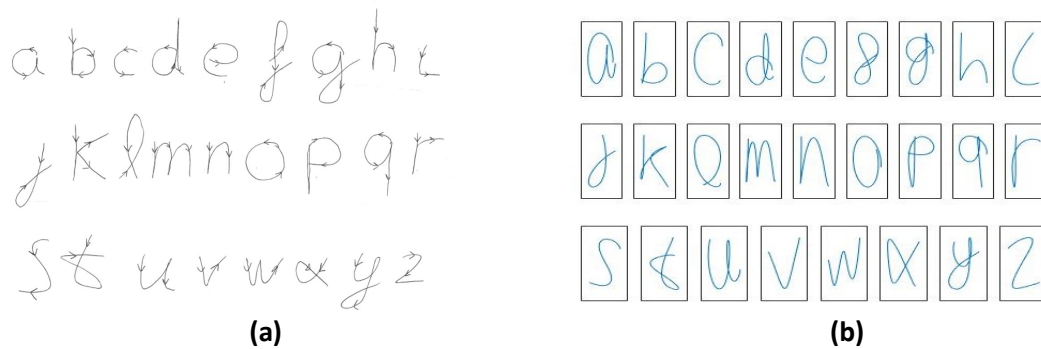


Fig. 8. Guía de Trayectorias y Resultados para Patrones Alfabéticos

6.3. Procesado de Patrones

6.3.1. Captura

El sistema de bolígrafo-inercial está diseñado para poder empezar la captura desde la pulsación de cualquiera de sus botones hasta que se suelte dicho botón o se superen los 6 segundos de captura, límite impuesto por la memoria RAM del micro controlador.

Esto es debido a la frecuencia de muestreo seleccionada, 100 Hz, que exige 6 buffers e por cada sensor correspondientes ambos a los ejes X Y y Z. Lo que hace un total de $600 \times 3 \times 2 \times 2 = 7200$ bytes. Una vez capturadas las señales se realiza la comunicación de cada uno de los 6 buffers por separado.

Configuración Acelerómetro:

- Frecuencia 100 Hz
- FS $\pm 4g$

Configuración Giróscopo:

- Frecuencia 100 Hz
- FS500 dps

Los buffers son recibidos en nivel digital, para ajustar a valores reales dividimos el valor del acelerómetro entre 128 (correspondiente al valor de aceleración de la gravedad) y el giróscopo para adaptar la señal a radianes se multiplica por 0.0175 grados por segundo y multiplicamos por $2\pi/360$ para pasarlo a radianes/segundo. Finalmente y para que coincidan los ejes de acelerómetro y giróscopo se intercambian los ejes del giróscopo X e Y.

6.3.2. Segmentación

Para realizar la segmentación necesitamos conocer los valores absolutos de aceleración y velocidad angular, tomando como punto de comienzo del patrón, el primer valor superior a un umbral determinado y como punto final el último valor que esté por debajo de dichos umbrales.

Aquí podemos observar el resultado de segmentar un patrón, se trata de un dígito 0. En la figura de la izquierda podemos ver las señales de valor absoluto tanto de aceleración como velocidad angular y donde las líneas rojas indican el comienzo y el final de las señales segmentadas.

En la columna central podemos ver los valores de aceleración, superior, y velocidad angular inferior, otra vez con las bandas indicativas del comienzo y el final. Y ya en la última columna se puede observar las señales de aceleración una vez segmentadas.

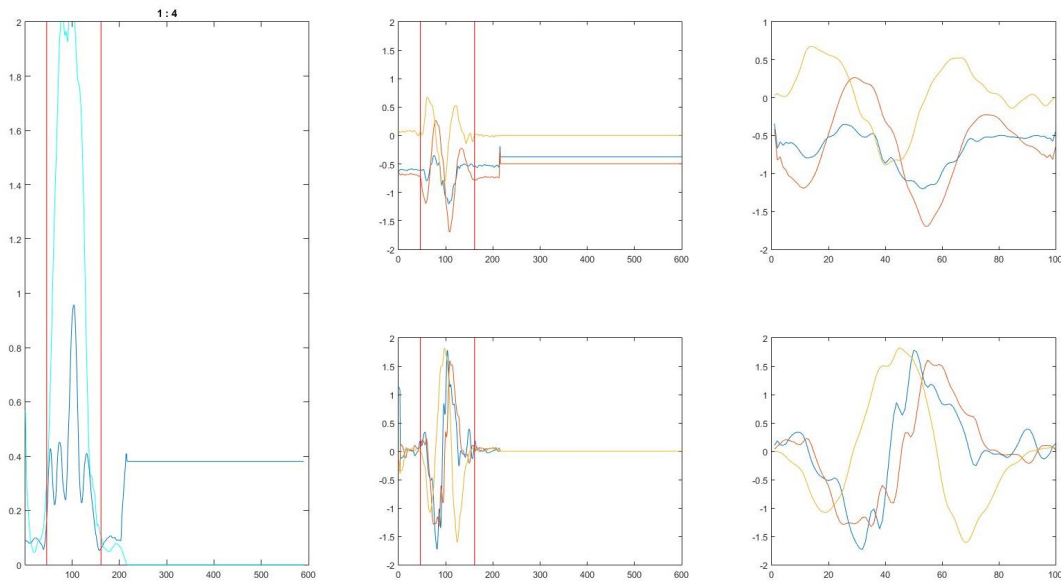


Fig. 9. Evolución Algoritmo Compensación Gravedad

Una vez segmentadas se extrapolan o interpolan puntos dependiendo de la longitud del segmento temporal final y se recalculan las señales para que todas tengan una longitud de 100 muestras.

6.3.3. Compensación Gravedad

Para realizar la compensación de la gravedad se ha recogido un algoritmo presente en estudio anterior [1]. Este estudio se basa en calculando la orientación inicial mediante el acelerómetro y cada una de las orientaciones siguientes, mediante la velocidad angular extraída del giróscopo, recolocar los valores de aceleración en ejes coincidentes en Z con el valor de la gravedad terrestre.

De esta manera se puede eliminar los valores de gravedad y así, poder hacer una extracción de la trayectoria más fiel. Además en nuestro proceso hemos añadido una reorientación de planos para extraer la aceleración de la punta del bolígrafo. En las siguientes imágenes se pueden ver las aceleraciones en los ejes del acelerómetro (a), aceleraciones en el plano de la gravedad (b), aceleraciones con gravedad compensada (c) y las aceleraciones en el plano perpendicular al bolígrafo inercial.

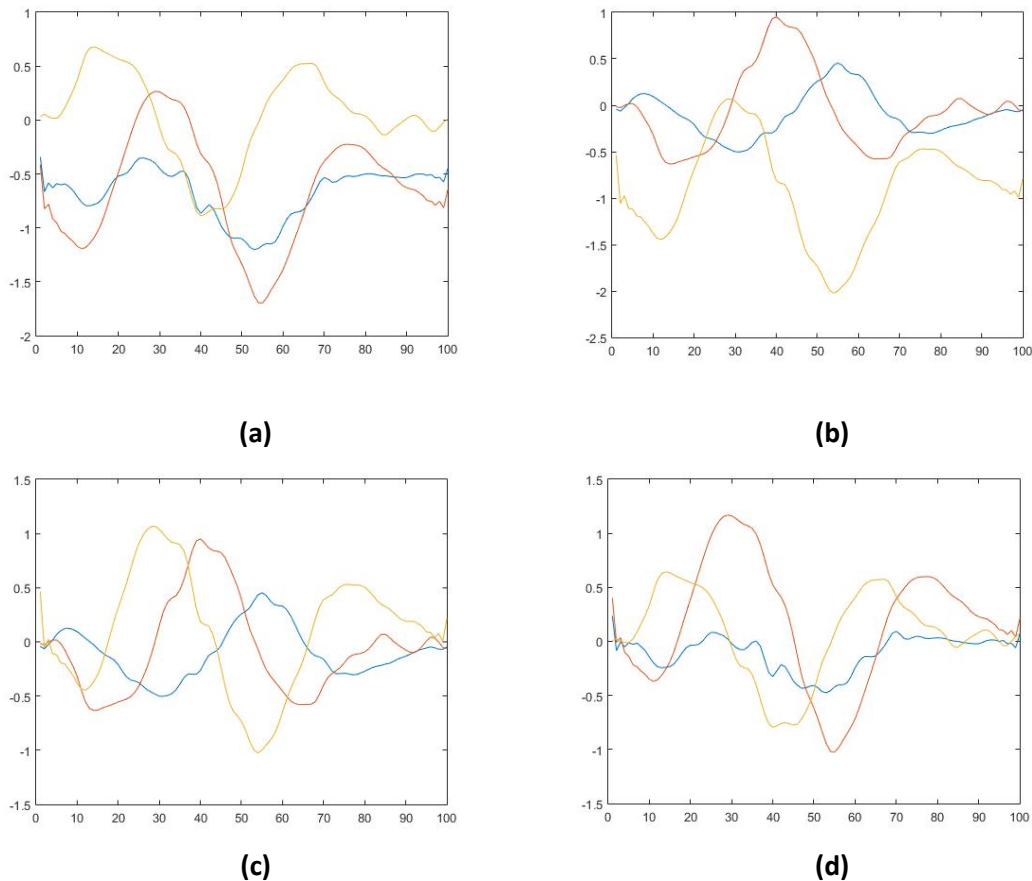


Fig. 10. Ejemplo de Compensación de Aceleración de la Gravedad

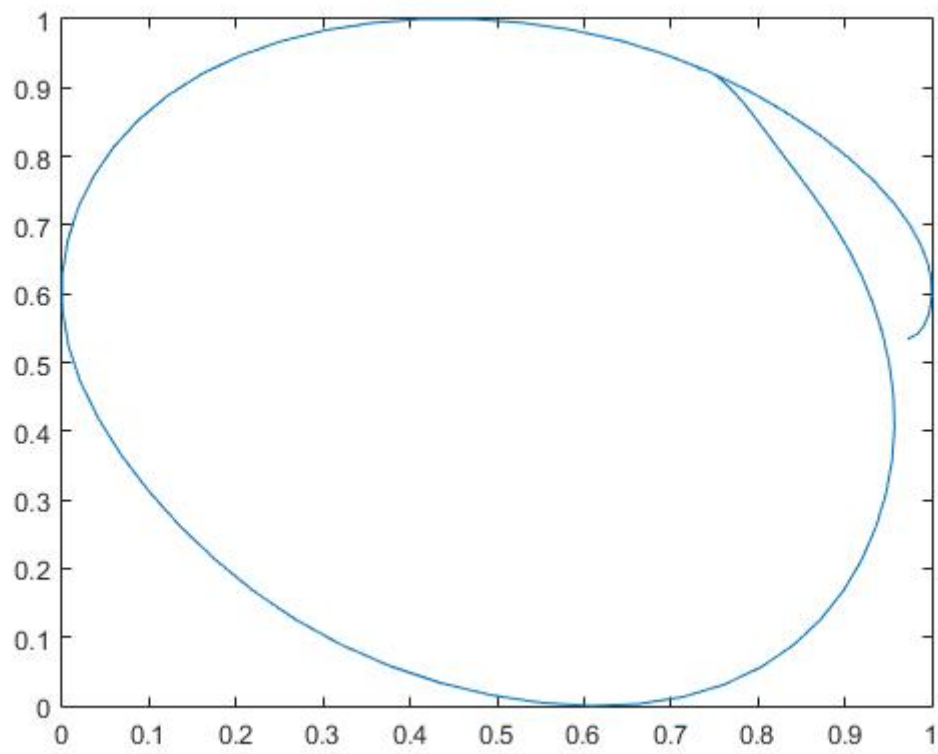
En estas imágenes se puede observar como entre la (a) y la (d) la principal diferencia es que el valor de todos los ejes está centrado en 0 pero las curvas coinciden pudiendo así extraer las trayectorias en función de los ejes del acelerómetro, Y y Z.

En el Anexo C se puede encontrar más información sobre este algoritmo y la aportación de este proyecto.

Una vez extraídas las aceleraciones integramos la señal dos veces para extraer la trayectoria, esta queda almacenada en forma de coordenadas temporales. Siendo la primera correspondiente al punto temporal inicial y la última muestra al último punto.

6.3.4. Normalización

Con la finalidad de independizar las trayectorias de la velocidad y la amplitud con las que han sido dibujadas es necesario normalizar las señales a un valor común. Así, las coordenadas anteriormente comentadas, quedaran normalizadas entre 0 y 1 siendo el punto más inferior en la trayectoria $y=0$ y el superior $y=1$ y con derecha e izquierda de manera similar pero en $x=0$ y $x=1$. El mismo ejemplo de dígito 0 que hemos puesto en casos anteriores se puede observar en la siguiente imagen.



7. Técnicas Matemáticas Estudiadas

Para realizar el reconocimiento se prueban las tres técnicas más extendidas en el mundo del reconocimiento de caracteres escritos, cada una de las técnicas se aplica en cuatro fases. Reconocimiento de las trayectorias para números y letras y, por otro lado, aplicaremos las mismas técnicas a los valores de aceleración, con la intención de comprobar si en las sucesivas integrales que se realizan sobre estas señales para extraer las trayectorias pudiese arrastrarse un error acumulativo que afectase a dichos patrones.

A continuación se encuentra

7.1. DTW (Alineamiento temporal dinámico)

Se trata de una técnica matemática para la comparación de dos trayectorias temporales que tiene como resultado un valor equivalente a la distancia que entre ellas se puede observar. En [8] se puede obtener información sobre el software auxiliar utilizado para estos ensayos.

Partiendo de la base de dos trayectorias, de posición o aceleración, determinadas en el tiempo. Resulta fácil pensar en la posibilidad de aplicar DTW como una técnica de reconocimiento de firmas, cada trayectoria XY correspondería a una firma, de manera que pudiese resultar un algoritmo eficaz.

Este algoritmo se basa en seleccionar la mínima distancia entre dos puntos de dos secuencias temporales distintas. A diferencia de una mínima distancia simple, en este caso se realiza un ajuste temporal, permitiendo seleccionar muestras con $\pm w$ (ventana temporal) como computables. Así, cada muestra correspondiente a la primera señal tendrá una distancia mínima con otra muestra en la segunda señal dentro de esa ventana temporal.

El cómputo de la mínima distancia de todas las muestras de la primera señal es el resultado que indica la coincidencia de una señal con otra. En la siguiente imagen se puede ver un grafico descriptivo del sistema junto a sus ecuaciones. Para más información observar la siguiente referencia [].

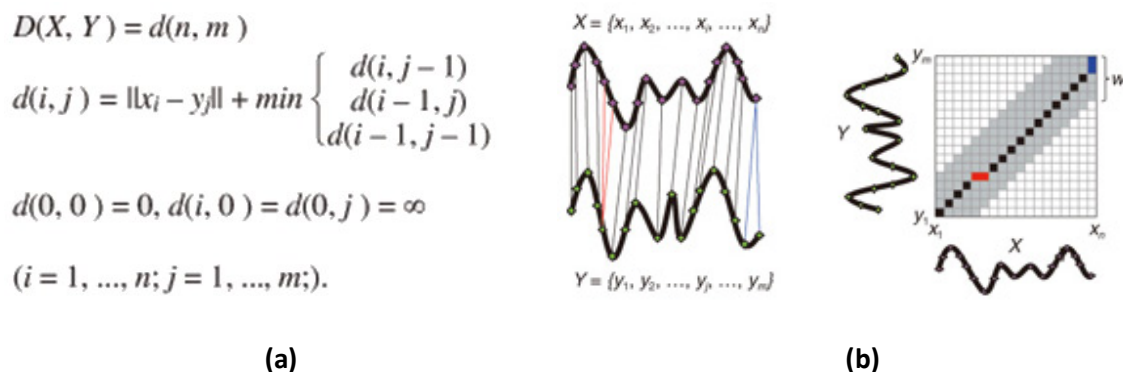


Fig. 11. Ecuaciones y Gráficos DTW

$$DTW(P1_m, P1_n) = x_1$$

.....

- Dados los parámetros del modelo, computar la probabilidad de una secuencia de salida en particular. Este problema se resuelve con el algoritmo de avance-retroceso.
- Dados los parámetros del modelo, encuéntrase la secuencia más probable de estados ocultos que puedan haber generado una secuencia de salida data. Este problema se resuelve con el algoritmo de Viterbi.
- Dada una secuencia de salida o un conjunto de ellas, se puede calcular el número de estados de transición y probabilidades de salida más probables. En otras palabras, podemos entrenar un conjunto de estados de transición a partir de secuencias de datos dadas. Este problema se resuelve con el algoritmo de Baum-Welch.

De estas indicaciones se deduce que para realizar un sistema de reconocimiento deberemos afrontar dos de estas problemáticas.

Como partida de inicio tendremos varios vectores de estados (muestras en una señal temporal) con sus correspondientes estados de observación. Con estos datos podríamos entrenar un modelo oculto con el algoritmo de Baum-Welch para, posteriormente, mediante el algoritmo de avance-retroceso calcular la probabilidad de que, un nuevo vector pertenezca al mismo grupo de vectores entrenamiento.

Para realizar los cálculos de entrenamiento se han usado los patrones iniciales (60% de entrenamiento) y para el cálculo de eficacia (40% de patrones de test).

Para convertir nuestros patrones, trayectorias temporales XY, en cadenas de Markov se han realizado un reacondicionamiento de la señal, ajustando muestras al número de estados y redondeando los valores al número de observaciones.

Así, una muestra temporal de 100 muestras de las coordenadas de X en función del tiempo se puede convertir a 10 muestras (10 estados) con unos valores redondeados de 1 a 5 (estados observables). En las imágenes inferiores se puede observar un ejemplo, donde en (a) se puede ver la señal original y en (b) la señal reacondicionada para 10 estados ocultos y 5 observaciones.

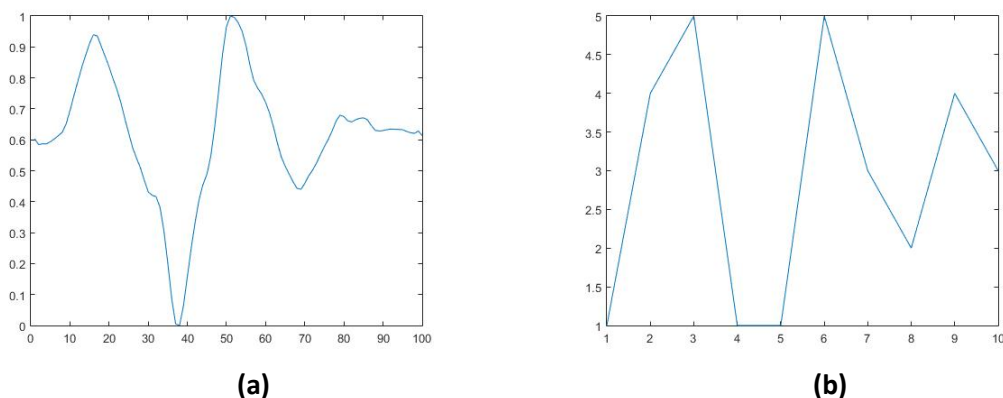


Fig. 12. Ejemplo reacondicionamiento para HMM

Para maximizar los resultados se realiza una comparativa en el que se valora la eficacia en función del número de estados ocultos y del número de estados observables desde 1 hasta 15. En el apartado de resultados se entrara en detalle sobre este tema.

7.3. NN (Redes Neuronales Aplicadas al Reconocimiento de Patrones)

Las redes de neuronas artificiales son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso biológico. Se trata de un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. En inteligencia artificial es frecuente referirse a ellas como redes de neuronas o redes neuronales. En [10] se puede obtener información sobre el software auxiliar utilizado para estos ensayos.

Para realizar este estudio se ha escogido una arquitectura de red determinada compuesta por una capa de entrada de entre 10 y 100 neuronas, una capa intermedia de entre 1 y 40 neuronas con función de transferencia tangencial sigmoidea 'tansig' y tantas neuronas de salida como patrones sea necesario reconocer, 10 para número y 26 para letras. Esta última capa tiene como función de transferencia exponencial normalizada. En la imagen inferior se puede observar una red de 31 patrones de entrada, 15 neuronas en la capa oculta y 10 de salida para el reconocimiento de números.

La función de entrenamiento seleccionada es método de gradiente conjugado.

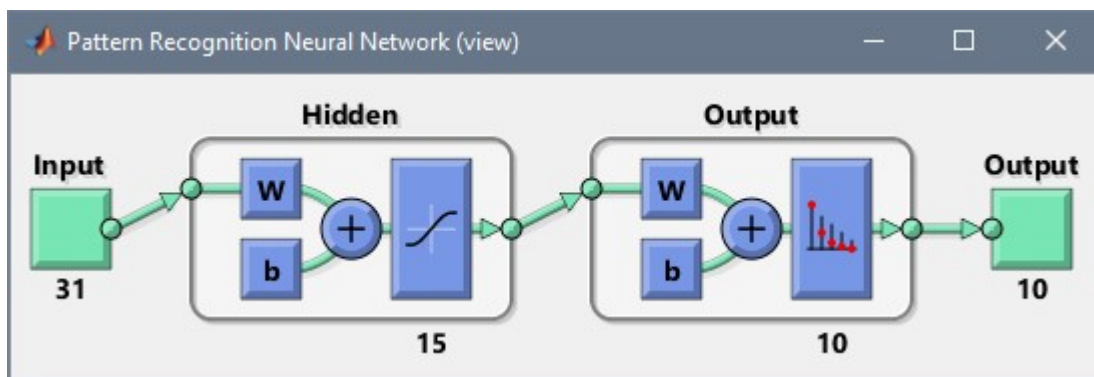


Fig. 13. Red Neuronal Tipo Seleccionada

¿Cómo se organizan los patrones de entrada?

El estudio contiene dos tipos de patrones de entrada, la señal de aceleración de un patrón procesada para que coincida en número de muestras con el número de entradas, repetimos entre 10 y 100, y lo mismo pero para señales de coordenadas en la trayectoria. Además, estos valores de entrada son normalizados a 1.

8. Resultados

Para poder extraer unas conclusiones finales del estudio realizado el formato de los resultados seguirá la siguiente estructura en cada uno de los métodos matemáticos.

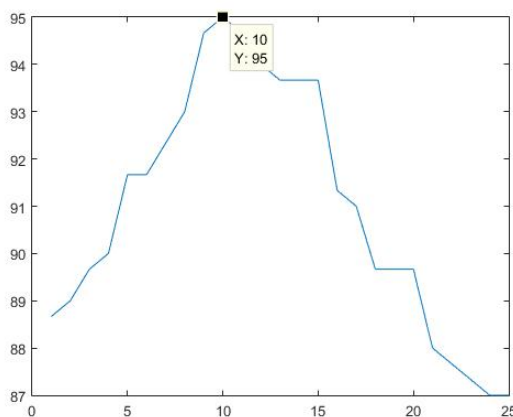
- Resultados globales del estudio completo de números.
- Resultados particularizados al mejor caso dentro de los números.
- Resultados globales del estudio completo de letras.
- Resultados particularizados al mejor caso dentro de letras.
-

8.1. DTW (Alineamiento temporal dinámico)

Para extraer la máxima eficacia de DTW se ha probado con distintos anchos de ventana de adaptación, visto en apartado 7. La variación ha sido llevada a cabo entre 1 y 25 muestras dentro de los vectores completos de 100 muestras de cada patrón.

8.1.1. Resultados estudio de números (aceleración)

Teniendo en cuenta que el número de patrones sometidos a DTW con el patrón medio de cada dígito es 300, el número de aciertos marcara la eficacia del método DTW en cada caso. En la siguiente grafica se puede observar cómo evoluciona la eficacia en función de la ventana.



En la imagen a la izquierda se puede observar como el máximo ratio de aciertos varía entre el 87 y el 95% alcanzando este máximo con una ventana temporal de 10 muestras.

El acierto medio se sitúa en un 91%, con todos estos datos seleccionamos como mejor caso el de ventana temporal de 10 muestras y con el hacemos un análisis más pormenorizado.

Fig. 14. Evolución de aciertos en función de ventana de adaptación temporal tomando la aceleración como patrón.

Para observar la efectividad del sistema de una manera rápido se ha organizado la Tabla 1 en escala de colores los valores de salida de cada DTW realizado. De manera que los que menor distancia guardan entre ellos se ven en colores más verdosos y los que mayor distancia guardan se ven en colores rojizos. Cada casilla representa un la comparación de un patrón con

uno de los medios, de manera que cada columna corresponde a la salida de un patrón con todos los medios de 0 a 9 y cada fila la de un patrón medio con los 300 patrones estudiados.

Tabla 1. Validación Cruzada DTW Números Aceleración

		Patrones Individuales									
		0	1	2	3	4	5	6	7	8	9
Patrones Medios	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Como se puede observar en la tabla, los números entre los que se producen menores distancias sin corresponder a un valor correcto son el 5 con el 6 y el 6 con el 9. Además, los que más diversidad tienen entre ellos son los cuatros.

Seleccionando el mínimo de los valores como el válido para la Decisión obtendremos el siguiente gráfico escalado, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

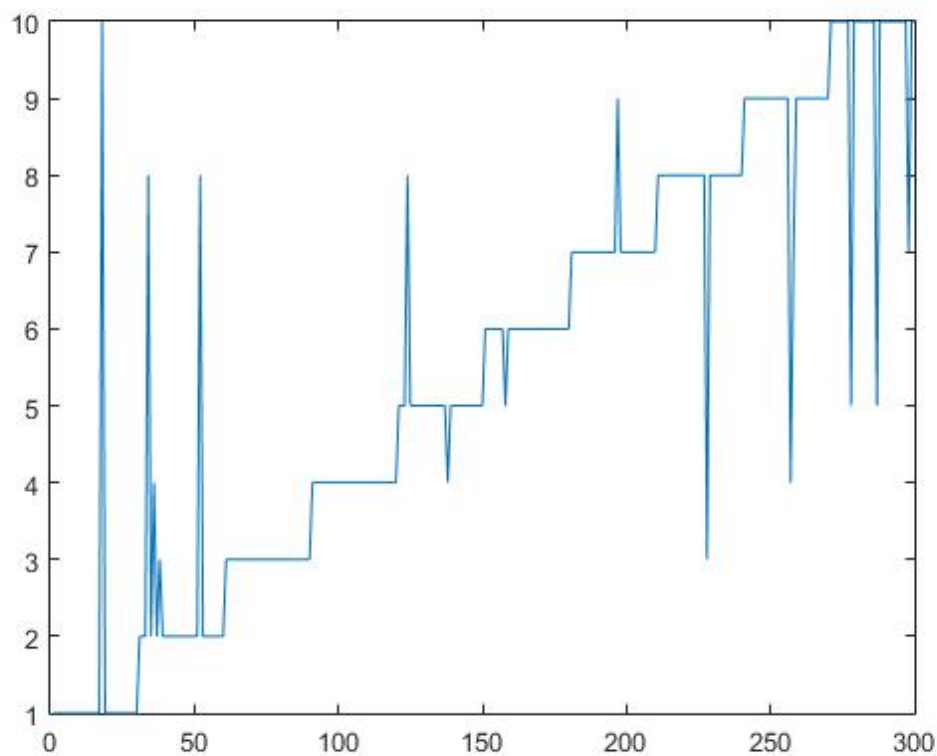


Fig. 15. Decisión DTW Números Aceleración

8.1.2. Resultados estudio de números (trayectoria)

En este apartado se resumen los resultados de un estudio similar al anterior, pero en este caso las señales de origen no serán las de aceleración, sino las de las trayectorias. Así, se repite el formato anterior.

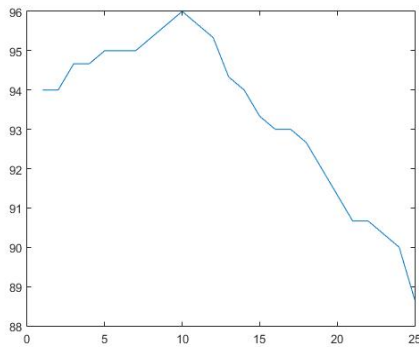


Fig. 16. Evolución de aciertos en función de ventana de adaptación temporal tomando como patrón las trayectorias.

En la imagen a la izquierda se puede observar como el máximo ratio de aciertos varía entre el 87 y el 96% alcanzando este máximo con una ventana temporal de 10 muestras.

El acierto medio se sitúa en un 93.37%, con todos estos datos, seleccionamos como mejor caso el de ventana temporal de 10 muestras y con el hacemos un análisis más pormenorizado.

Se organiza una tabla de manera similar al anterior caso.

Tabla 2. Validación Cruzada DTW Números Trayectorias

		Patrones Individuales									
		0	1	2	3	4	5	6	7	8	9
Patrones Medios	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Como se puede observar en la tabla, los números entre los que se producen menores distancias erróneamente siguen siendo el 5 con el 6 y pero no el 6 con el 9. Además, los que más diversidad tienen pasan a ser los ochos en vez de los cuatros.

Seleccionando el mínimo de los valores como el válido para la Decisión obtendremos el siguiente gráfico escalár, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

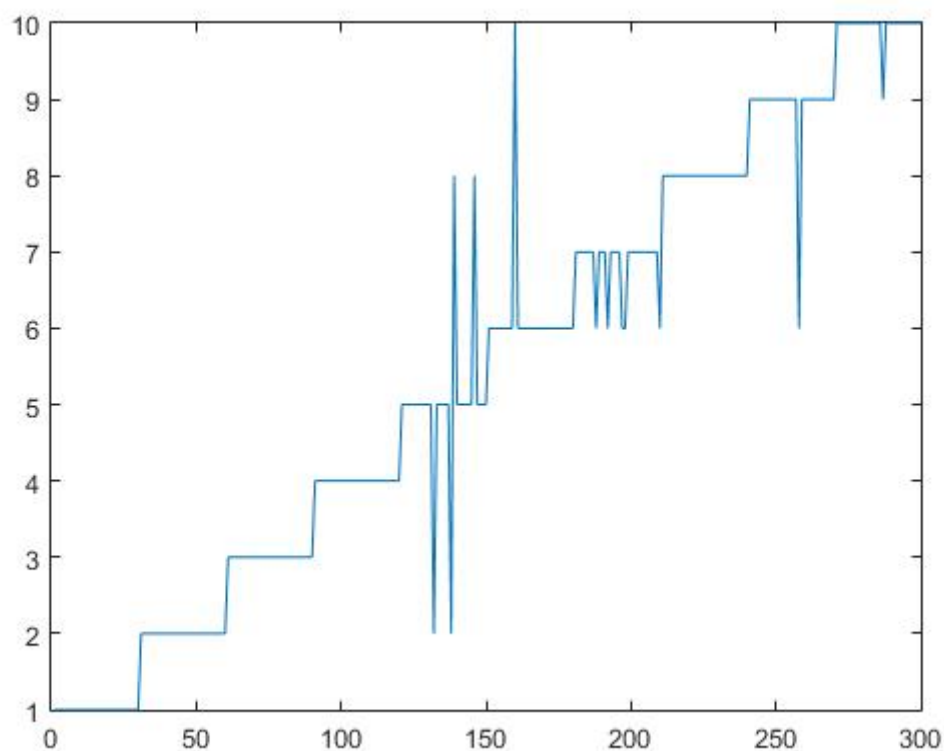


Fig. 17. Decisión DTW Números Trayectorias

8.1.3. Resultados estudio de números (aceleración + trayectorias)

En vista de los resultados observados se comprueba la unión de las dos opciones, simplemente sumando ambos resultados. Así, sumando los resultados con una ventana de 10 los aciertos alcanzan la cifra del 96.66%.

Se organiza una tabla de manera similar al los casos anteriores.

Tabla 3. Validación Cruzada DTW Números Aceleración + Trayectoria

		Patrones Individuales									
		0	1	2	3	4	5	6	7	8	9
Patrones Medios	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

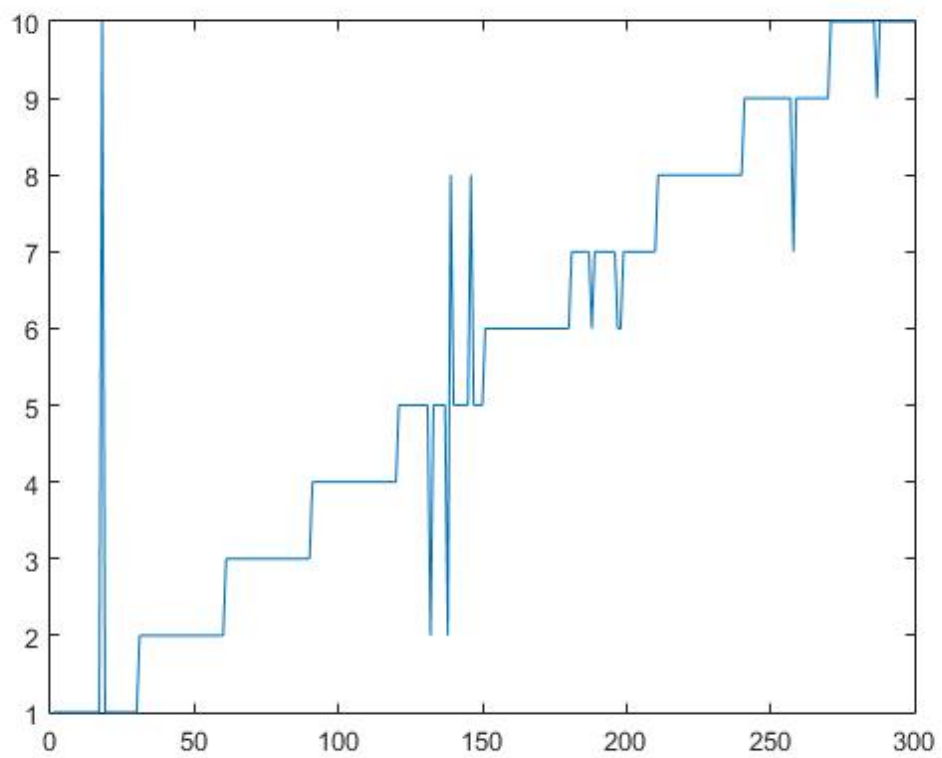
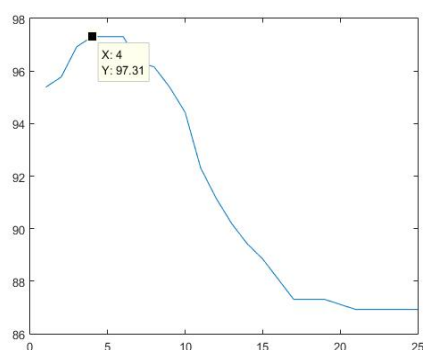


Fig. 18. Decisión DTW Números Aceleración + Trayectoria

Una vez vistos los resultados en referencia a los números, toca observar los resultados referentes a las letras.

8.1.4. Resultados estudio de letras (aceleración)

Teniendo en cuenta que el número de patrones sometidos a DTW con el patrón medio de cada dígito es 520 (26x20), el número de aciertos marcara la eficacia del método DTW en cada caso. En la siguiente grafica se puede observar cómo evoluciona la eficacia en función de la ventana.



En la imagen a la izquierda se puede observar como el máximo ratio de aciertos varía entre el 87 y el 97% alcanzando este máximo con una ventana temporal de 4 muestras.

El acierto medio se sitúa en un 91.5%, con todos estos datos seleccionamos como mejor caso el de ventana temporal de 4 muestras y con el hacemos un análisis más pormenorizado.

Fig. 19. Evolución de aciertos en función de ventana de adaptación temporal tomando la aceleración como patrón.

Para observar la efectividad del sistema de una manera rápido se ha organizado la Tabla 4 en escala de colores los valores de salida de cada DTW realizado. De manera que los que menor distancia guardan entre ellos se ven en colores más verdosos y los que mayor distancia guardan se ven en colores rojizos. Cada casilla representa un la comparación de un patrón con uno de los medios, de manera que cada columna corresponde a la salida de un patrón con todos los medios de a a z y cada fila la de un patrón medio con los 520 patrones estudiados.

Tabla 4. Validación Cruzada DTW Letras Aceleración

	Patrones Individuales																									
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Patrones Medios	a																									
	b																									
	c																									
	d																									
	e																									
	f																									
	g																									
	h																									
	i																									
	j																									
	k																									
	l																									
	m																									
	n																									
	o																									
	p																									
	q																									
	r																									
	s																									
	t																									
	u																									
	v																									
	w																									
	x																									
	y																									
	z																									

Como se puede observar en la tabla, hay varias letras que se confunden con otras. En cambio, al contrario que en los números, no se encuentra una gran diversidad entre patrones únicos.

Seleccionando el mínimo de los valores como el válido para la Decisión obtendremos el siguiente gráfico escalonado, donde los puntos que se salen fuera de la diagonal serán los fallos introducidos.

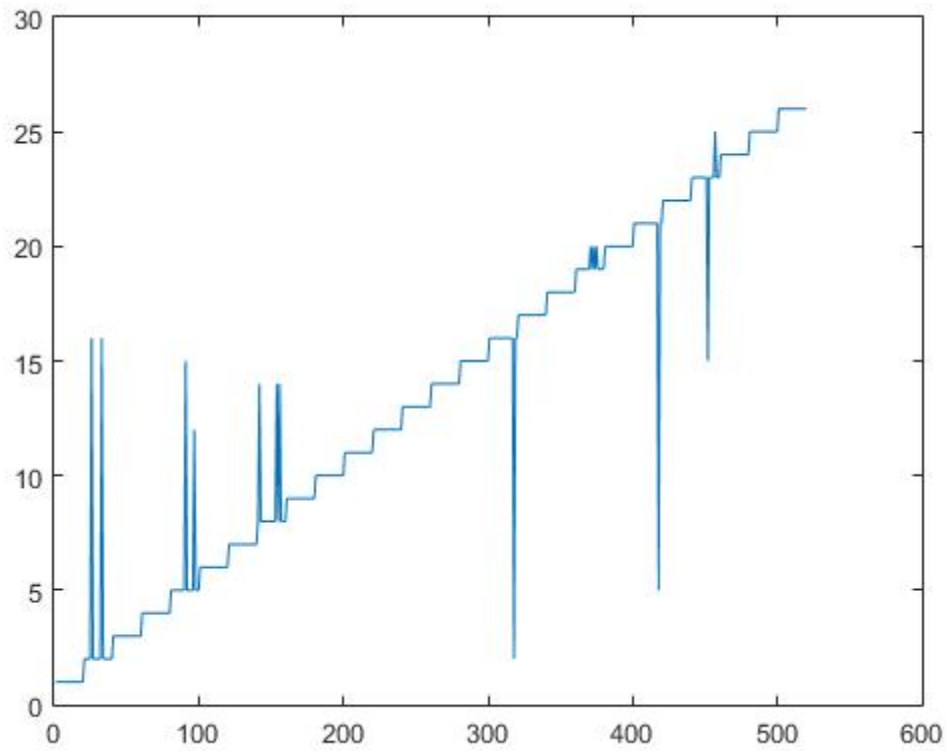


Fig. 20. Decisión DTW Letras Aceleración

8.1.5. Resultados estudio de letras (trayectoria)

En este apartado se resumen los resultados de un estudio similar al anterior, pero en este caso las señales de origen no serán las de aceleración, sino las de las trayectorias. Así, se repite el formato anterior.

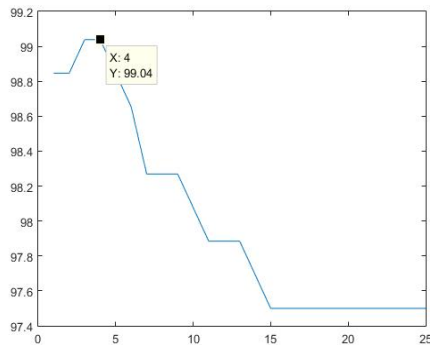


Fig. 21. Evolución de aciertos en función de ventana de adaptación temporal tomando como patrón las trayectorias.

En la imagen a la izquierda se puede observar como el máximo ratio de aciertos varía entre el 97 y el 99% alcanzando este máximo con una ventana temporal de 10 muestras.

El acierto medio se sitúa en un 98%, con todos estos datos, seleccionamos como mejor caso el de ventana temporal de 4 muestras y con el hacemos un análisis más pormenorizado.

Se organiza una tabla de manera similar al anterior caso.

Tabla 5. Validación Cruzada DTW Letras Trayectorias

	Patrones Individuales																									
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a																										
b																										
c																										
d																										
e																										
f																										
g																										
h																										
i																										
j																										
k																										
l																										
m																										
n																										
o																										
p																										
q																										
r																										
s																										
t																										
u																										
v																										
w																										
x																										
y																										
z																										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

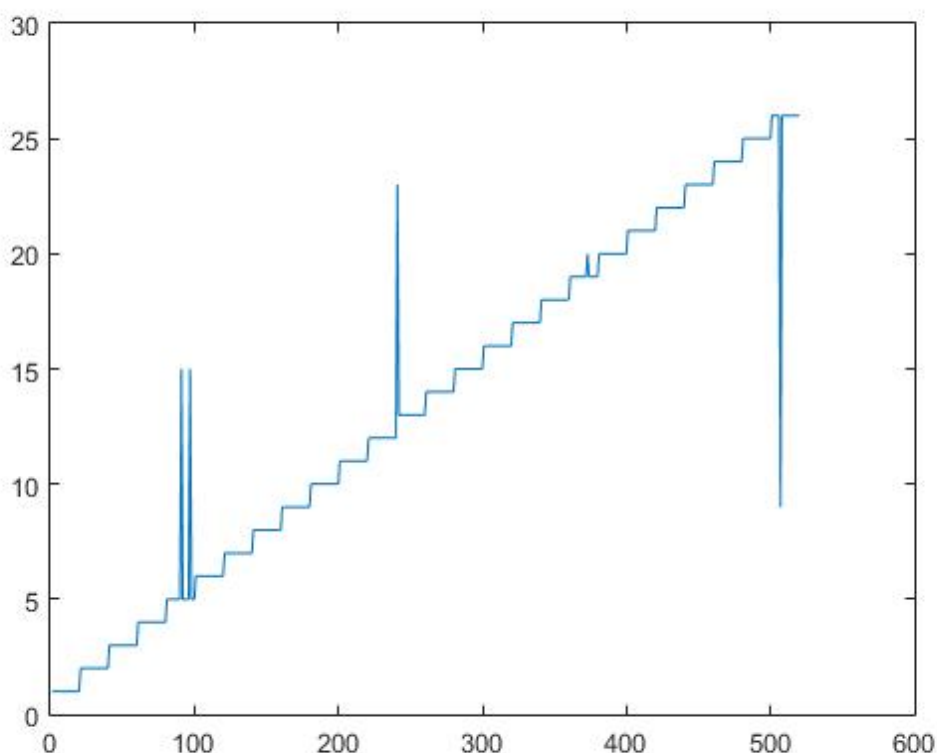


Fig. 22. Decisión DTW Letras Trayectorias

8.1.6. Resultados del estudio de letras (aceleración + trayectorias)

En vista de lo sucedido en el entorno de los números, se comprueba también si la mejoría se extiende a las letras al sumar los resultados producidos con la aceleración más trayectorias. Así, se observa como la eficacia se aumenta hasta el 99,3%.

Tabla 6. Validación Cruzada DTW Letras Aceleración + Trayectorias

	Patrones Individuales																									
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a																										
b																										
c																										
d																										
e																										
f																										
g																										
h																										
i																										
j																										
k																										
l																										
m																										
n																										
o																										
p																										
q																										
r																										
s																										
t																										
u																										
v																										
w																										
x																										
y																										
z																										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalár, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

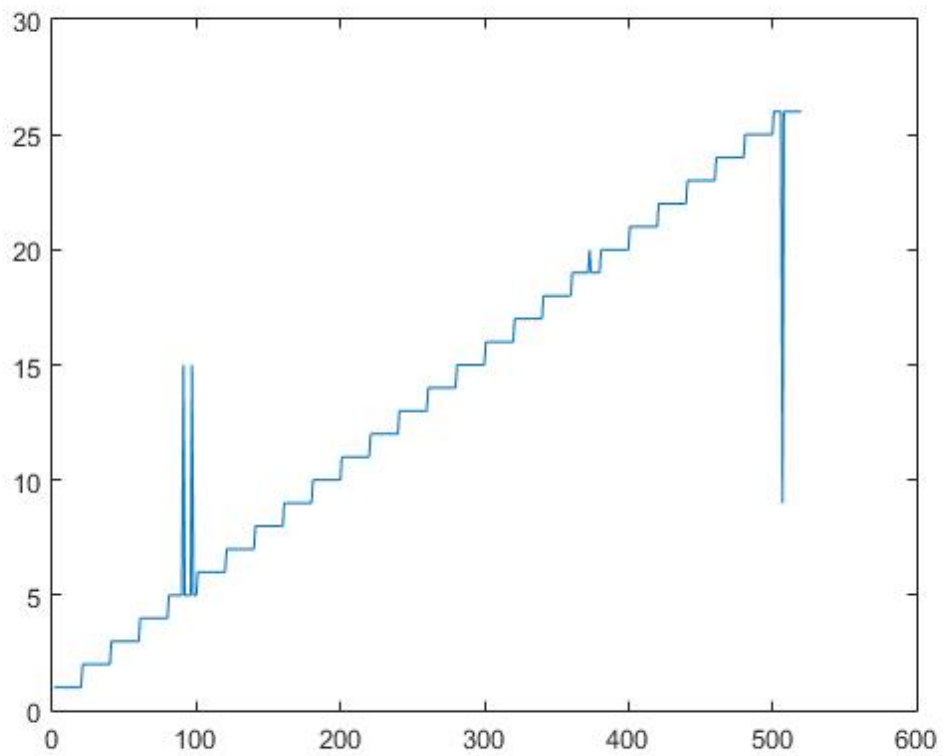


Fig. 23. Decisión DTW Letras Trayectorias+Aceleración

8.2. HMM (Modelos Ocultos de Markov)

El estudio se ha realizado en función del número de estados y del número de observaciones, así se puede obtener un análisis sobre qué tamaño de HMM es el más adecuado para nuestro trabajo.

Además, en línea con lo realizado en el apartado DTW se harán cuatro estudios diferenciados; números y letras con aceleración y trayectoria.

8.2.1. Resultados estudio de números (aceleración)

Teniendo en cuenta que el número de patrones (números) de test sometidos a estudio con los modelos de Markov entrenados con los patrones de entrenamiento de cada dígito es 120, el número de aciertos marcará la eficacia del método HMM en cada caso. En la siguiente gráfica se puede observar cómo evoluciona la eficacia en función de la de los estados ocultos y los observables.

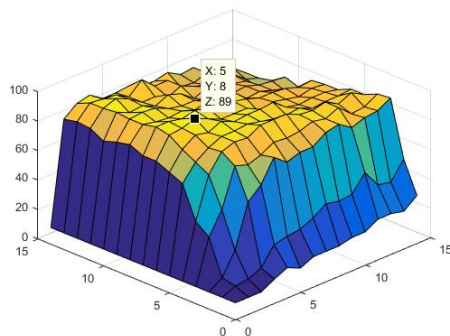


Fig. 24. Evolución de aciertos en función de ventana de Adaptación temporal tomando la aceleración como patrón.

En la imagen a la izquierda se puede observar cómo evoluciona la eficacia en función de la de los estados ocultos y los observables. En la imagen a la izquierda se puede observar cómo evoluciona la eficacia en función de la de los estados ocultos y los observables. En la imagen a la izquierda se puede observar cómo evoluciona la eficacia en función de la de los estados ocultos y los observables. En la imagen a la izquierda se puede observar cómo evoluciona la eficacia en función de la de los estados ocultos y los observables.

Para observar la efectividad del sistema de una manera rápida se ha organizado la Tabla bajo estas líneas, en escala de colores se puede observar la validación cruzada de los patrones de test con los modelos ocultos de 11 estados ocultos y 4 de observación. De manera que los más probables se observan en colores verdosos y los menos probables en rojo.

Tabla 7. Validación Cruzada HMM Números Aceleración

		Patrones Individuales									
		0	1	2	3	4	5	6	7	8	9
Patrones Medios	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Seleccionando el máximo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

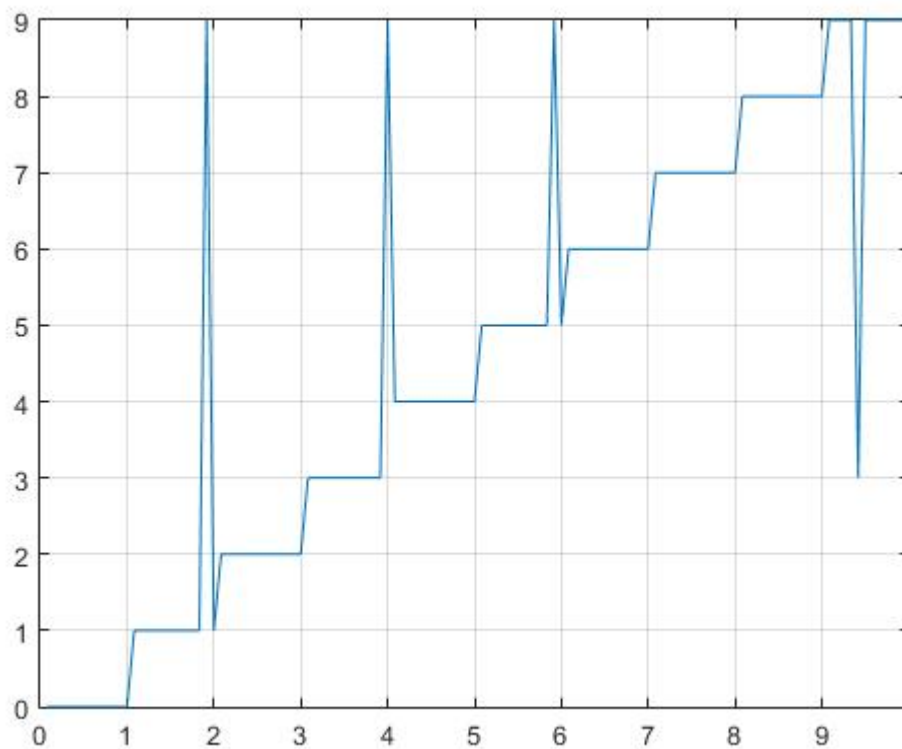
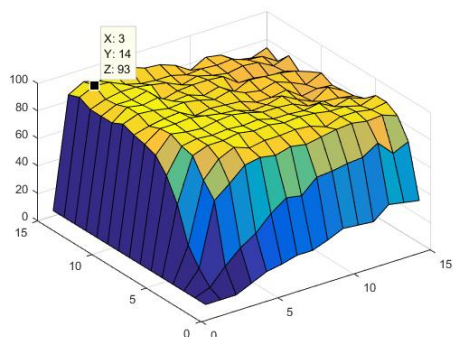


Fig. 25. Decisión HMM Números Aceleración

8.2.2. Resultados estudio de números (trayectoria)

En este caso, cumpliendo los mismos parámetros del anterior, podemos observar unos resultados parecidos.



En la imagen a la izquierda se puede observar como el máximo número de aciertos se sitúa en 115, 95.8 %. Este se alcanza entrenando un modelo de Markov de 13 estados ocultos y 6 de observación.

Fig. 26. Evolución de aciertos en función de ventana de adaptación temporal tomando como patrón las trayectorias.

Se organiza una tabla de manera similar al anterior caso pero con 13 estados ocultos y 6 de observación.

Tabla 8. Validación Cruzada HMM Números Trayectorias

		Patrones Individuales									
		0	1	2	3	4	5	6	7	8	9
Patrones Medios	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

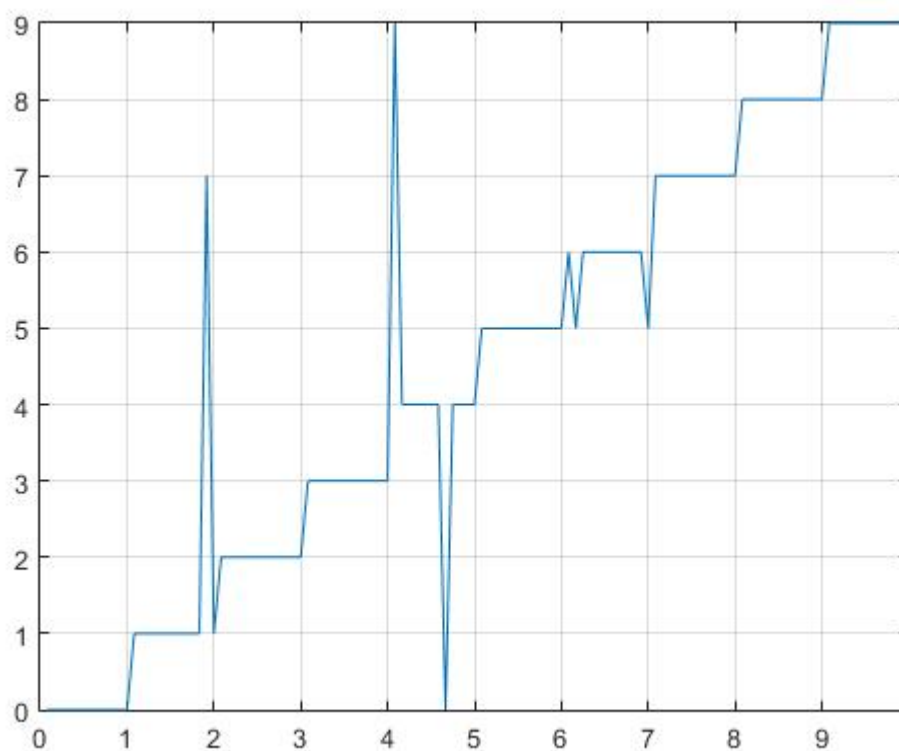


Fig. 27. Decisión HMM Números Trayectorias

8.2.3. Resultados estudio de números (aceleración + trayectoria)

En vista de los resultados vistos se prueba la unión de las dos opciones, simplemente sumando ambos resultados. Así, sumando los resultados anteriores se alcanzan unos resultados del 96,6 %. Esto significa que no mejora el mejor de los dos resultados anteriores.

Se organiza una tabla de manera similar al los casos anteriores.

Tabla 9. Validación Cruzada HMM Números Aceleración + Trayectoria

		Patrones Individuales									
		0	1	2	3	4	5	6	7	8	9
Patrones Medios	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

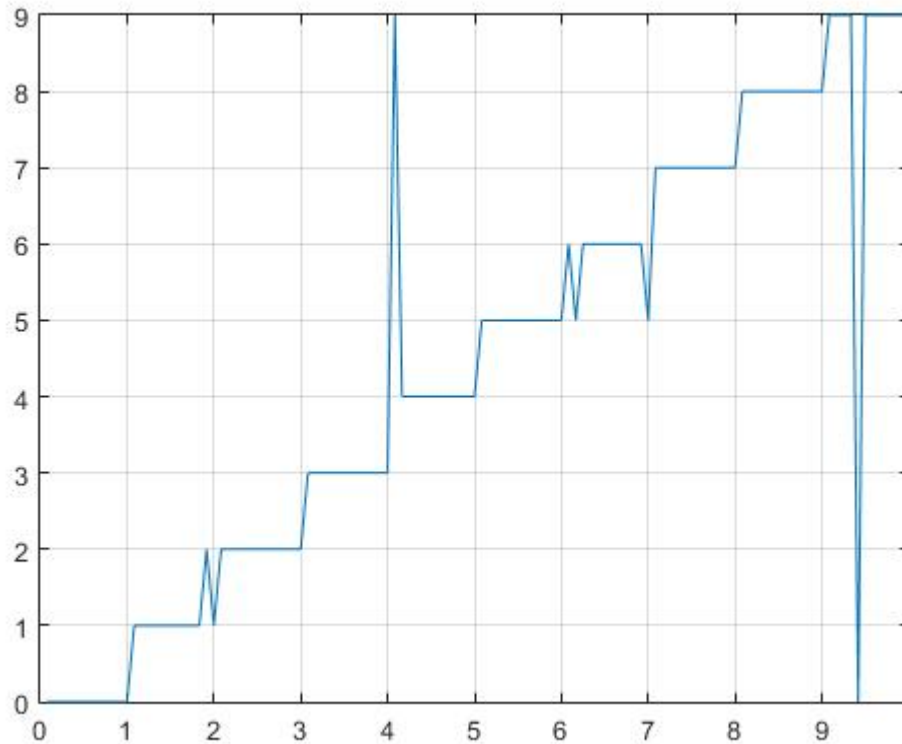


Fig. 28. Decisión HMM Números con Aceleración Trayectoria

8.2.4. Resultados estudio de letras (aceleración)

Teniendo en cuenta que el número de patrones (números) de test sometidos a estudio con los modelos de Markov entrenados con los patrones de entrenamiento de cada dígito es 208 (8x26), el número de aciertos marcará la eficacia del método HMM en cada caso. En la siguiente gráfica se puede observar cómo evoluciona la eficacia en función de la de los estados ocultos y los observables.

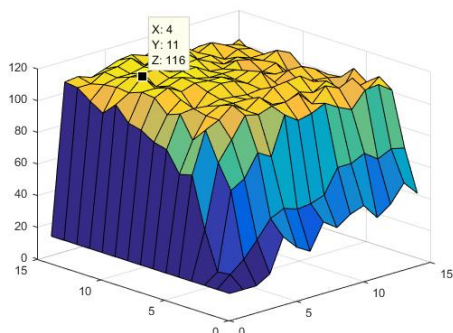


Fig. 29. Evolución de aciertos en función de ventana de adaptación temporal tomando la aceleración como patrón.

En la imagen a la izquierda se puede observar como el máximo número de aciertos se sitúa en 116, 96.6 %. Este se alcanza entrenando un modelo de Markov de 8 estados ocultos y 5 de observación.

Para observar la efectividad del sistema de una manera rápida se ha organizado la Tabla 10, en escala de colores se puede observar la validación cruzada de los patrones de test con los modelos ocultos de 8 estados ocultos y 5 de observación. De manera que los más probables se observan en colores verdosos y los menos probables en rojo.

Tabla 10. Validación Cruzada HMM Letras Aceleración

		Patrones Individuales																													
		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z				
Patrones Medios	a																														
	b																														
	c																														
	d																														
	e																														
	f																														
	g																														
	h																														
	y																														
	j																														
	k																														
	l																														
	m																														
	n																														
	o																														
	p																														
	q																														
	r																														
	s																														
	t																														
	u																														
	v																														
	w																														
	x																														
	y																														
	z																														

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde los puntos que se salen fuera de la diagonal serán los fallos introducidos.

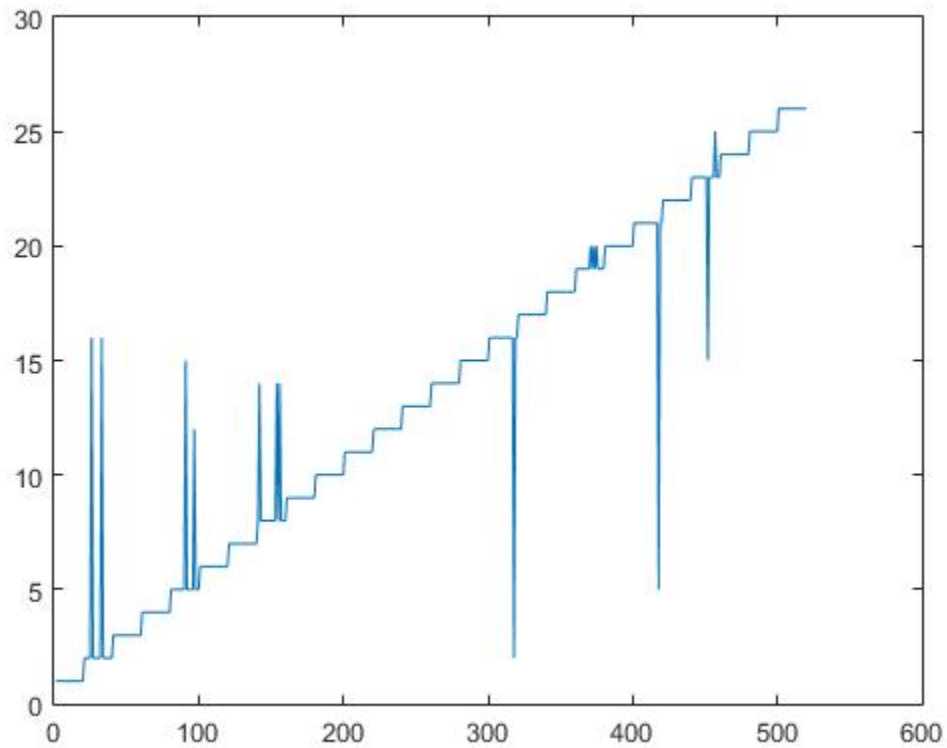
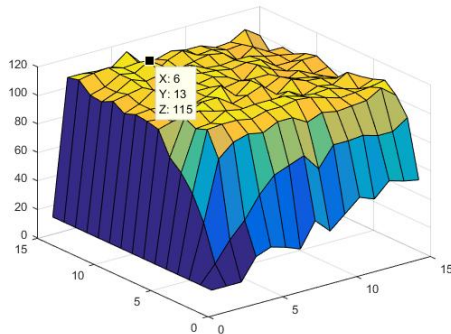


Fig. 30. Decisión HMM Letras Aceleración

8.2.5. Resultados estudio de letras (trayectoria)

Siguiendo el modelo de los casos anteriores podemos ver la eficacia de HMM en función de estados ocultos y observaciones.



En la imagen a la izquierda se puede observar como el máximo número de aciertos se sitúa en 194, 95.8 %. Este se alcanza entrenando un modelo de Markov de 13 estados ocultos y 6 de observación.

Fig. 31. Evolución de aciertos en función de ventana de adaptación temporal tomando la aceleración como patrón.

Para observar la efectividad del sistema de una manera rápido se ha organizado la tabla bajo estas líneas en escala de colores la validación cruzada de las salidas de cada HMM realizado. De manera que los que menor distancia guardan entre ellos se ven en colores más verdosos y los que mayor distancia guardan se ven en colores rojizos. Cada casilla representa la comparación de un patrón con uno de los medios, de manera que cada columna corresponde a la salida de un patrón con todos los medios de a a z y cada fila la de un patrón medio con los 520 patrones estudiados.

Tabla 11. Validación Cruzada HMM Letras Trayectorias

		Patrones Individuales																									
		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Patrones Medios	a																										
	b																										
	c																										
	d																										
	e																										
	f																										
	g																										
	h																										
	y																										
	j																										
	k																										
	l																										
	m																										
	n																										
	o																										
	p																										
	q																										
	r																										
	s																										
	t																										
	u																										
	v																										
	w																										
	x																										
	y																										
	z																										

Como se puede observar en la tabla, hay varias letras que se confunden con otras. En cambio, al contrario que en los números, no se encuentra una gran diversidad entre patrones únicos.

Seleccionando el mínimo de los valores como el válido para la Decisión obtendremos el siguiente gráfico escalar, donde los puntos que se salen fuera de la diagonal serán los fallos introducidos.

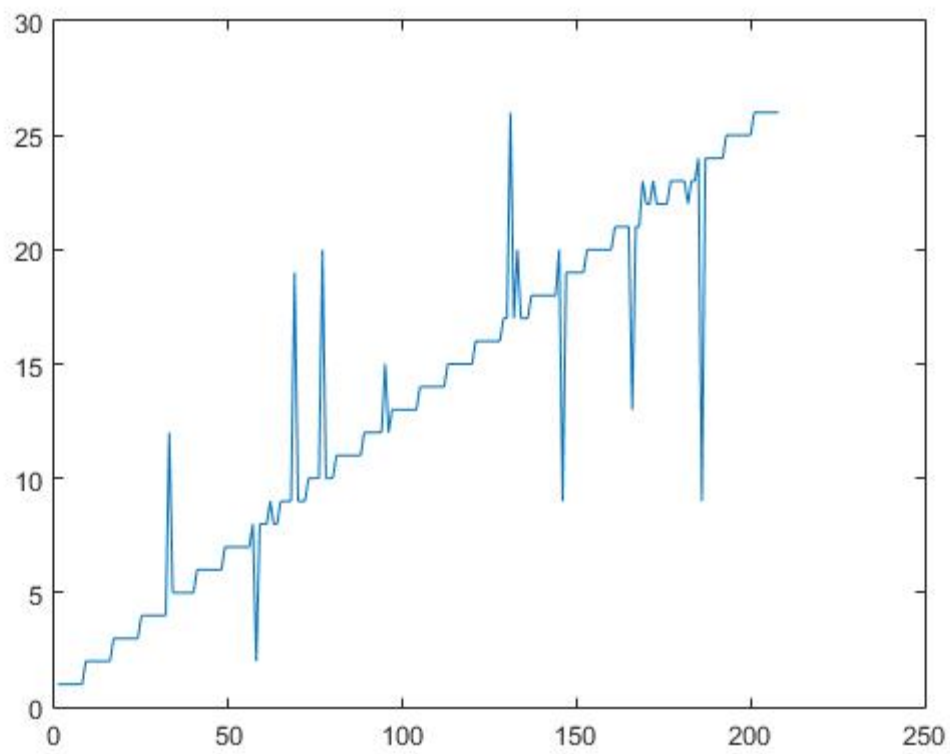


Fig. 32. Decisión HMM Letras Trayectorias

8.2.6. Resultados estudio de letras (aceleración + trayectoria)

En vista de los resultados vistos se prueba la unión de las dos opciones, simplemente sumando ambos resultados. Así, sumando los resultados con una ventana de 10 los aciertos alcanzan la cifra del 96.66%.

Se organiza una tabla de manera similar al los casos anteriores.

Tabla 12. Respuesta HMM Letras Aceleración Trayectoria

		Patrones Individuales									
		0	1	2	3	4	5	6	7	8	9
Patrones Medios	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

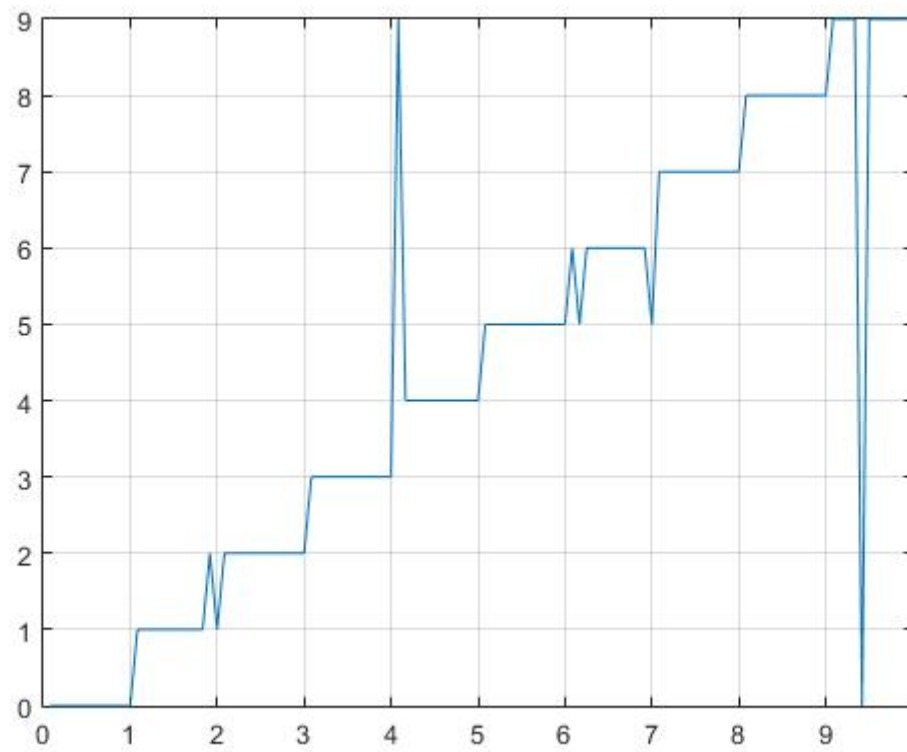


Fig. 33. Decisión HMM Letras Aceleración Trayectoria

8.3. NN (Redes Neuronales Aplicadas al Reconocimiento de Patrones)

El estudio se ha realizado en función del número de estados y del número de observaciones, así se puede obtener un análisis sobre que tamaño de HMM es el más adecuado para nuestro trabajo.

Además, en línea con lo realizado en los apartados anteriores se harán cuatro estudios diferenciados; números y letras con aceleración y trayectoria.

Teniendo en cuenta que el número con los que es llevado a cabo el test de las redes neuronales, en el caso de los números, es de 12, el total de patrones testeados es 120. La eficacia del sistema la calcularemos dividiendo el número de aciertos entre los patrones de test, aciertos/120.

Mientras en el caso de las letras, son 8 por letra, $8 \times 26 = 208$. Calculando la eficacia de la red dividiendo el número de aciertos entre estos 208.

8.3.1. Resultados estudio de números (aceleración)

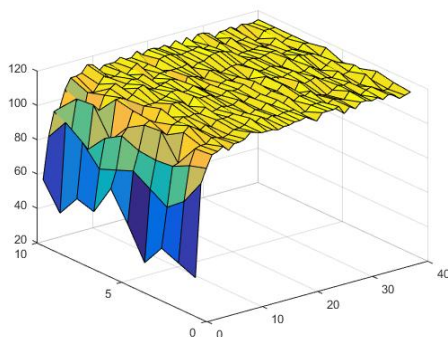


Fig. 34. Evolución de aciertos en función de ventana de número de muestras de entrada y neuronas en la capa oculta.

En la imagen a la izquierda se puede observar como el número de aciertos de la red depende, en gran medida del número de neuronas en la capa oculta, principalmente debido a que el aumento de patrones de entrada tan solo depende de la definición que queremos de la señal. Así, al añadir patrones de entrada no añadimos información fundamental.

Por ello se han seleccionado redes con entradas de 50 muestras. Y en este caso la eficacia es de $116/120 = 96.6\%$.

Para observar la efectividad del sistema de una manera rápida se muestra la validación cruzada de los resultados en la tabla bajo estas líneas.

Tabla 13. Validación Cruzada NN Números Aceleración

		Patrones Individuales							
		0	1	2	3	4	5	6	7
Patrones Medios	0								
	1								
	2								
	3								
	4								
	5								
	6								

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

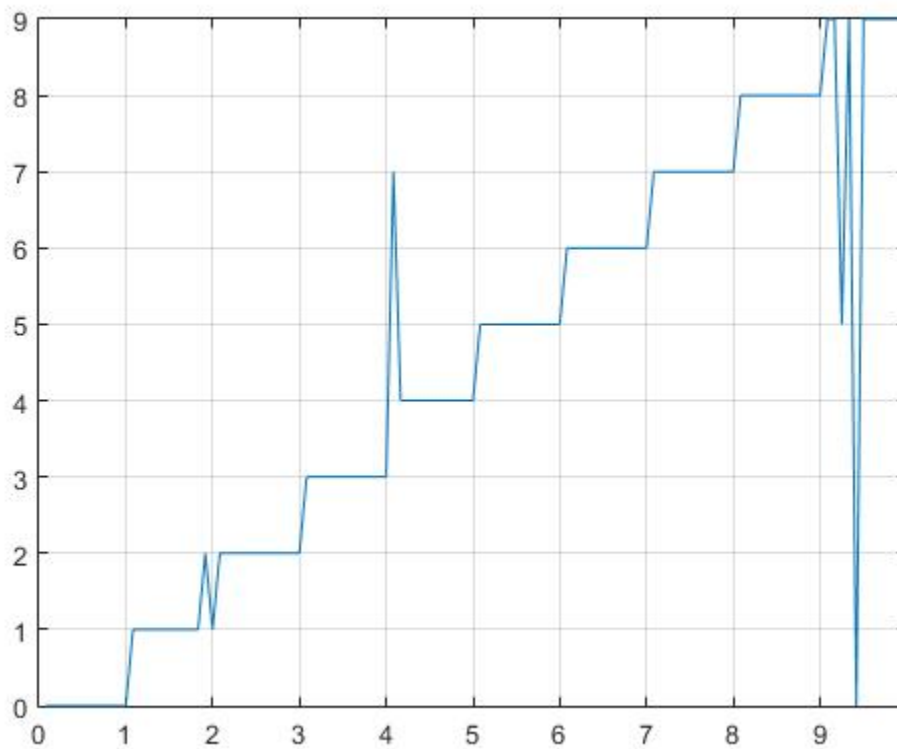
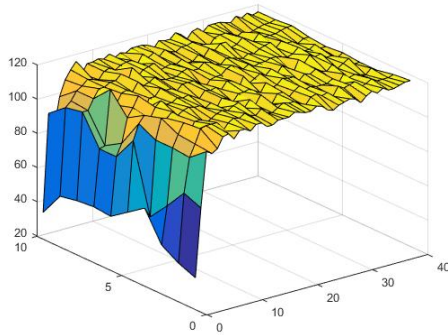


Fig. 35. Decisión NN Números con Aceleración

8.3.2. Resultados estudio de números (trayectoria)



Eficacia es de $116/120=96.6\%$.

Fig. 36. Evolución de aciertos en función de ventana de número de muestras de entrada y neuronas en la capa oculta

Se organiza una tabla de manera similar al anterior caso.

Tabla 14. Validación Cruzada NN Números Trayectorias

		Patrones Individuales									
		0	1	2	3	4	5	6	7	8	9
Patrones Medios	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

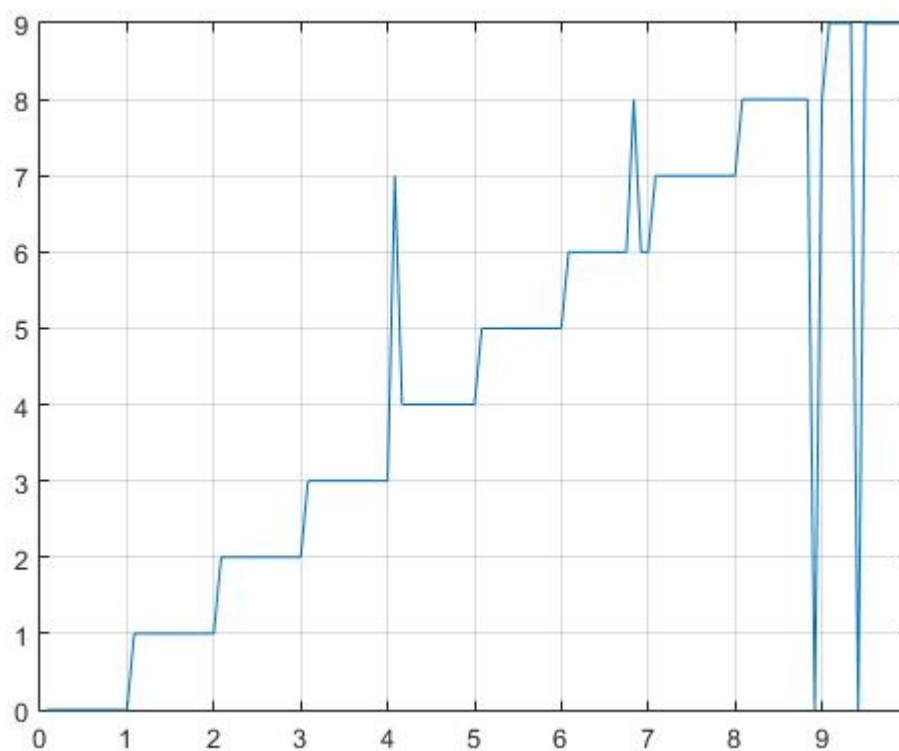


Fig. 37. Decisión DTW Números Trayectorias

8.3.3. Resultados estudio de números (aceleración + trayectoria)

En vista de los resultados vistos se prueba la unión de las dos opciones, simplemente sumando ambos resultados. Así, sumando los dos resultados anteriores se obtiene un acierto del 97.5.

Tabla 15. Validación Cruzada NN Números Aceleración Trayectoria

		Patrones Individuales									
		0	1	2	3	4	5	6	7	8	9
Patrones Medios	0										
	1										
	2										
	3										
	4										
	5										
	6										
	7										
	8										
	9										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde las líneas que se salen fuera de la diagonal serán los fallos introducidos.

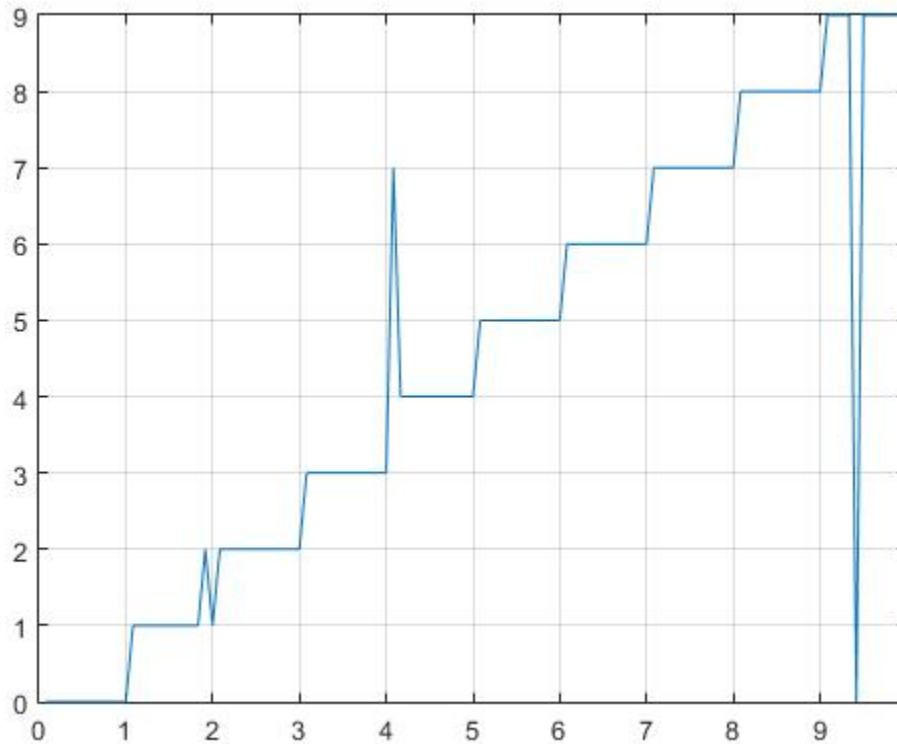
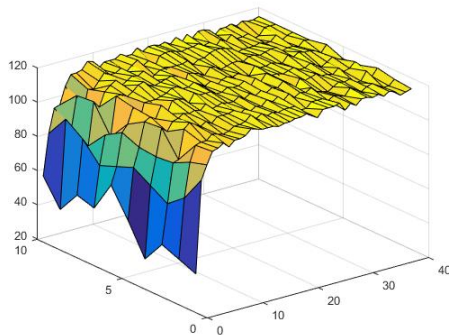


Fig. 38. Decisión NN Números Trayectorias

8.3.4. Resultados estudio de letras (aceleración)



Eficacia es de $196/208=94.6\%$.

**Fig. 39. Evolución de aciertos
en función de ventana de numero
de muestras de entrada y neuronas en
la capa oculta**

Bajo estas líneas se puede observar la tabla de validación cruzada de las respuestas de las redes neuronales.

Tabla 16. Respuesta NN Números Aceleración

		Patrones Individuales																										
		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	
Patrones Medios	a																											
	b																											
	c																											
	d																											
	e																											
	f																											
	g																											
	h																											
	y																											
	j																											
	k																											
	l																											
	m																											
	n																											
	o																											
	p																											
	q																											
	r																											
	s																											
	t																											
	u																											
	v																											
	w																											
	x																											
	y																											
	z																											

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde los puntos que se salen fuera de la diagonal serán los fallos introducidos.

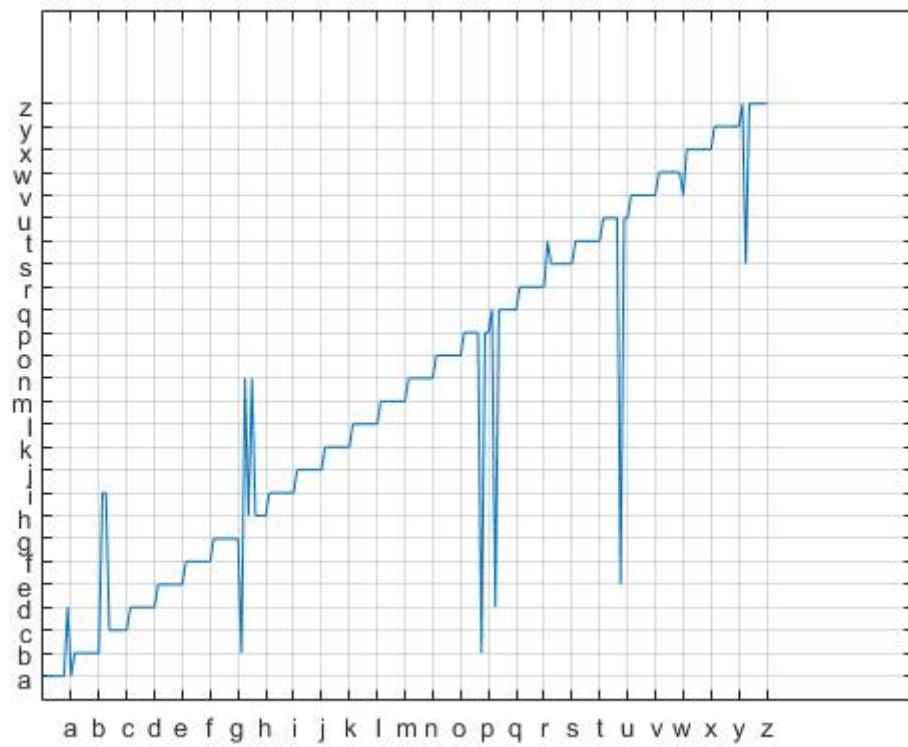


Fig. 40. Decisión NN Letras Aceleración

8.3.5. Resultados estudio de letras (trayectoria)

Eficacia es de $202/208=97.12\%$.

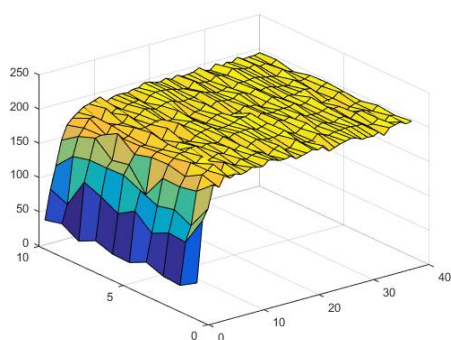


Fig. 41. Evolución de aciertos en función de ventana de número de muestras de entrada y neuronas en la capa oculta

Bajo estas líneas se puede observar la validación cruzada.

Tabla 17. Respuesta NN Letras Trayectorias

		Patrones Individuales																									
		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Patrones Medios	a																										
	b																										
	c																										
	d																										
	e																										
	f																										
	g																										
	h																										
	y																										
	j																										
	k																										
	l																										
	m																										
	n																										
	o																										
	p																										
	q																										
	r																										
	s																										
	t																										
	u																										
	v																										
	w																										
	x																										
	y																										
	z																										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde los puntos que se salen fuera de la diagonal serán los fallos introducidos.

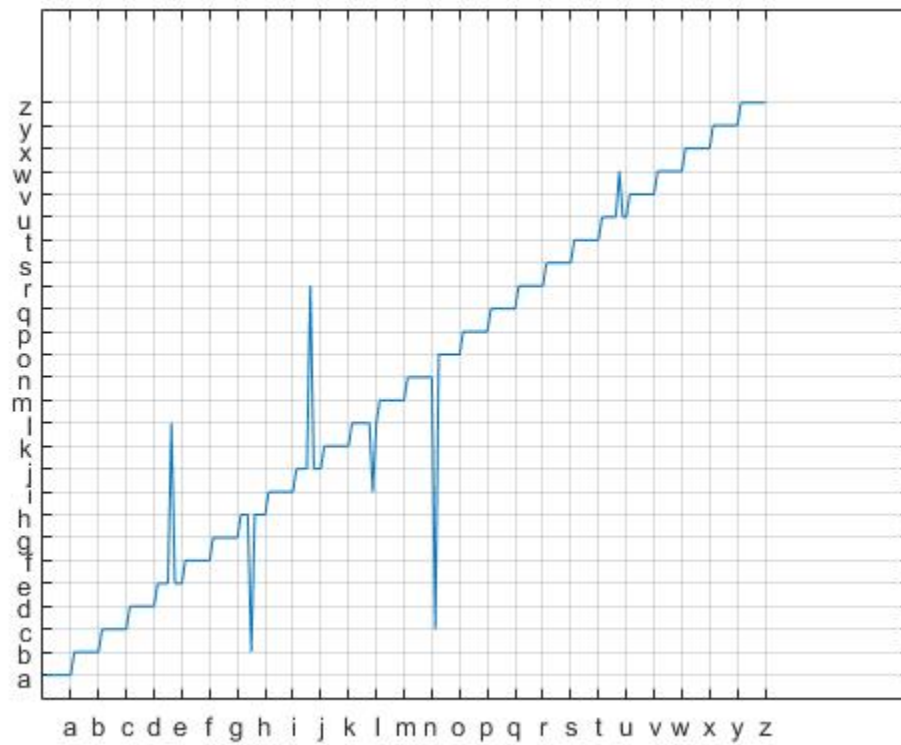


Fig. 42. Decisión NN Letras Aceleración

8.3.6. Resultados estudio de letras (trayectoria + aceleración)

Eficacia es de $202/208=97.12\%$.

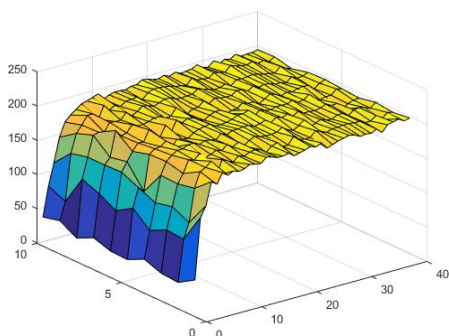


Fig. 43. Evolución de aciertos en función de ventana de numero de muestras de entrada y neuronas en la capa oculta

Bajo estas líneas se puede observar la validación cruzada.

Tabla 18. Respuesta NN Letras (Trayectorias + Aceleración)

		Patrones Individuales																									
		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Patrones Medios	a																										
	b																										
	c																										
	d																										
	e																										
	f																										
	g																										
	h																										
	y																										
	j																										
	k																										
	l																										
	m																										
	n																										
	o																										
	p																										
	q																										
	r																										
	s																										
	t																										
	u																										
	v																										
	w																										
	x																										
	y																										
	z																										

Seleccionando el mínimo de los valores como el valido para la Decisión obtendremos el siguiente grafico escalar, donde los puntos que se salen fuera de la diagonal serán los fallos introducidos.

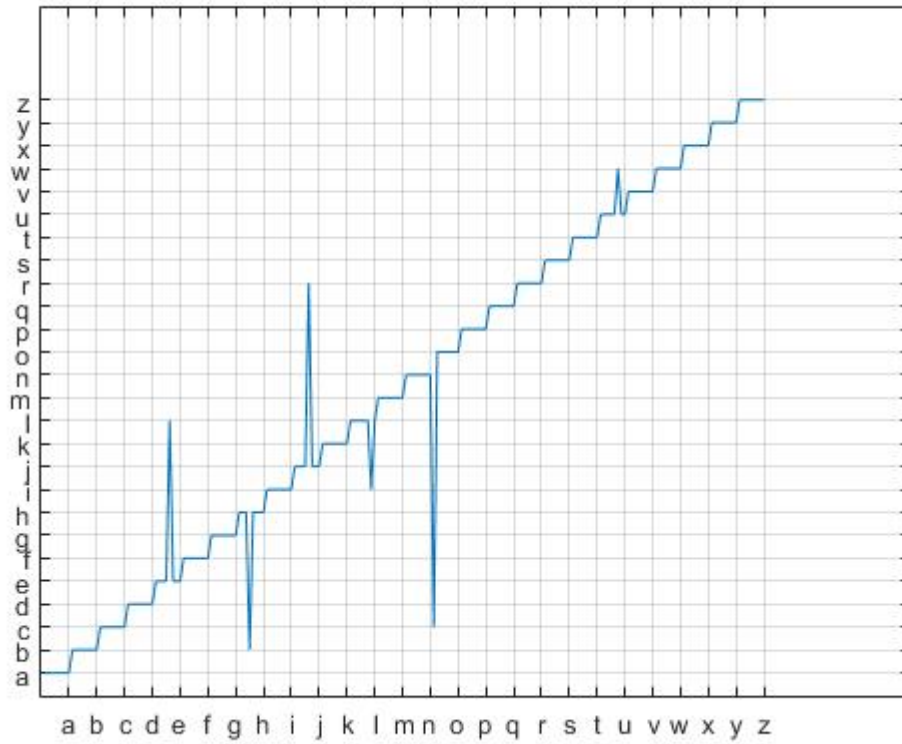


Fig. 44. Decisión NN Letras Aceleración

En este caso los resultados son exactamente iguales que en anterior, letras solo con trayectorias.

9. Conclusiones

Para finalizar, podemos concluir con un resumen de los resultados que se muestra en la siguiente tabla.

Tabla 19. Resumen Resultados

		DTW	HMM	NN
Números	Aceleración	95,0%	96,6%	96,6%
	Trayectorias	96,0%	95,8%	96,6%
	Ace + Tra	96,6%	96,6%	97,5%
Letras	Aceleración	97,0%	96,6%	94,6%
	Trayectorias	99,0%	95,8%	97,1%
	Ace + Tra	99,3%	96,7%	97,1%

De estos resultados se puede concluir que siendo las letras de nuestra base de datos, patrones con mayor regularidad el sistema de DTW se comporta con mayor precisión que los sistemas entrenados HMM y NN. Mientras, siendo los número patrones que en observación visual se presentan como más irregulares, mayor variabilidad, el sistema de NN es el más preciso de todos. En una extensión de la base de datos, las irregularidades son más presentes

Finalmente, y con la ambición de representar todo el trabajo realizado de manera visual. Se realiza una aplicación de PC, que pretende funcionar a modo de driver cargando los datos de un patrón y realizando los cálculos con la mejor selección de los procesos matemáticos evaluados.

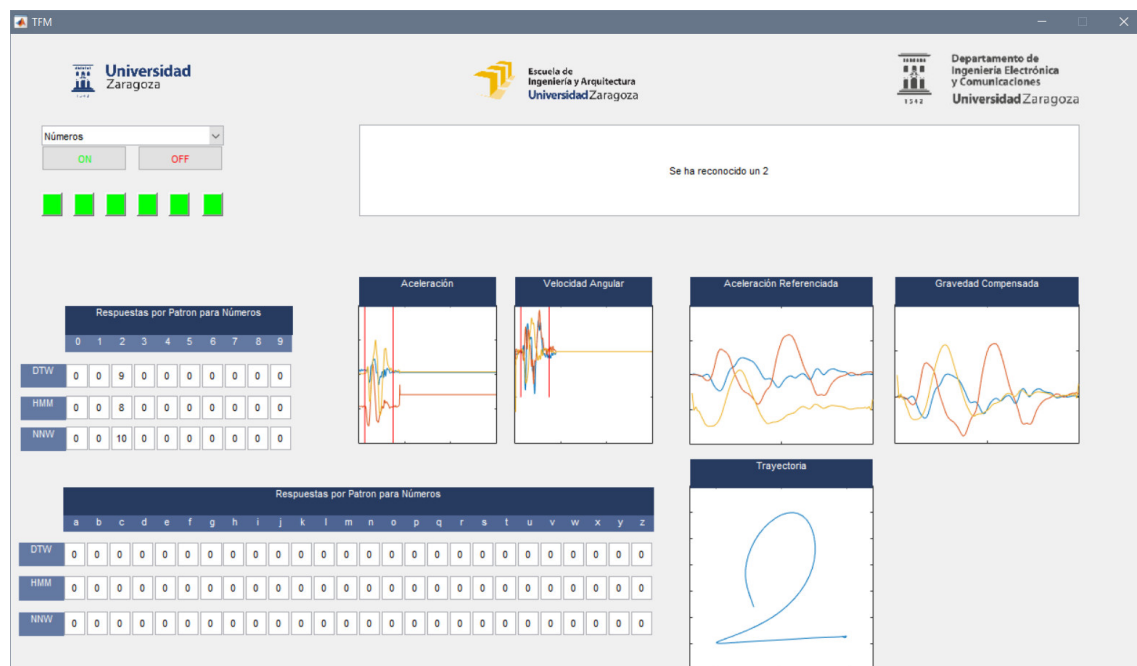


Fig. 45. Pantallazo de la Aplicación

10. Futuros Desarrollos

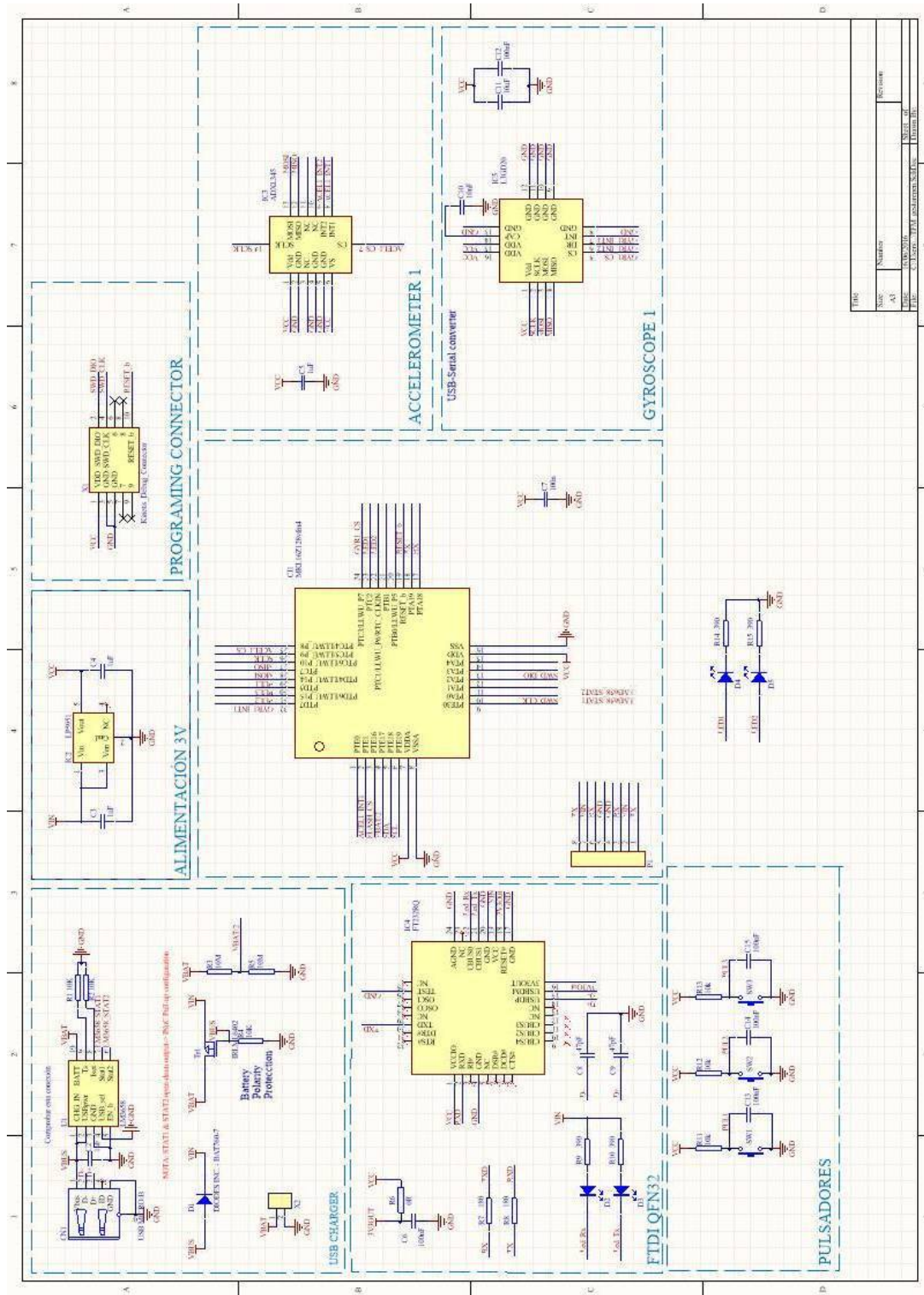
Actualmente el sistema de segmentación está basado en atravesar los umbrales de aceleración o velocidad angular absoluta, de manera que se hace necesario partir y terminar en reposo para cada carácter. Sería interesante pensar en el desarrollo de un sistema que fuese capaz de discernir el punto de inicio y final de cada carácter sin ser necesario el estado de reposo entre ellos.

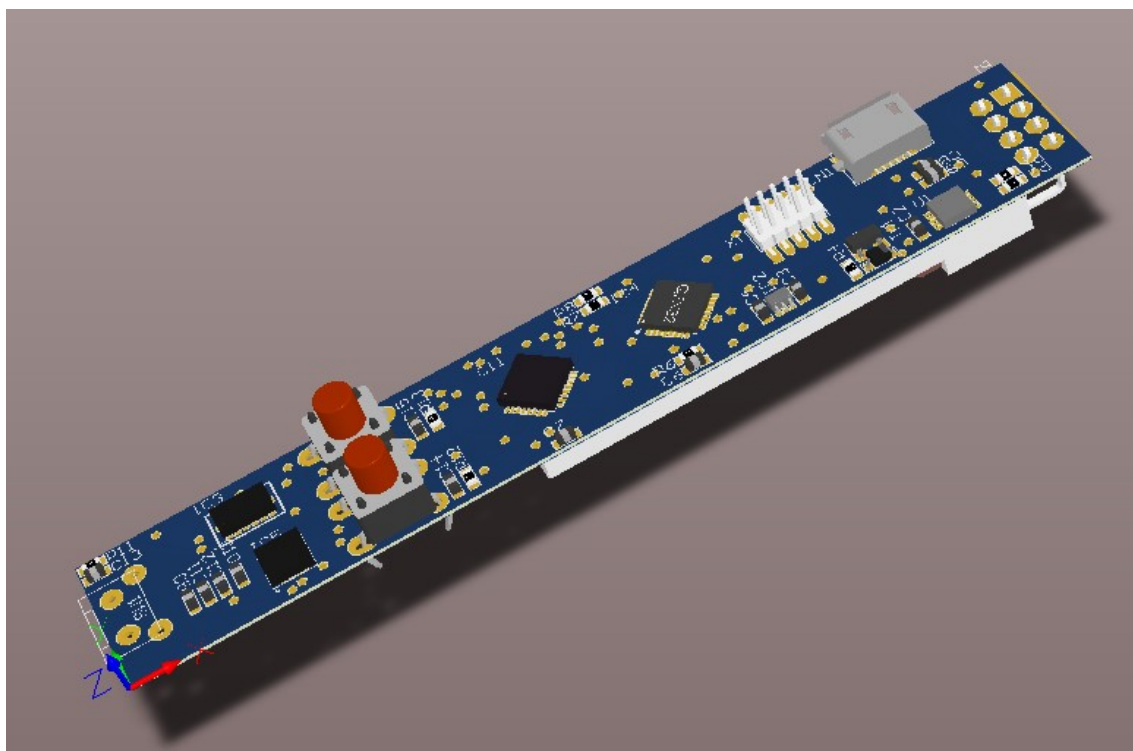
Este podría plantearse mediante redes neuronales evaluando las características de una señal continua en el tiempo en periodos de distintas longitudes temporales. Algunas de estas características básicas son, varianza, desviación estándar, valor medio, cruces por cero entre otras.

11. Bibliografía

- [1] Xie, R.; Cao, J. "Accelerometer-Based Hand Gesture Recognition by Neural Network and Similarity Matching", *Sensors Journal, IEEE*, On page(s): 4537 - 4545 Volume: 16, Issue: 11, June1, 2016
- [2] A. Akl, C. Feng and S. Valaee, "A novel accelerometer-based gesture recognition system", *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6197-6205, 2011
- [3] Y. Hsu, C. Chu, Y. Tsai, and J. Wang, "An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition", DOI10.1109/JSEN.2014.2339843, *IEEE Sensors Journal*.
- [4] J. S. Wang, and F. C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition, " *IEEE Trans. on Industrial Electronics*. Vol. 59 (7), pp. 2998-3007, 2012.
- [5] Dennis Arsenault ; Moving Picture Co., Montreal, QC, Canada ; Anthony D. Whitehead, "Gesture recognition using Markov Systems and wearable wireless inertial sensors, " *IEEE Trans. on Industrial Electronics*. Vol. 59 (7), pp. 2998-3007, 2012.
- [6] ANNE I. DE VRIES¹ , FRANCISCA GALINDO GARRE¹ , LUUK H. ENGBERS¹ , VINCENT H. HILDEBRANDT¹ , and STEF VAN BUUREN^{1,2} ¹ Department of Health Promotion, TNO Quality of Life, Leiden, THE NETHERLANDS; and ² Department of Methodology and Statistics, FSS, University of Utrecht, Utrecht, THE NETHERLANDS
- [7] Annapurna Sharma, and Young-Dong Lee, "High Accuracy Human Activity Monitoring using Neural network, "Division of Electronics, Computer and Telecommunication Engineering, Pukyong National University, Busan 608-737, Korea
- [8] Algoritmo para Implementación DTW Dynamic Time Warping, <https://www.mathworks.com/matlabcentral/fileexchange/43156-dynamic-time-warping--dtw->
- [9] Toolbox para Implementación Hidden Markov Model, <https://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>
- [10] Toolbox para Implementación de RN en Reconocimiento de Patrones, <http://es.mathworks.com/help/nnet/ref/patternnet.html>

12. Anexo A: HW - Esquema y PCB





13. Anexo B: Códigos Principales

13.1. Firmware Microcontrolador:

```

/*****
****//**
    Universidad de Zaragoza

    \file      System.c
    \brief     This file implements the main state machine
    \author    Rade Tecnologias
    \version   1.0
    \date      27/02/2015

    * History:   27/02/2015 - File creation.
    *
    * No part of this document must be reproduced in any form - including
    copied,
    * transcribed, printed or by any electronic means - without specific
    written
    * permission from Javier Atrián Blasco.
    *

*****/

#include "system.h"
#include "functions.h"
#include "PE_Types.h"
#include "string.h"

#include "Cpu.h"
#include "SM1.h"
#include "AS1.h"
#include "PUL1.h"
#include "PUL2.h"
#include "ACEL1_INT1.h"
#include "RX.h"
#include "TX.h"

#include "adx1345.h"
#include "L3GD20.h"
#include "PC_com.h"
#include "HC_06.h"

/*****
****
    ** VARIABLES DECLARATION
    **

*****/

//FLAGS
```

```

unsigned char
flag_SM1_sent,flag_SM1_received,flag_AS1_sent,flag_AS1_received,flag_I
2C_sent,flag_I2C_received;

//MACHINE STATES POSITION
u8 machine_state=MACHINE_STATE_INIT;

u8 a,b;

//BUFFERS
char buffer_SCI_sent[10];

//ACCEL BUFFERS
short int buffer_accel_X[600],buffer_accel_Y[600],buffer_accel_Z[600];
unsigned char buffer_accel_XYZ[6];
long accelerometer_position, gyroscope_position,i;

//ACCEL BUFFERS
short int buffer_gyr_X[600],buffer_gyr_Y[600],buffer_gyr_Z[600];
unsigned char buffer_gyr_XYZ[6];

/*
**
=====
**      Method      :   states_machine_control ()
**/
/*!
**      @brief
**              This function implements the main state machine that
**              manages the system functionality.
**
**      @return
**              None
**/
/*
=====*/

/*****
*/
/*      Cambios para solucionar el tema de la interrupcion incontrolada
***
*
* He quitado el sp9210Reset(); Para evitar el parpadeo      post reset
****
* He medido el tiempo que le lleva a la pantalla hacer el refresco
* quedando en 43 ms
* Así he cambiado el timer de interrupcion en STOP mode a 45 ms desde
10 ms
* creo que se ha arreglado
*/

void states_machine_control(void)
{
    u8 sample=0,flag_new_sample;

    switch (machine_state)
    {
        /* Init state: Sleep */

        case MACHINE_STATE_INIT:

```

```

        //init_HC06();
        read_ID_ADXL345();
        read_ID_L3GD20();

        machine_state=MACHINE_STATE_STAND_BY;

        break;

        /* Stand-by state when awake */
    case MACHINE_STATE_STAND_BY:

        RX_ConnectPin(RX_DeviceData);
        TX_ConnectPin(TX_DeviceData);

        Cpu_SetOperationMode(DOM_STOP, NULL, NULL); /* next
interrupt will wake up us */

        i=(!PUL1_GetVal(PUL1_DeviceData))||(!PUL2_GetVal(PUL1_DeviceData
));
        //
        if(i){
            machine_state=MACHINE_STATE_MEASURING;
            start_measurement_ADXL345();
            start_measurement_L3GD20();
            accelerometer_position=0;
            gyroscope_position=0;
        }

        break;

        /* Measuring state while PUL1 is pulled */
    case MACHINE_STATE_MEASURING:

        if((!PUL1_GetVal(PUL1_DeviceData))||(!PUL2_GetVal(PUL1_DeviceData)))
        {
            a=check_INT1_SPI_L3GD20();
            b=check_INT1_SPI_ADXL345();
            if(a&0x08){
                //
                read_XYZ_L3GD20();
                //
                buffer_gyr_X[gyroscope_position]=(signed
int)(buffer_gyr_XYZ[1]<<8|buffer_gyr_XYZ[0]);
                //
                buffer_gyr_Y[gyroscope_position]=(signed
int)buffer_gyr_XYZ[3]<<8|buffer_gyr_XYZ[2];
                //
                buffer_gyr_Z[gyroscope_position]=(signed
int)buffer_gyr_XYZ[5]<<8|buffer_gyr_XYZ[4];
                //
                gyroscope_position++;
            }
            if(b&0x80){
                read_XYZ_L3GD20();
                buffer_gyr_X[gyroscope_position]=(signed
int)(buffer_gyr_XYZ[1]<<8|buffer_gyr_XYZ[0]);
                buffer_gyr_Y[gyroscope_position]=(signed
int)buffer_gyr_XYZ[3]<<8|buffer_gyr_XYZ[2];

```

```

        buffer_gyr_Z[gyroscope_position]=(signed
int)buffer_gyr_XYZ[5]<<8|buffer_gyr_XYZ[4];
        gyroscope_position++;
        read_XYZ_ADXL345();
        buffer_accel_X[accelerometer_position]=(signed
int)(buffer_accel_XYZ[1]<<8|buffer_accel_XYZ[0]);
        buffer_accel_Y[accelerometer_position]=(signed
int)buffer_accel_XYZ[3]<<8|buffer_accel_XYZ[2];
        buffer_accel_Z[accelerometer_position]=(signed
int)buffer_accel_XYZ[5]<<8|buffer_accel_XYZ[4];
        accelerometer_position++;
    }

    if(accelerometer_position==600){
        machine_state=MACHINE_STATE_SEND;
    }

    }else{
        machine_state=MACHINE_STATE_SEND;
    }

    break;

    /* Measuring state while PUL1 is pulled */
case MACHINE_STATE_SEND:

    stop_measurement_ADXL345();
    stop_measurement_L3GD20();

    AS1_ConnectPin(AS1_DeviceData,LDD_SERIAL_RX_PIN);
    AS1_ConnectPin(AS1_DeviceData,LDD_SERIAL_TX_PIN);

    // TRAMA ACELEROMETRO
    write_character();

    for(i=0;i<600;i++){
        buffer_accel_X[i]=0;
        buffer_accel_Y[i]=0;
        buffer_accel_Z[i]=0;
        buffer_gyr_X[i]=0;
        buffer_gyr_Y[i]=0;
        buffer_gyr_Z[i]=0;
    }

    machine_state=MACHINE_STATE_STAND_BY;
    break;
}
//WDog1_Clear(WDog1_DeviceData);
} // End states_machine_control()

```


13.2. Código con algoritmo captura y compensación de gravedad. (MATLAB)

```

clear all
clc

text_block_size=5;
ms=0.001;
% s = serial('COM12');
% set(s,'BaudRate',57600);
% set(s,'TimeOut',10);

% type_character=input('Introduce type character: ','s');

s=Bluetooth('PENGEASURE',1);
fprintf('Blue Port: Generated');
fopen(s);

%% CAPTURA SEÑAL

while(1)

    text_block_size=5;
    ms=0.001;

    fprintf('Write\n');
    char(fread(s,text_block_size));
    pause(ms);
    fwrite(s,'OKEY\n');
    %      fprintf('BT transmitting\n');

    %% ACCEL_X
    for(i=1:6)
        accel_x_aux((i-1)*200+1:(i-1)*200+200)=fread(s,200);
        pause(ms);
        fwrite(s,'OKEY\n');
        %      pause(ms);
    end
    fprintf(' , ACCEL_X readed\n');

    %% ACCEL_Y
    for(i=1:6)
        accel_y_aux((i-1)*200+1:(i-1)*200+200)=fread(s,200);
        pause(ms);
        fwrite(s,'OKEY\n');
        %      pause(ms);
    end
    fprintf(' , ACCEL_Y readed\n');

    %% ACCEL_Z
    for(i=1:6)
        accel_z_aux((i-1)*200+1:(i-1)*200+200)=fread(s,200);
        pause(ms);
        fwrite(s,'OKEY\n');
        %      pause(ms);
    end
    fprintf(' , ACCEL_Z readed\n');

```

```

%% GYRO_X
for (i=1:6)
    gyro_x_aux((i-1)*200+1:(i-1)*200+200)=fread(s,200);
    pause(ms);
    fwrite(s,'OKEY/n');
    % pause(ms);
end
fprintf(' , GYRO_X readed\n');

%% GYRO_Y
for (i=1:6)
    gyro_y_aux((i-1)*200+1:(i-1)*200+200)=fread(s,200);
    pause(ms);
    fwrite(s,'OKEY/n');
    % pause(ms);
end
fprintf(' , GYRO_Y readed\n');

%% GYRO_Z
for (i=1:6)
    gyro_z_aux((i-1)*200+1:(i-1)*200+200)=fread(s,200);
    pause(ms);
    fwrite(s,'OKEY/n');
    % pause(ms);
end
fprintf(' , GYRO_Z readed\n');

%
    fclose(s);
%% CONVERT
for (i=1:1:600)
    accel_x(i)=accel_x_aux(2*i-1)*256+accel_x_aux(2*i);
    if(accel_x(i)>32768)
        accel_x(i)=accel_x(i)-65536;
    end
    accel_y(i)=accel_y_aux(2*i-1)*256+accel_y_aux(2*i);
    if(accel_y(i)>32768)
        accel_y(i)=accel_y(i)-65536;
    end
    accel_z(i)=accel_z_aux(2*i-1)*256+accel_z_aux(2*i);
    if(accel_z(i)>32768)
        accel_z(i)=accel_z(i)-65536;
    end
    gyro_x(i)=gyro_x_aux(2*i-1)*256+gyro_x_aux(2*i);
    if(gyro_x(i)>32768)
        gyro_x(i)=gyro_x(i)-65536;
    end
    gyro_y(i)=gyro_y_aux(2*i-1)*256+gyro_y_aux(2*i);
    if(gyro_y(i)>32768)
        gyro_y(i)=gyro_y(i)-65536;
    end
    gyro_z(i)=gyro_z_aux(2*i-1)*256+gyro_z_aux(2*i);
    if(gyro_z(i)>32768)
        gyro_z(i)=gyro_z(i)-65536;
    end

    accel_data=[accel_x;accel_y;accel_z]';
    gyro_data=[gyro_x;gyro_y;gyro_z]';

```

```

    ancho_filtro=2;
    for k=1:size(accel_data,1)-ancho_filtro
        accel_data(k,:)=mean(accel_data(k:k+ancho_filtro,:),1);
        gyro_data(k,:)=mean(gyro_data(k:k+ancho_filtro,:),1);
    end

    u=0;
    for s=1:600
        if(accel_data(s,:)==[0 0 0])
            accel_data(s,:)=accel_data(u-1,:);
        else
            u=s;
        end
    end

%% or

%    load example

    accel=accel_data;
    gyro=gyro_data;

% Bias compensation
%    gyro=gyro-repmat([60 40 60],600,1);

    accel=accel/128;
    gyro=gyro*0.0175*pi/180;
    gyro=[gyro(:,2) -gyro(:,1) gyro(:,3)];

    clearvars -except accel gyro s

%% SEGMENTACION

    Ma_r=abs(sqrt(accel(:,1).^2+accel(:,2).^2+accel(:,3).^2)-1);
    Mg_r=sqrt(gyro(:,1).^2+gyro(:,2).^2+gyro(:,3).^2);

    for i=1:size(Ma_r,1)-10

        Ma(i,:)=mean(Ma_r(i:i+10,:),1);
        Mg(i,:)=mean(Mg_r(i:i+10,:),1);

    end

    for i=25:size(Ma,1)
        if((Mg(i)>0.40)||Ma(i)>0.25)    %% Original 0.3    0.2
            t_start=i-15;
            break;
        end
    end
    for i=t_start+50:size(Ma,1)
        if((Mg(i)<0.2)&&(Mg(i+1)<0.2))
            t_end=i+10;
            break;
        end
    end

%% NORMALIZACION

```

```

ax=accel(t_start:t_end,1);
ax=resample(ax,round(10000/size(ax,1)),1);
ax=resample(ax,1,round(size(ax,1)/100));
accel_b(:,1)=ax(1:100);

ay=accel(t_start:t_end,2);
ay=resample(ay,round(10000/size(ay,1)),1);
ay=resample(ay,1,round(size(ay,1)/100));
accel_b(:,2)=ay(1:100);

az=accel(t_start:t_end,3);
az=resample(az,round(10000/size(az,1)),1);
az=resample(az,1,round(size(az,1)/100));
accel_b(:,3)=az(1:100);

ax=gyro(t_start:t_end,1);
ax=resample(ax,round(10000/size(ax,1)),1);
ax=resample(ax,1,round(size(ax,1)/100));
gyro_b(:,1)=ax(1:100);

ay=gyro(t_start:t_end,2);
ay=resample(ay,round(10000/size(ay,1)),1);
ay=resample(ay,1,round(size(ay,1)/100));
gyro_b(:,2)=ay(1:100);

az=gyro(t_start:t_end,3);
az=resample(az,round(10000/size(az,1)),1);
az=resample(az,1,round(size(az,1)/100));
gyro_b(:,3)=az(1:100);

accel=accel_b;
gyro=gyro_b;

%% REFERENCIANDO EJES
g_value_init=mean([accel(1:10,1) accel(1:10,2) accel(1:10,3)]);

x=g_value_init(1);
y=g_value_init(2);
z=g_value_init(3);

if z<0
    roll=atan(y/z);
elseif (z>0)&&(y>0)
    roll=-pi+atan(y/z);
elseif (z>0)&&(y<0)
    roll=+pi+atan(y/z);
end

pitch=atan(x/sqrt(y^2+z^2));

yaw=0;

roll_init=roll;
pitch_init=pitch;
yaw_init=yaw;

```

```

q_init=angle2quat(yaw_init,pitch_init,roll_init);

[yaw_init pitch_init roll_init]=quat2angle(q_init);
q=q_init;

for i=2:100
    omega(:,:,i)= [ 0 -gyro(i,1) -gyro(i,2) -gyro(i,3);
                    gyro(i,1) 0 gyro(i,3) -gyro(i,2);
                    gyro(i,2) -gyro(i,3) 0 gyro(i,1);
                    gyro(i,3) gyro(i,2) -gyro(i,1) 0];
end

for i=2:100
    q(i,:)=(expm(omega(:,:,i)/190)*q(i-1,:))';
    [yaw(i) pitch(i) roll(i)]=quat2angle(q(i,:));
end

eu=[roll' pitch' yaw'];

eu_d=eu/pi*180;

g_value_end=mean([accel(98:100,1) accel(98:100,2)
accel(98:100,3)]);
x=g_value_end(1);
y=g_value_end(2);
z=g_value_end(3);

if z<0
    roll_end=atan(y/z);
elseif (z>0)&&(y>0)
    roll_end=-pi+atan(y/z);
elseif (z>0)&&(y<0)
    roll_end=+pi+atan(y/z);
end
pitch_end=atan(x/sqrt(y^2+z^2));
yaw_end=yaw(100);

[yaw_end2 pitch_end2 roll_end2]=quat2angle(q(100,:));
q_end=angle2quat(yaw_end,pitch_end,roll_end);
mq=(q(100,:)-q_end)/100;

for i=1:100
    q_corr(i,:)=q(i,:);%-mq*i;
    [yaw(i) pitch(i) roll(i)]=quat2angle(q_corr(i,:));
end
angle=[roll' pitch' yaw'];

q0=q_corr(:,1);
q1=q_corr(:,2);
q2=q_corr(:,3);
q3=q_corr(:,4);

for i=1:100
    T(:,:,i)=[q0(i)^2+q1(i)^2-q2(i)^2-q3(i)^2 2*(q1(i)*q2(i)-
q0(i)*q3(i)) 2*(q1(i)*q3(i)+q0(i)*q2(i));
              2*(q1(i)*q2(i)+q0(i)*q3(i)) q0(i)^2-q1(i)^2+q2(i)^2-
q3(i)^2 2*(q2(i)*q3(i)-q0(i)*q1(i))];

```

```

                2*(q1(i)*q3(i)-q0(i)*q2(i)) 2*(q2(i)*q3(i)+q0(i)*q1(i))
q0(i)^2-q1(i)^2-q2(i)^2+q3(i)^2];
end

for i=1:100

    referenced_acceleration(i,:)=(T(:,:,i)*accel(i,:))';

end

%% COMPENSACION GRAVEDAD

for i=1:100

    compensated_acceleration(i,:)=referenced_acceleration(i,:)+[0
0 1];

end

compensated_velocity(1,:)= [0 0 0];
compensated_position(1,:)= [0 0 0];

for i=2:100

    compensated_velocity(i,:)=compensated_velocity(i-
1,:)+compensated_acceleration(i,:);

end

mv=(compensated_velocity(100,:)-compensated_velocity(1,:))/100;

for i=1:100

    compensated_velocity(i,:)=compensated_velocity(i,:)-mv*i;

end

for i=2:100

    compensated_position(i,:)=compensated_position(i-
1,:)+compensated_velocity(i,:);

end

for i=1:100

body_acceleration(i,:)=(inv(T(:,:,i))*compensated_acceleration(i,:))'
;

end

body_velocity(1,:)= [0 0 0];

```

```

body_position(1,:)=[0 0 0];

for i=2:100

    body_velocity(i,:)=body_velocity(i-
1,:)+body_acceleration(i,:);

end

mv=(body_velocity(100,:)-body_velocity(1,:))/100;

for i=1:100

    body_velocity(i,:)=body_velocity(i,:)-mv*i;

end

for i=2:100

    body_position(i,:)=body_position(i-1,:)+body_velocity(i,:);

end

%% COORDENADAS Y NORMALIZACION

% plot(-body_position(:,3),-body_position(:,2));
coorx=-body_position(:,3);
coory=-body_position(:,2);

maximox=max(max(coorx));maximoy=max(max(coory));
minimox=min(min(coorx));minimoy=min(min(coory));

desviacionx=maximox-minimox;
desviaciony=maximoy-minimoy;

coorx=(coorx-repmat(minimox,100,1))./repmat(desviacionx,100,1);
coory=(coory-repmat(minimoy,100,1))./repmat(desviaciony,100,1);
clearvars -except coorx coory s
% plot(coorx,coory);
%
% drawnow;
% pause(3);

load netexport

data1=resample(coorx,round(10000/size(coorx,1)),1);
data1=resample(data1,1,round(size(data1,1)/30));
data2=resample(coory,round(10000/size(coory,1)),1);
data2=resample(data2,1,round(size(data2,1)/30));

data1(data1>1)=1;
data2(data2>1)=1;
data1(data1<0)=0;
data2(data2<0)=0;

plot(coorx,coory);
drawnow;
pause(0.1);

```

```
for i=1:size(net1_e,2)
    res(:,i,1)=net1_e{1,i}(data1);
    res(:,i,2)=net2_e{1,i}(data2);
end

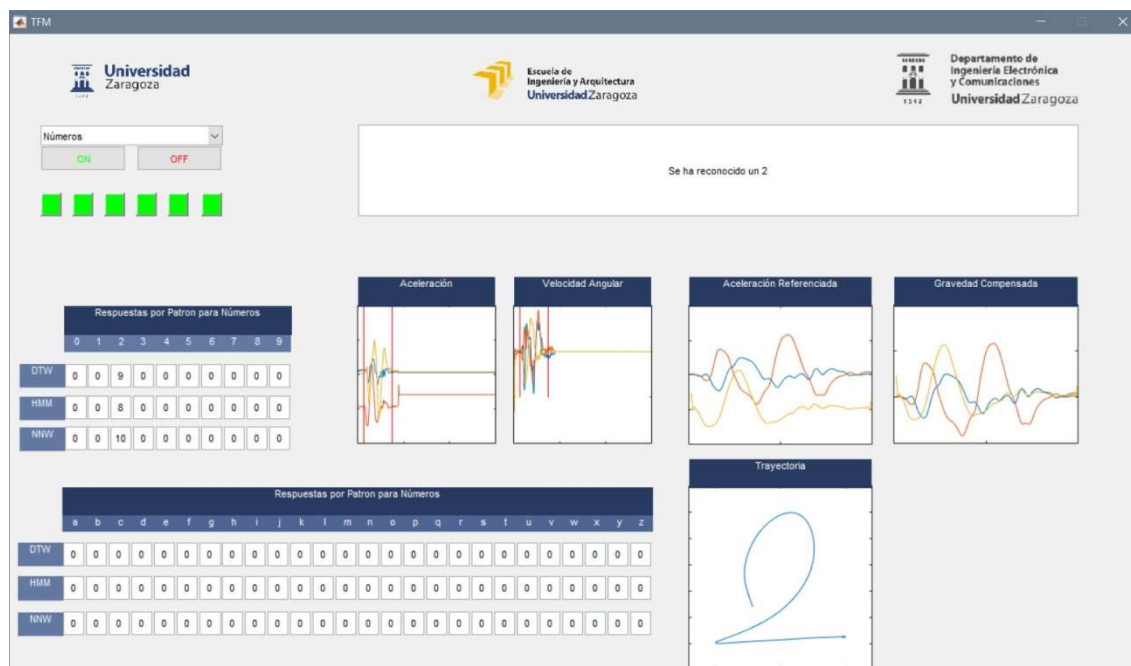
res=res(:,:,1)+res(:,:,2);
res=mean(res,2);

[valor numero]=max(res);

fprintf([num2str(numero-1) '\n']);
```

End

14. Anexo C: Aplicación PC



En el siguiente grafico podemos observar el grafo de estados por los que pasa la aplicación durante su operación.

