

Implementación de un Algoritmo de Detección de Caídas con Contextualización del Evento usando SVM y Sensores Inerciales

Roberth Arturo Álvarez Jiménez
Ingeniero Electrónico



Universidad Industrial de Santander
Facultad de Ingenierías Físico-Mecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Maestría en Ingeniería Electrónica
Bucaramanga
2017

Implementación de un Algoritmo de Detección de Caídas con Contextualización del Evento Usando SVM y Sensores Inerciales

Roberth Arturo Álvarez Jiménez
Ingeniero Electrónico

Trabajo de investigación presentado como requerimiento parcial para optar al título de:
Magíster en Ingeniería Electrónica

Directora
Edith Pulido Herrera
Ph.D. en Ingeniería

Codirector
Daniel Alfonso Sierra Bueno
Ph.D. en Ingeniería Biomédica



Universidad Industrial de Santander
Facultad de Ingenierías Físico-Mecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Maestría en Ingeniería Electrónica
Bucaramanga
2017

Agradecimientos

Primero le agradezco a Dios por darme la paciencia y el entendimiento para lograr la culminación de este trabajo de investigación.

Agradezco a mis padres y mi hermana por su apoyo y acompañamiento en este proceso, sé que les llenará de orgullo tanto como a mí este logro.

Gracias a los profesores Edith Pulido Herrera y Daniel Alfonso Sierra Bueno por su dirección, exigencia, atención y consejos durante el desarrollo de este proyecto.

Gracias a Juan Camilo, Jenny, Germán, Felipe y Clara por su acompañamiento y consejos, fueron bastante útiles para mi proceso de aprendizaje e investigación.

Gracias a mi novia Tatiana Rodríguez, por su paciencia y apoyo incondicional durante mis estudios de maestría.

Finalmente, gracias a amigos y compañeros de estudio por sus buenos consejos y recomendaciones.

Tabla de Contenido

Introducción	13
1 Marco Conceptual y Estado del Arte	15
1.1 Introducción	15
1.2 Características y Fases de una Caída	15
1.3 Características de ADL desde el Punto de Vista Inercial	17
1.4 Algoritmos de Detección de Caídas	18
1.5 Algoritmos de Detección de Actividades Cotidianas (ADL)	21
1.6 Resumen	22
2 Creación de Base de Datos de Actividades Cotidianas y Caídas	23
2.1 Introducción	23
2.2 Actividades a Detectar	23
2.3 Equipos y Montaje para la Adquisición de Datos	23
2.3.1 Dispositivos y Sensores.	23
2.3.2 Ubicación de Sensores y Señales Adquiridas.	25
2.4 Protocolo de Toma de Datos	26
2.4.1 Datos para Actividades Cotidianas.	26
2.4.2 Datos de Entrenamiento para la Detección de Caídas.	27
2.5 Resumen	27
3 Selección de Características y Diseño del Algoritmo de Detección de Caídas y ADL	29
3.1 Introducción	29
3.2 Selección de Características	29
3.3 Diseño del Algoritmo	30
3.3.1 Adquisición y Pre-procesamiento de Datos.	30
3.3.2 Clasificación de Actividades sin Desplazamiento.	31
3.3.3 Clasificación de Actividades con Desplazamiento.	31
3.3.4 Detección de Caídas.	31
3.4 Resumen	35
4 Implementación de Algoritmos de Detección de ADL y Caídas	36
4.1 Introducción	36
4.2 Algoritmo de Detección de Actividades y Caídas Offline	36
4.2.1 Clasificadores SVM.	36
4.2.2 Validación del Algoritmo.	37
4.3 Implementación en Tiempo Real del Algoritmo de Detección de Caídas y ADL	38
4.3.1 Selección de Python para la Implementación del Algoritmo.	38
4.3.2 Implementación del Algoritmo.	38
4.3.3 Pruebas de Validación para el Algoritmo de Detección de Caídas	40
4.3.4 Pruebas de Validación para el Algoritmo de Detección de Actividades	42

4.3.5 Desempeño del Algoritmo Completo en Tiempo Real	42
4.4 Resumen	43
5 Conclusiones y Trabajo Futuro	45
5.1 Conclusiones Generales	45
5.2 Trabajo Futuro	46
Bibliografía	51
Referencias	56
Apéndices	57

Lista de Figuras

Figura 1	Fases de una caída observadas sobre la señal de aceleración vertical. Adaptado de [47]	16
Figura 2	Aceleraciones en X, Y y Z del sensor sobre la cintura	16
Figura 3	Señales de aceleración correspondientes a la ausencia y presencia de movimiento . .	17
Figura 4	Señales de aceleración vertical (arriba) y diferencial de presión (abajo) para movimiento en ascensor (azul) y permanecer quieto (rojo).	18
Figura 5	Comparación de señales de aceleración vertical para caminar y correr	19
Figura 6	Comparación entre las señales de presión y aceleración vertical para caminata en escaleras, hacia adelante y correr	19
Figura 7	AltIMU 10 v4 (izquierda) e IMU xsens MTi-1 (derecha)	24
Figura 8	Montaje de sensores sobre el usuario	26
Figura 9	Ubicación de la IMU Xsens a la altura de la cintura sobre una barra para pruebas de adquisición de datos	27
Figura 10	Señal de aceleración de una caída simulada ubicando sensores sobre una barra (roja) y simulada con una persona (azul)	28
Figura 11	Distribución de datos en el espacio de características para las actividades caminar y correr	30
Figura 12	Distribución de datos en el espacio de características para caídas (azul) y Otras actividades (rojo).	31
Figura 13	Primera parte del algoritmo de reconocimiento de actividades.	33
Figura 14	Segunda parte del algoritmo de reconocimiento de actividades.	34
Figura 15	Ejemplo de detección de actividades. Los marcadores indican la actividad detectada en cada iteración	37
Figura 16	Señal de presión filtrada a 1 Hz (arriba) y 0.3 Hz (abajo)	40
Figura 17	Interfaz de usuario para el algoritmo en tiempo real	40
Figura 18	Ejemplo de pruebas realizadas para la validación de caídas realizada con voluntarios	41
Figura 19	Recorrido realizado durante las pruebas de validación del algoritmo de detección de actividades	42
Figura 20	Recorrido realizado durante las pruebas de validación del algoritmo de detección de actividades	43

Lista de Tablas

Tabla 1	Ubicaciones comunes de IMUs sobre el cuerpo	20
Tabla 2	Actividades detectables para el algoritmo	23
Tabla 3	Especificaciones de los sensores contenidos en la altIMU 10 v4	24
Tabla 4	Especificaciones de la IMU Xsens MTi-1	25
Tabla 5	Características de la BeagleBone Black	25
Tabla 6	Señales tomadas de los sensores durante las pruebas experimentales	26
Tabla 7	Características obtenidas luego del proceso de selección usando MATLAB	30
Tabla 8	Parámetros de desempeño para el clasificador SVM “caminar - correr”	36
Tabla 9	Parámetros de desempeño para el clasificador SVM “caídas - otros”	37
Tabla 10	Parámetros de desempeño para el algoritmo completo de detección de caídas y ADL (offline)	38
Tabla 11	Parámetros de desempeño para el clasificador SVM “caminar - correr”	39
Tabla 12	Parámetros de desempeño para el clasificador SVM “caídas - otros”	39
Tabla 13	Parámetros de desempeño para el algoritmo de detección de caídas y actividades (tiempo real)	43

Lista de Apéndices

Apéndices	57
Apéndice A Listado de Características	58
Apéndice B Máquinas de Soporte Vectorial (SVM)	61
Apéndice C Algoritmo de Ajuste de Longitud del Paso	64
Apéndice D Validación de Algoritmos de Reconocimiento de Patrones	68
D.0.1 Validación Cruzada	69

Resumen

TÍTULO: IMPLEMENTACIÓN DE UN ALGORITMO DE DETECCIÓN DE CAÍDAS CON CONTEXTUALIZACIÓN DEL EVENTO USANDO SVM Y SENSORES INERCIALES

AUTOR: ROBERTH ARTURO ÁLVAREZ JIMÉNEZ¹

PALABRAS CLAVES: Máquinas de soporte vectorial, unidad de medida inercial (IMU), Reconocimiento de actividades.

DESCRIPCIÓN:

Uno de los problemas que amenaza la integridad física de los adultos mayores son las caídas, la necesidad de brindar atención oportuna a alguien que sufra una caída ha llevado al desarrollo de múltiples dispositivos y algoritmos enfocados en esta problemática. En el presente trabajo se propone un algoritmo para la detección de actividades cotidianas para personas (ADL²), enfocado en detectar caídas brindando información contextualizada basada en otras actividades y usando unidades de medida inercial (IMU³). Para el cumplimiento de este objetivo se planteó la siguiente metodología:

Se realizó una revisión del estado del arte en cuanto a alternativas para la detección de caídas y ADL, así como las limitaciones al trabajar con unidades de medida inercial. Seguidamente, se creó una base de datos a partir de pruebas experimentales realizadas por voluntarios entre los 20 y 30 años. Con esta base de datos se realizó un análisis de las diferentes señales adquiridas, permitiendo la selección de características para entrenar el algoritmo de detección de caídas, caminata normal, correr, caminata en escaleras, permanecer de pie y acostado. Se desarrolló un algoritmo completo para reducir la aparición de falsos positivos en la detección, especialmente en las caídas. Inicialmente el algoritmo fue evaluado de manera *offline*.

Al implementar en tiempo real, se encuentra que al combinar actividades con caídas se pueden causar errores en la detección y se realizaron las modificaciones necesarias para evitar estos errores. se realizaron pruebas de validación en las cuales los usuarios realizaban las diferentes actividades que el algoritmo puede clasificar. Se obtuvo la tasa de error cercana al 8 % en la detección. Se plantea la posibilidad del desarrollo de un dispositivo que ejecute el algoritmo, e incluso la implementación en teléfonos inteligentes se considera como una opción viable.

¹Facultad de Ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

Directora: Edith Pulido Herrera, PhD.

Codirector: Daniel Alfonso Sierra Bueno, PhD.

²siglas para “Activities of Daily Living”

³siglas para “Inertial Measurement Units”

Abstract

TITLE: IMPLEMENTATION OF AN ALGORITHM FOR FALLING DETECTION WITH EVENT CONTEXTUALIZATION USING SVM AND INERTIAL SENSORS⁴

AUTHOR: ROBERTH ARTURO ÁLVAREZ JIMÉNEZ⁵

KEYWORDS: Inertial measurement unit (IMU), Support vector machines (SVM), activities identification.

DESCRIPTION:

Falling is one of the main problems that affects the elderly, causing the creation of systems and algorithms to give a proper attention to someone who suffer a fall. This Master research project proposes an algorithm for the detection of Activities of daily living (ADL), focused on falling detection giving contextualized information based on other activities and using inertial measurement units (IMU). In order to fulfill the main objective, the following methodology was proposed:

Initially, a revision of the state of the art related with ADL and falling detection was conducted, to study the detection techniques and features involved in this tasks, also the actual limitations when working with IMUs.

Afterwards, a dataset was created from experimental tests performed by volunteers in ages between 20 and 30 years. Using this dataset it was possible to perform a complete analysis of the acquired signals, achieving the feature selection to train the algorithm and perform the detection of falling, walking, running, stairs walking, standing and lying. Once the classifier is trained, a full algorithm was designed to reduce the false positives in the detection, specially for falling. In first instance, the algorithm was evaluated offline to offer a reference to the real time performance.

For experimental validation of the algorithm, some test similar to the ones performed to create de database were conducted, the participants followed a route where they had to do the different activities that the algorithm can classify. A general error rate of around 8%. Finally, based on the results obtained we concluded that the performance of the algorithm is good and has a low error rate detecting ADL and falls. The development of a device for the algorithm execution is proposed as future work, also the smartphones implementation is considered as a viable option.

⁴Research Work

⁵Facultad de Ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.
Director: Edith Pulido Herrera, PhD. Codirector: Daniel Alfonso Sierra Bueno, PhD.

Introducción

Esta comprobado que las caídas son uno de los eventos que pueden comprometer la integridad física de las personas, como es el caso de mujeres embarazadas o adultos mayores, se estima que de los adultos mayores que sufren caídas el 4 % sufrirán fracturas, y de estos el 11 % sufrirán lesiones graves [40]. Para combatir esta problemática, se han planteado diversos sistemas para la detección de caídas y así lograr una atención oportuna al afectado, el principal problema de estos sistemas está en que la mayoría se enfocan solo en la detección de caídas, dejando de lado la influencia de las otras actividades que una persona puede realizar, En escenarios reales es más común observar caídas luego de actividades cotidianas como caminar, bajar escaleras, etc. En dichos casos puede no ser tan claro el impacto o confundirse con las señales producidas por otra actividad, lo que conlleva a la existencia de falsos positivos en la detección. Algunos sistemas son rechazados por los usuarios debido a estas falsas alarmas que incomodan tanto al usuario como a familiares y cuidadores. Además, algunos sistemas funcionan basados en “botones de pánico” que no pueden ser accionados si el usuario cae inconsciente.

Basándose en lo anterior, un algoritmo capaz de identificar actividades cotidianas como correr, caminar, subir o bajar escaleras es un complemento ideal para un sistema detector de caídas, dado que permite reducir la cantidad de falsos positivos al evitar la confusión de una caída con actividades de otro tipo. En este proyecto se propone un algoritmo automático de detección de caídas contextualizado, es decir, capaz de detectar actividades cotidianas y diferenciarlas de una caída, con lo que también se puede estimar que ocurrió con el usuario antes y después de caer. Además, la detección se hace a partir de señales inerciales (Aceleraciones y velocidades angulares) y de presión adquiridas por un sensor ubicado en la cintura del usuario, manteniendo así la privacidad del individuo a diferencia de otros sistemas como los implementados usando sistemas de visión que además pueden resultar bastante costosos.

El algoritmo propuesto permite obtener información más completa sobre las caídas al considerar los momentos pre y post impacto, lo cual es útil para estimar la gravedad de la caída y ofrecer asistencia oportuna. Se mantiene como eje principal el desarrollo desde el punto de vista de *software*, usando herramientas de simulación *offline* para diseñar el algoritmo y pasar luego a una implementación en tiempo real. Además, se creó una base de datos con datos de aceleración, campo magnético, velocidades angulares y presión que queda a disposición para investigaciones futuras.

Con los resultados de la implementación del algoritmo de detección de caídas se brinda una herramienta para el cuidado especialmente de adultos mayores, quienes son los más propensos a sufrir lesiones debido a estos eventos, sin dejar de lado que puede aplicarse también a otras poblaciones a las que se requiera monitorear, por ejemplo: mujeres embarazadas y niños. Considerando que en Colombia estos sistemas no son utilizados ampliamente, se propone como un punto de partida para el desarrollo de dispositivos de este tipo en el país. Además se espera que pueda ser utilizado por diferentes usuarios sin inconveniente.

A partir de los resultados obtenidos de esta investigación, se presentó el artículo “*Climbing / Descending Stairs Detection using PCA and a SVM Classifier.*” en el “Congreso Latinoamericano de Ingeniería Biomédica” que se realizó en Bucaramanga del 26 al 28 de octubre de 2016.

Este documento se estructura de la siguiente manera: el primer capítulo contiene el marco conceptual necesario para la interpretación del trabajo de investigación, y los aspectos más importantes de la revisión del estado del arte en cuanto a algoritmos de detección de caídas y actividades de la vida diaria. Luego, en el segundo capítulo se describe detalladamente el proceso de adquisición de datos para el entrenamiento del algoritmo, teniendo en cuenta sensores, señales adquiridas y autorización del comité de ética para llevar a cabo las pruebas.

En el tercer capítulo se muestra la selección de características y la descripción del flujo de datos del algoritmo desglosando por separado la detección de caídas y la de detección de las demás actividades, para después en el cuarto capítulo describir las implementación *offline* y en tiempo real del algoritmo, resaltando los desafíos y condiciones necesarias para lograr una buena implementación en tiempo real. en esta sección se incluye también el proceso de validación del algoritmo, así como los resultados obtenidos a partir del análisis de desempeño de cada una de las actividades.

En el quinto capítulo se presentan las conclusiones finales del trabajo, así como el trabajo futuro derivado de esta investigación. Por último se muestran en los anexos algunos conceptos importantes como las herramientas para validar el algoritmo, el funcionamiento de las máquinas de soporte vectorial, entre otros.

Capítulo 1

Marco Conceptual y Estado del Arte

1.1. Introducción

Los algoritmos de aprendizaje de máquina supervisados involucran dos problemas principales: regresión y clasificación. La detección de actividades realizadas por personas se describe como un problema de clasificación, en este tipo de algoritmos se busca a través de análisis de movimiento poder establecer patrones que obedezcan a las actividades que se busca identificar [2]. El movimiento de una persona puede ser registrado usando diferentes sensores o sistemas tales como: sensores inerciales, sistemas de visión artificial, sensores de presión, etc [36].

La detección de actividades cotidianas (Ej: caminar, correr, sentarse) ha sido estudiada ampliamente en la literatura ofreciendo diferentes metodologías para este propósito [11]. En el caso específico de eventos irregulares como las caídas, se han planteado diferentes alternativas donde el mayor inconveniente es la aparición de falsos positivos que pueden alertar a cuidadores y/o familiares cuando no hay ninguna emergencia.

En el caso de algoritmos basados en la toma de datos con sensores inerciales, se tiene la ventaja de que estos dispositivos pueden ser ubicados sobre el cuerpo de manera no invasiva, permitiendo recolectar datos directamente del usuario. Al usar este tipo de sensores es necesario tener especial cuidado en seleccionar su ubicación sobre el cuerpo, debido a que las señales muestran diferentes patrones dependiendo de la ubicación sobre el cuerpo.

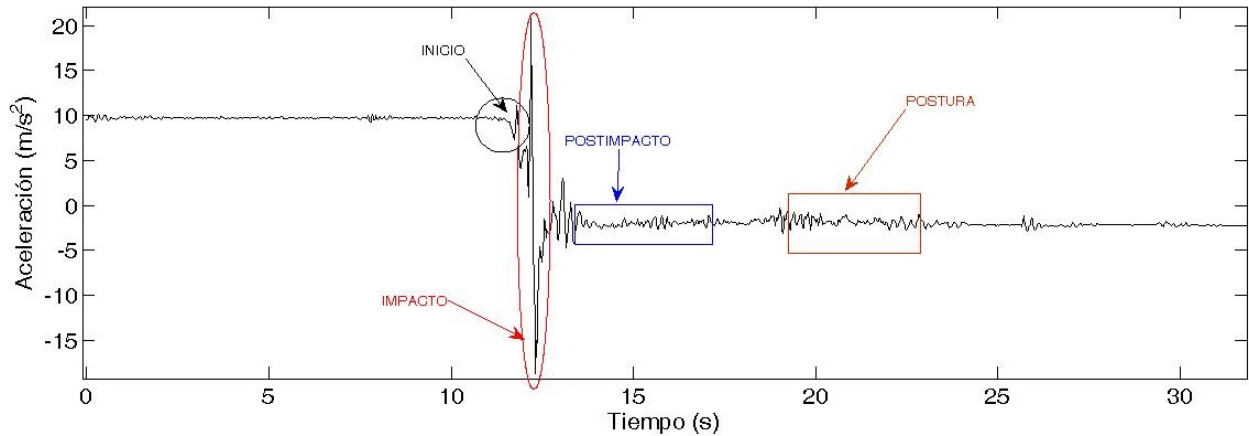
Algunas de las metodologías planteadas atacan el problema haciendo análisis de las diferentes señales inerciales (aceleraciones y velocidades angulares) buscando reducir la cantidad de falsos positivos usando técnicas basadas en umbrales o técnicas de aprendizaje de máquina. En este capítulo se da una breve descripción del problema de las caídas en adultos mayores para luego repasar los diferentes algoritmos propuestos para la detección de caídas y las diferentes actividades cotidianas.

1.2. Características y Fases de una Caída

Las caídas son consideradas como un problema de salud pública que puede tener serias consecuencias especialmente en adultos mayores, es por eso que se estudian maneras de ayudar en su prevención o asistencia rápida. De acuerdo a como se menciona en [47], en la imagen 1 se observan las fases de una caída sobre la señal de aceleración vertical. las etapas de una caída se describen de la siguiente manera:

- ❖ **Inicio:** en esta fase el sujeto empieza a caer y sobre la señal de aceleración puede verse un descenso en el valor de la gravedad que se registra mientras el usuario se encuentra de pie.

Figura 1: Fases de una caída observadas sobre la señal de aceleración vertical. Adaptado de [47]



- ❖ **Impacto:** luego de perder el equilibrio, se da el impacto contra el suelo u otros objetos, se alcanza un pico de aceleración bastante alto superando los $20 m/s^2$.
- ❖ **Post-impacto:** durante esta fase, el sujeto se mantiene inmóvil independientemente de la gravedad de la caída, luego de un momento puede intentar levantarse.
- ❖ **Postura:** luego de la caída, se da un cambio de postura en el cual las medidas de los ejes de aceleración toman mediciones bastante diferentes a las que se registraban antes del impacto. Por ejemplo, en la imagen 2 se puede ver que el eje vertical ahora muestra valores cercanos a cero, mientras que antes de la caída registraba el valor de la gravedad, dicho valor ahora se muestra en el eje x de la aceleración.

La estructura mostrada anteriormente corresponde a una caída aislada, es decir no se dan otros movimientos antes o después del evento.

Figura 2: Aceleraciones en X, Y y Z del sensor sobre la cintura

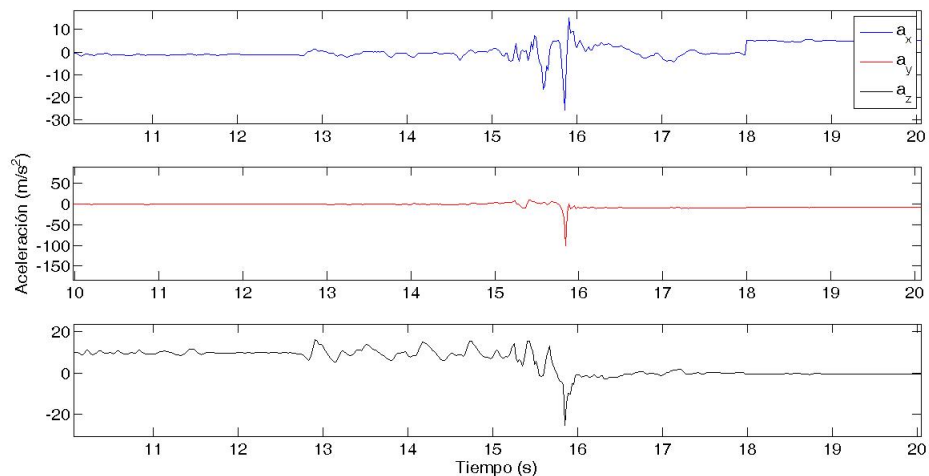
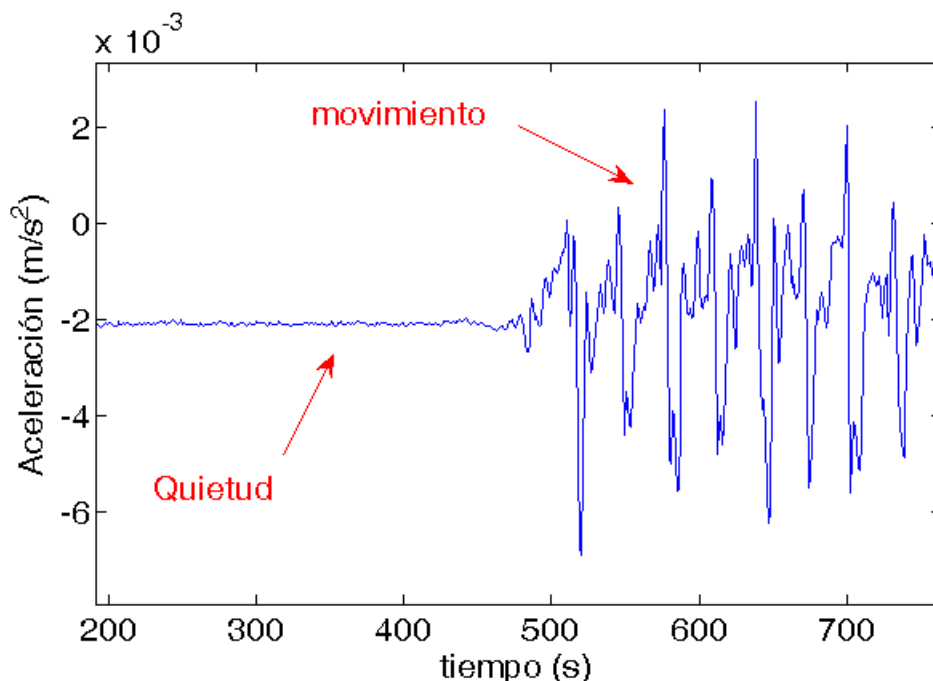


Figura 3: Señales de aceleración correspondientes a la ausencia y presencia de movimiento



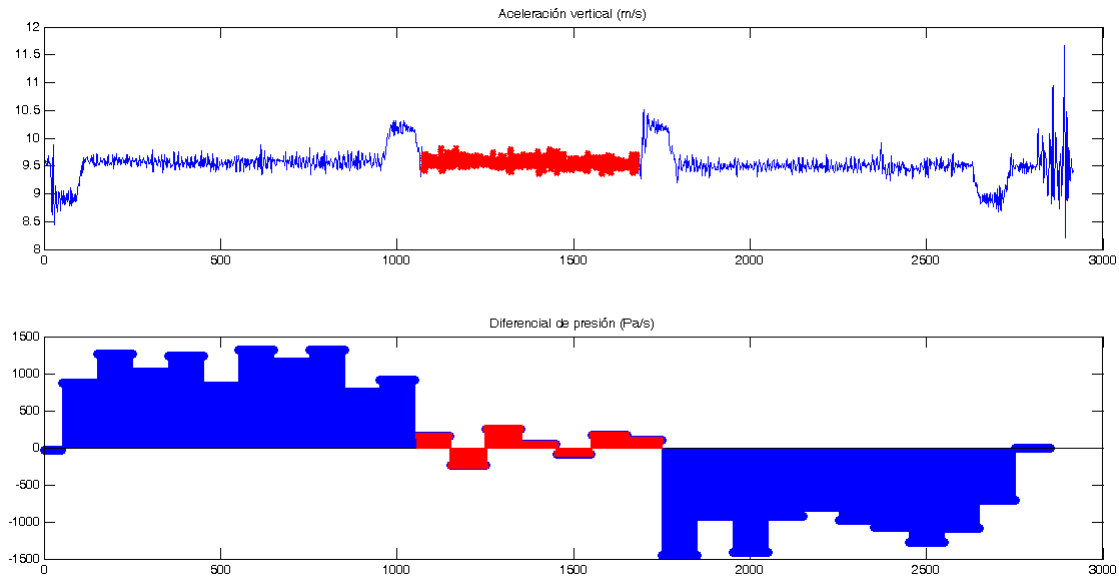
1.3. Características de ADL desde el Punto de Vista Inercial

A partir de la adquisición de señales inerciales es posible encontrar patrones que obedecen a las diferentes actividades realizadas por una persona, el número de sensores inerciales requeridos aumenta dependiendo de la complejidad de la actividad. Las actividades que se manejan dentro de este trabajo son detectadas con ayuda de un sensor ubicado en la cintura. A continuación se hará una descripción de la asociación entre las diferentes actividades y los patrones observados en las señales inerciales u de presión, cabe resaltar que el sensor es ubicado sobre la cintura del usuario.

la clasificación más sencilla de realizar se da entre la ausencia y presencia de movimiento. En el caso de mantenerse estático, se observa que la aceleración vertical muestra un valor constante igual al valor de la gravedad, mientras que los demás ejes muestran valores iguales a cero, también se puede observar que la varianza de la aceleración es aproximadamente cero. Cuando hay movimiento se observan picos en la señal de aceleración en todos los ejes y la varianza toma valores mayores a 1. En la figura 3 se puede observar la transición entre quietud y movimiento sobre la señal de aceleración vertical.

En caso de que no haya movimiento, pueden manejarse tres casos: la persona está “de pie”, “acostada” o “sentada”. Al usar un solo sensor sobre la cintura no es posible diferenciar entre estar de pie o sentado, para esto sería necesario un sensor adicional que se ubicara sobre el tobillo para analizar cambios de postura en cintura y pies. Para distinguir entre estar “de pie” y “acostado” se puede estimar la postura del usuario al observar en cual de los diferentes ejes se observa el valor de la aceleración, en otras palabras si el valor medio de la aceleración vertical está muy por debajo del valor de la gravedad (al menos la mitad), se considera que la persona se encuentra acostada. Un caso adicional que se considera es el movimiento en un ascensor, conociendo que se puede hacer una equivalencia entre la presión y la altura, se analizan los cambios de presión que son bastante pronunciados a diferencia del valor constante de presión que se da cuando no hay desplazamiento. Un ejemplo de los diferenciales de presión para movimiento en ascensor y permanecer quieto se puede ver en la imagen 4

Figura 4: Señales de aceleración vertical (arriba) y diferencial de presión (abajo) para movimiento en ascensor (azul) y permanecer quieto (rojo).



En caso de haber movimiento, se consideran tres casos: caminata normal, correr y desplazamiento en escaleras. Para diferenciar entre caminar y correr se puede recurrir a medir la velocidad de desplazamiento, energía o varianza de la señal de aceleración, pero dado que no todas las personas corren o caminan a la misma velocidad no puede hacerse la clasificación con un umbral simple. En la figura 5 se observa la magnitud de aceleración para “caminar” y “correr”, puede verse que la señal que corresponde a “correr” tiene picos más pronunciados que los presentes en la señal para “caminar”. La caminata en escaleras puede ser similar a los dos casos anteriores, la mayor diferencia está en que se avanza verticalmente, similar a lo que sucede en el caso del ascensor. El diferencial de presión puede ser útil para saber si se camina en escaleras o en línea recta, siendo este positivo o negativo dependiendo de si se bajan o suben escaleras respectivamente. En la figura 6 puede verse la comparación entre señales de aceleración y presión asociadas a caminar en línea recta y en escaleras.

1.4. Algoritmos de Detección de Caídas

Como se mencionó anteriormente, las señales obtenidas de los sensores inerciales pueden cambiar dependiendo de la ubicación del sensor, debido a esto, múltiples ubicaciones se han estudiado para identificar cual ofrece más información sobre las caídas. En la tabla 1, se muestra que la cintura es la ubicación más común, debido a que en este punto se encuentra el centro de gravedad del cuerpo. Otras ubicaciones para los sensores son: el cuello, las muñecas, el pecho, los muslos y los tobillos. Muchos investigadores han reportado que en el caso de los teléfonos inteligentes, la implementación puede ser bastante complicada debido a la variabilidad en la ubicación del dispositivo, teniendo en cuenta que las personas pueden llevar sus teléfonos en diferentes partes del cuerpo [19].

Teniendo en cuenta que según la organización mundial de la salud para el 2050 el 20 % de la población mundial será mayor a 60 años [40], es necesario tomar medidas para mejorar la calidad de vida de los adultos mayores. Para la detección de caídas se han implementado métodos basados en diferentes técnicas, que podrían ser clasificadas en dos grupos: técnicas analíticas y técnicas de aprendizaje de máquina [41]. Las técnicas analíticas corresponden a aquellas donde se definen modelos lógicos caracterizados principal-

Figura 5: Comparación de señales de aceleración vertical para caminar y correr

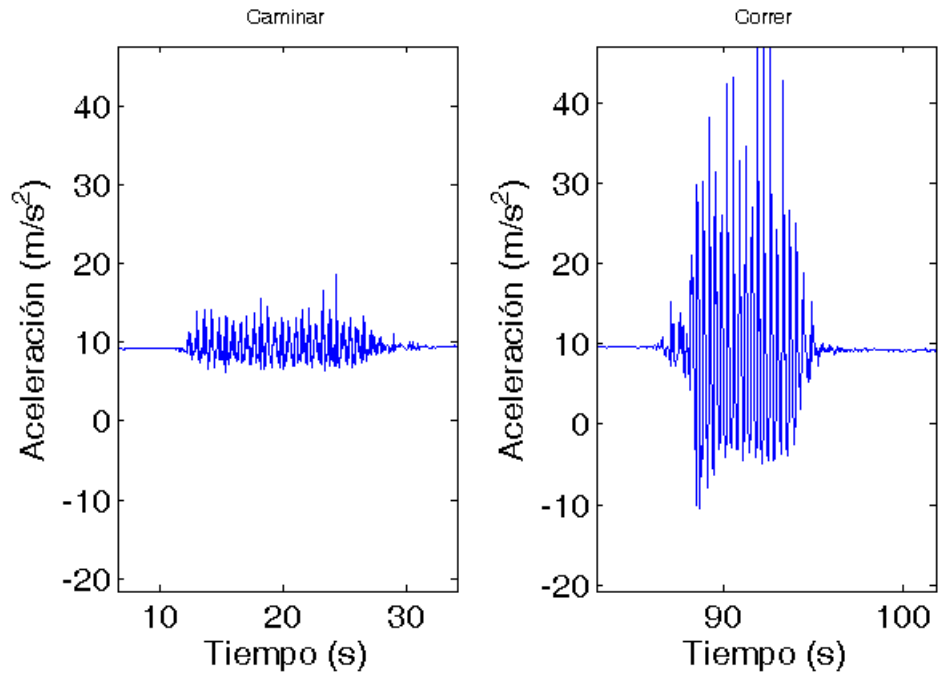
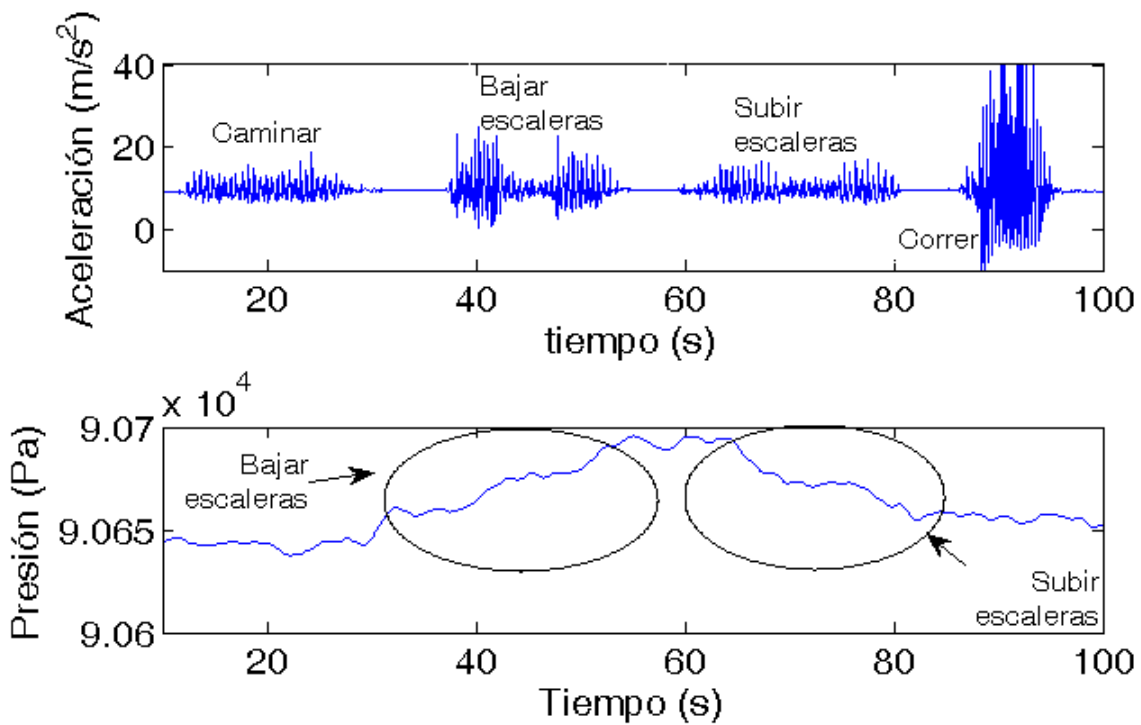


Figura 6: Comparación entre las señales de presión y aceleración vertical para caminata en escaleras, hacia adelante y correr



mente por umbrales, como es el caso del trabajo presentado en [51] donde se evalúan los momentos pre y post impacto de caídas usando la señal de un acelerómetro y un barómetro ubicados en la cintura del

Tabla 1: Ubicaciones comunes de IMUs sobre el cuerpo

Posición del sensor	Referencias
Cintura	[37] [64] [24] [22] [23] [52] [10] [15] [32] [57] [12] [47] [33]
Pecho	[18] [35] [20]
Tobillo	[3] [14]
Cabeza	[59] [32]

usuario, midiendo cambios en la velocidad vertical y diferencias de alturas con umbrales definidos para la detección del evento. Lo anterior se hace para ofrecer una mayor robustez al tener varias etapas de detección. Riva et al [50] presentan un método para evaluar el riesgo de caídas, esto usando mediciones no lineales: la proporción armónica, el índice de armonía, y entropía multiescala de la señal de aceleración del tronco, plantean como ventaja de este método el no requerir la detección del paso, argumentando que es una fuente potencial de errores si no se hace de forma precisa.

Dado que una detección con umbrales puede implicar mayor numero de falsos positivos, la integración de varios sensores puede ayudar a disminuir este problema. En [54] donde ubican una IMU en el tobillo y otra en la cintura, se busca tener una mejor estimación de los cambios de postura de la persona y así distinguir efectivamente entre una posible caída y una caída real, se emplean las mediciones de magnitud de aceleración y velocidad angular obteniendo valores de especificidad y sensibilidad por encima del 95 %.

Las técnicas que involucran aprendizaje de máquina son bastante útiles debido a que analizan datos a partir de una fase de entrenamiento sin involucrar un modelo previamente diseñado. Zhang et al. plantean un método basado en máquinas de soporte vectorial entrenado para reconocer positivos de caídas con datos provenientes de una unidad de medida inercial ubicada en la cintura. Teniendo en cuenta que al momento de una caída se dan cambios bruscos en la aceleración del cuerpo, se toman como características las variaciones de la aceleración en la sección transversa y en el eje vertical del cuerpo [64]. Luego, este trabajo es integrado en un teléfono inteligente y reforzado con el uso del discriminante de Fischer y el método del vecino más cercano, con esto se pretende eliminar falsos positivos que puedan aparecer cuando no hay un cambio completo de posición[63]. En [9] se muestra un algoritmo de detección de caídas usando lógica difusa, en dicho trabajo se analizan cambios bruscos en la orientación de la cintura comparando los resultados ubicando el sensor en la espalda baja o a un lado de la cintura, llegando a la conclusión de que hay una variación significativa de los resultados entre los resultados en estas dos ubicaciones. Otro trabajo basado en aprendizaje de máquina es el presentado en [58] donde su utilizan arboles de decisión para detectar caídas a partir de la señal de acelerómetros colocados en el pecho y el muslo, comparando su desempeño. El trabajo propone un algoritmo basado en micro anotación que es un análisis en el que solo una muestra puede ser clasificada como una caída, usando como características: mínimos y máximos de la velocidad, valor promedio de la aceleración, y estado de actividad.

El trabajo presentado por Lai et al. [32] muestra la implementación de un sistema enfocado al cuidado de adultos mayores, utilizan seis IMU ubicadas en diferentes partes del cuerpo para determinar el nivel de daño y la zona más afectada luego de una caída, se basa en la detección de picos en la señal de aceleración, combinado con el método de agrupación de datos (clusters) substractiva partiendo de la idea de que los movimientos se agrupan y si se sale del grupo indica un accidente.

Con el surgimiento de los teléfonos inteligentes y aprovechando las posibilidades de desarrollo que estos ofrecen, también se han implementado detectores de caídas en estos dispositivos. En [38] se implementa un detector de caídas en un teléfono inteligente ubicado en uno de los bolsillos delanteros del pantalón

del usuario, usan como método de detección la definición de umbrales en la magnitud de la aceleración, el cambio en la altitud, y la tasa de cambio de la velocidad angular, para detectar algunas actividades de rutina diaria (ADL) como: caminar, postura al sentarse y levantarse, recoger objetos y pasar de sentarse a estar acostado, se usaron redes neuronales para una detección robusta. Una forma de evitar que la ubicación del teléfono inteligente afecta la calidad de la detección es ubicar sensores externos en una ubicación fija, mientras que el procesamiento se realiza en el teléfono. Un ejemplo de lo mencionado es el trabajo presentado en [1], donde se utiliza un sensor que se comunica de forma inalámbrica con un teléfono equipado con *android*, se plantea un algoritmo con verificación posterior de actividades a una caída, la cual se detecta usando umbrales de 3g sobre la señal de aceleración. Con esta metodología se puede reducir la cantidad de falsos positivos pero aún es posible que se presenten confusiones con otras actividades.

También se han hecho trabajos comparativos como el presentado en [10] donde se implementan alrededor de 21 algoritmos y se presenta también un nuevo algoritmo que pone un umbral sobre la velocidad vertical del tronco, se compara su desempeño al ser probados con caídas y ADL, llegando a la conclusión de que un análisis de velocidad, impacto y postura es el más indicado para la detección de caídas. En [22], también se muestra un estudio comparativo de algoritmos en diferentes partes del cuerpo (cintura, cabeza y muñeca), llegando a la conclusión de que usando algoritmos basados en umbrales con los sensores sobre la cabeza o cintura, se puede llegar a especificidad y sensibilidad por encima del 96 %, pero se recomienda ubicar el sensor sobre la cintura para evitar limitar el movimiento del usuario al ubicarlo sobre la cabeza.

Uno de los principales inconvenientes con los algoritmos de detección de caídas recae en la dificultad para obtener información de caídas reales, dado que no es posible provocar estos eventos por motivos éticos y la conservación de la integridad de participantes en pruebas. En la mayoría de los trabajos realizados, las caídas son simuladas por personas que se lanzan sobre colchones y se recolectan los datos. Otra alternativa es la presentada por [29] donde se utiliza una barra para simular el cuerpo humano, se ubica el sensor a la altura de la cintura y se deja caer la barra para obtener los datos necesarios. En el trabajo realizado por Bagala et al. [5], se evalúan algunos de los algoritmos presentados en la literatura que se han probado con caídas simuladas. A partir de las pruebas realizadas con caídas reales, se llega a la conclusión de que la especificidad y sensibilidad de los algoritmos se ve afectada.

Cabe resaltar que en los trabajos presentados y en la literatura en general [13] [39] [19], los sensores son ubicados en su mayoría a la altura de la cintura, generalmente en la espalda. Lo anterior debido a que en la cintura se encuentra el centro de masa del cuerpo y los movimientos se concentran en dicho punto.

1.5. Algoritmos de Detección de Actividades Cotidianas (ADL)

Los sistemas de detección de actividades son utilizados en diversos campos, entre los que están el monitoreo de actividad física, cuidado de adultos mayores, monitoreo de equipos de emergencia, entre otros [36]. La detección de caminata se hace a través de la definición de umbrales en la señal de aceleración, dado que al existir movimiento esta empieza a mostrar picos característicos que indican pasos y por ende caminata [31]. La detección de ausencia de movimiento también puede medirse mediante umbrales; en [16] miden la desviación estándar de la aceleración y si esta no supera un umbral previamente definido entonces es posible afirmar que el usuario no se está desplazando.

Para detectar si el usuario está corriendo, un método adoptado comúnmente es el presentado por [4] donde se analiza la varianza de la señal de aceleración la cual se hace más grande al correr. En este trabajo se presenta también un método para la detección de caminatas hacia atrás basado en la correlación de las señales del acelerómetro, tomando como criterio de decisión el signo de dicha correlación (positiva indica caminata hacia atrás), los movimientos laterales se identifican con el análisis de la transformada de Fourier

de la aceleración y verificando si el valor máximo en la frecuencia de caminata se da en el eje que apunta hacia el costado del cuerpo.

En [34] se muestra un sistema de localización diseñado para equipos de emergencia, este sistema es complementado con un algoritmo de reconocimiento de actividades basado en la suma, el producto y la energía de la aceleración vertical y la aceleración en dirección del desplazamiento, logrando clasificar el subir y bajar escaleras, arrastrarse y caminatas laterales. Obtuvieron un error del 9.8 % y hacen la salvedad de que la calidad de estimación de longitud se reduce debido a los movimientos realizados por los equipos de emergencia.

Las técnicas de aprendizaje de máquina han sido utilizadas ampliamente en la detección de actividades, resaltan los modelos de Markov que son de gran utilidad al analizar datos secuenciales [48]. Yusuf et al. [56] presentan un sistema de reconocimiento de actividades y *dead reckoning* también para equipos de emergencia capaz de reconocer las mismas actividades que en [34] pero ubicando el sensor en el pie del usuario y analizando el comportamiento de la señal de aceleración usando un modelo oculto de Markov (HMM). Otro trabajo en el que se utilizan HMM es el presentado en [44] donde ubicando una IMU en el pecho del usuario son capaces de detectar caminata, subida y bajada de escaleras, quietud o carrera. Este sistema se diseña para luego ser integrado con un sistema de visión artificial.

Las máquinas de soporte vectorial (SVM) también han sido utilizadas ampliamente en el campo de la detección de actividades, en [58] las SVM son combinadas con HMM para lograr una estimación robusta de actividades como: caídas, saltos, caminata lenta y rápida, correr, entre otras; primero el HMM es utilizado para generar los puntos a clasificar sobre un hiperplano, el cual es creado utilizando SVM, con esto se tiene una clasificación en dos etapas con efectividad de alrededor del 100 %. EN [3] se realiza la implementación de un sistema basado en SVM para la detección de caminata en escaleras y terreno plano usando dos características del nuevo espacio obtenido a partir de un análisis PCA, en este caso el sensor se ubica sobre el tobillo del usuario logrando una precisión del 98 % en la detección, se deja abierta la posibilidad de mejorar el clasificador al agregar una mayor cantidad de actividades.

Cabe resaltar que los HMM y SVM no son los únicos métodos utilizados para la detección de actividades, otras técnicas como los clasificadores de Bayes [6], las redes neuronales [55] y algunos no tan comunes como el *adaptive boosting* [28] también tienen validez y han sido utilizadas para dicho fin.

Una práctica recurrente es la utilización de varios sensores para lograr una mayor robustez del algoritmo, por ejemplo, en [42] se utilizan acelerómetro y barómetro para detectar el subir y bajar escaleras, subir y bajar en un elevador, caminata y ausencia de movimiento haciendo análisis en frecuencia de la señal de aceleración y de los cambios de presión del aire, además también calculan la velocidad de caminata, esto enfocado al cálculo del gasto energético del usuario.

1.6. Resumen

En esta sección se presentó la revisión del estado del arte correspondiente a algoritmos de detección de ADL y caídas, además se explican en detalle las características de cada una de las actividades a detectar, así como las fases de una caída. La descripción se realiza basándose en señales inerciales, explicando el comportamiento de las señales ante cada una de las actividades estudiadas.

A partir del estudio de las actividades desde el punto de vista inercial, se facilita el proceso de selección de características, debido a que se tiene un mayor entendimiento de los fenómenos que se están estudiando.

Capítulo 2

Creación de Base de Datos de Actividades Cotidianas y Caídas

2.1. Introducción

Para el desarrollo de un algoritmo de reconocimiento de patrones es necesario contar con una base de datos para el entrenamiento del mismo. Existen diferentes bases de datos donde se encuentran señales inerciales tomadas de voluntarios realizando diferentes actividades, como la *UCI machine learning database* [49] o la *URfall Detection Dataset* creada en la universidad de Rzeszow [30]. Como se ha mencionado ya, la ubicación del sensor afecta considerablemente la forma de onda de las señales inerciales, por lo que para este proyecto se creó una base de datos propia buscando obtener total control e información sobre los datos.

2.2. Actividades a Detectar

Teniendo en cuenta las diferentes actividades que realiza la población objetivo (adultos mayores), se determinó que es necesario definir como clases las actividades mostradas en la tabla 2. Con estas actividades será posible estimar que se encontraba haciendo el usuario antes y después de una eventual caída.

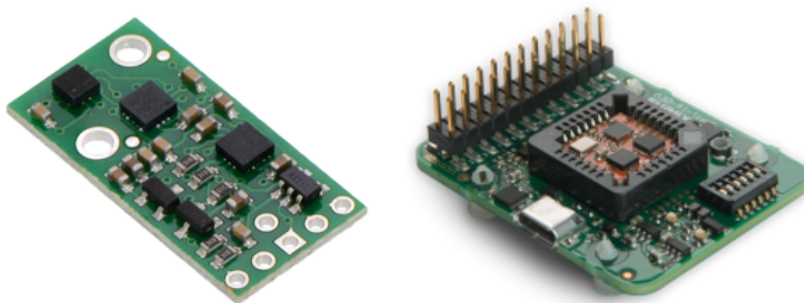
Tabla 2: Actividades detectables para el algoritmo

Clase	Actividad
Clase A	Caminata normal
Clase B	Correr
Clase C	Caminata en escaleras
Clase D	Caída
Clase E	Permanecer de pie
Clase F	Permanecer acostado

2.3. Equipos y Montaje para la Adquisición de Datos

2.3.1. Dispositivos y Sensores. Los equipos utilizados para la toma de datos fueron proporcionados por el proyecto “análisis de desorientación espacial. Caso de Estudio: adulto mayor” (ADEAM) cofinanciado por Colciencias y la Universidad Industrial de Santander.

Figura 7: AltIMU 10 v4 (izquierda) e IMU xsens MTi-1 (derecha)



Sensores Inerciales Magnetómetros y Barómetros

Las unidades de medida inercial (*IMU*, por sus siglas en inglés) son dispositivos compuestos por sensores inerciales como lo son: acelerómetros, giroscopios y magnetómetros. Actualmente pueden incluir barómetros para hacer un análisis más completo. Los acelerómetros son sensores dedicados a medir la aceleración de un cuerpo a través del efecto de fuerzas sobre una masa, sistema que ha sido reemplazado en los sensores miniaturizados por un objeto de silicio muy pequeño y se mide su interacción con campos magnéticos. Los giroscopios tienen como función medir la velocidad angular de un cuerpo en función del llamado efecto coriolis, con base a estas mediciones es posible calcular los cambios de orientación de un objeto. Los magnetómetros miden los campos magnéticos que afectan directamente a un cuerpo, el principal inconveniente con estos dispositivos es que son bastante sensibles a perturbaciones exteriores.

Los barómetros son sensores dedicados a medir la presión atmosférica con respecto al nivel del mar, estos son utilizados también para medir altitud debido a la relación que existe entre los cambios de presión y de altitud [42], han sido integrados en dispositivos para tareas como la medición del clima y también en sistemas de reconocimiento de actividades.

AltIMU 10 v4

La altIMU 10 v4 es una unidad de medida inercial que comprende acelerómetros y magnetómetros en un chip (LSM303D), giroscopios (L3GD20H), y barómetro digital (LPS25G). Esta IMU puede comunicarse mediante los protocolos I2C y SPI permitiendo acceso independiente a las medidas entregadas por los 10 sensores incluidos (Acelerómetros, giroscopios y magnetómetros son triaxiales). En la tabla 3 pueden verse las características principales de esta IMU y el dispositivo puede verse en la imagen 7.

Tabla 3: Especificaciones de los sensores contenidos en la altIMU 10 v4

AltIMU	Rango de medida	Frecuencia de muestreo
Acelerómetros	+/- 16g	1.6 KHz
Giroscopios	2000 °/s	800 Hz
Magnetómetros	12 G	100 Hz
Barómetro	26 kPa - 126 kPa	25 Hz

IMU Xsens MTi-1

EL kit de desarrollo que incluye la IMU MTi-1 diseñado por *Xsens* (ver imagen 7), permite tomar datos de una manera relativamente sencilla dado que incluye el *software MT manager* de *Xsens* para la toma de datos y configuración del dispositivo. A diferencia de la altIMU, esta IMU entrega los datos prefiltrados, y se comunica mediante el protocolo UART. Para tomar los datos usando la *Beaglebone black* se uso una versión modificada del controlador proporcionado por ROS.org. Algunas de las características de esta IMU se muestran en la tabla 4.

Tabla 4: Especificaciones de la IMU Xsens MTi-1

Xsens	Rango de medida	Frecuencia de muestreo
Acelerómetros	+/- 16g	Hasta 1 KHz
Giroscopios	+/- 2000 °/s	Hasta 1 KHz
Magnetómetros	+/- 0.8 G	Hasta 100 Hz
Roll, Pitch, Yaw	+/- 180 °	Hasta 0-100 Hz

BeagleBone Black (BBB)

La *BeagleBone Black* es una tarjeta de desarrollo de bajo presupuesto desarrollada por BeagleBoard.org, Esta tarjeta de desarrollo cuenta con dos puertos I2C, permitiendo la conexión de la altIMU, y también conexión UART para la IMU Xsens. Además, cuenta con una memoria flash de 4GB, suficiente para almacenar el código necesario para la implementación.

En cuanto al procesador cuenta con un Sitara XAM3359AZCZ100 Cortex A8 ARM de Texas Instruments. La velocidad de procesamiento de este procesador es de 1 GHz, con lo que se espera no tener inconvenientes en la implementación en tiempo real. La BBB cuenta con una entrada de voltaje de 5V y los pines de salida de 5V y 3.3V, además los pines I/O funcionan con un voltaje de 3.3V.

La BeagleBone black cuenta con un sistema operativo basado en Debian, usando este sistema es posible compilar y ejecutar códigos descritos en *Python*. Lo anterior, facilita la implementación del código y permite trabajar con las librerías para procesamiento de datos y aprendizaje de máquina desarrolladas para *Python*. El resumen de algunas características de la BBB puede verse en la tabla 5.

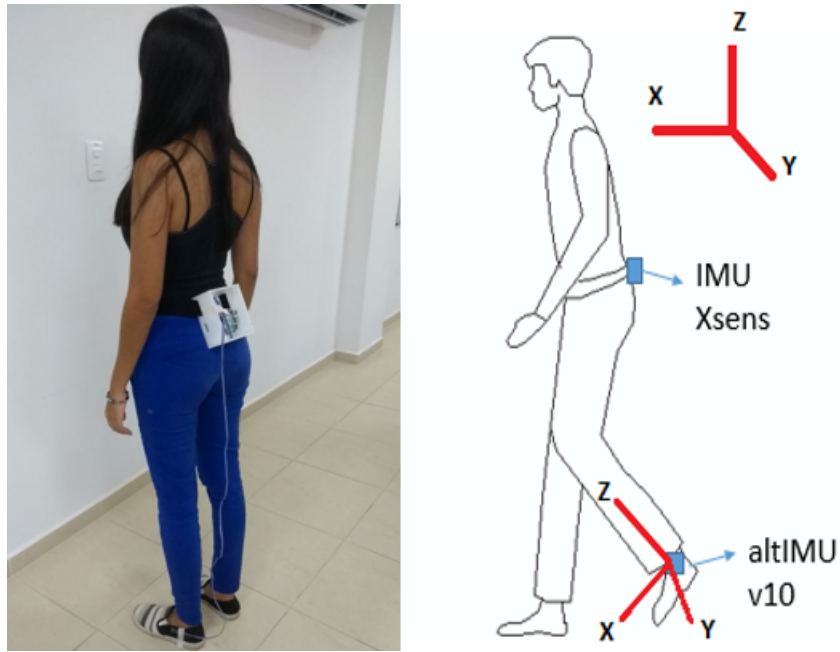
Tabla 5: Características de la BeagleBone Black

	Característica
Procesador	Sitara XAM3359AZCZ100, 1GHz
Memoria SDRAM	512MB DDR3L 800MHz
Memoria FLASH	4GB, 8bit MMC embebido
Fuente de alimentación	conector miniUSB, entrada 5VDC via conector de expansión
PCB	3.4"x 2.1"
Ethernet	10/100, RJ45

2.3.2. Ubicación de Sensores y Señales Adquiridas. Para la toma de datos durante pruebas experimentales, la configuración de sensores utilizada fue la mostrada en la figura 8. La IMU Xsens es colocada

sobre la cintura del usuario mientras que la altIMU es colocada en el tobillo, esto se hace para poder comparar las señales obtenidas en dichas ubicaciones y determinar así donde se obtiene la mayor separabilidad para las diferentes actividades a clasificar.

Figura 8: Montaje de sensores sobre el usuario



Para realizar un análisis completo, durante las pruebas se adquieren y se guardan en formato .txt todas las señales que pueden ser entregadas por las dos IMU, las cuales pueden apreciarse en la tabla 6.

Tabla 6: Señales tomadas de los sensores durante las pruebas experimentales

Xsens	altIMU
a_x, a_y, a_z (m/s^2)	a_x, a_y, a_z (m/s^2)
w_x, w_y, w_z (rad/s)	w_x, w_y, w_z (rad/s)
m_x, m_y, m_z (<i>Gauss</i>)	m_x, m_y, m_z (<i>Gauss</i>)
Roll, Pitch, Yaw	P (Pa)

2.4. Protocolo de Toma de Datos

El siguiente protocolo de experimentación se siguió para crear una base de datos para la detección de caídas y actividades cotidianas usando sensores inerciales. Las pruebas se realizaron con adultos jóvenes con edades entre 20 y 30 años, los participantes portaron los sensores altIMU y Xsens en el tobillo y cintura respectivamente como se muestra en la figura 8.

Durante estas pruebas, se tomaron los datos necesarios para posteriormente analizar las señales obtenidas y encontrar las características que permiten clasificar las diferentes actividades. Cabe aclarar que para la realización de estas pruebas se contó con el aval del comité de ética de la Universidad Industrial de Santander.

2.4.1. Datos para Actividades Cotidianas. El protocolo que se siguió para la recolección de estos datos fue el siguiente:

- ❖ **Caminata en línea recta I:** el usuario dio 30 pasos hacia adelante a la velocidad que consideró normal en línea recta, luego se detuvo hasta que se le indicó que ya podía moverse.
- ❖ **Caminata en línea recta II:** el usuario dio 30 pasos hacia adelante a velocidad de trote en línea recta, luego se detuvo hasta que se le indicó que ya puede moverse.
- ❖ **Caminata en escaleras:** Para esta prueba el usuario subió 15 escalones, luego se detuvo hasta que se le indicó que bajara los mismos 15 escalones. Esta prueba se repitió un total de 4 veces.
- ❖ **Transición sentarse/pararse:** Los participantes se sentaron en una silla, debían esperar alrededor de 3 segundos para luego ponerse de pie, este movimiento se repite 10 veces.

2.4.2. Datos de Entrenamiento para la Detección de Caídas. Para obtener datos de caídas, inicialmente se simularon caídas con los sensores sujetos sobre una barra a la altura de la cintura, como se muestra en [29] y puede observarse en la figura 9. Luego, se realizó la toma de datos ubicando los sensores sobre la cintura de una persona. Posteriormente, se realiza una comparación entre las señales para verificar la similitud entre las señales y entrenar con ambos conjuntos de datos.

Figura 9: Ubicación de la IMU Xsens a la altura de la cintura sobre una barra para pruebas de adquisición de datos

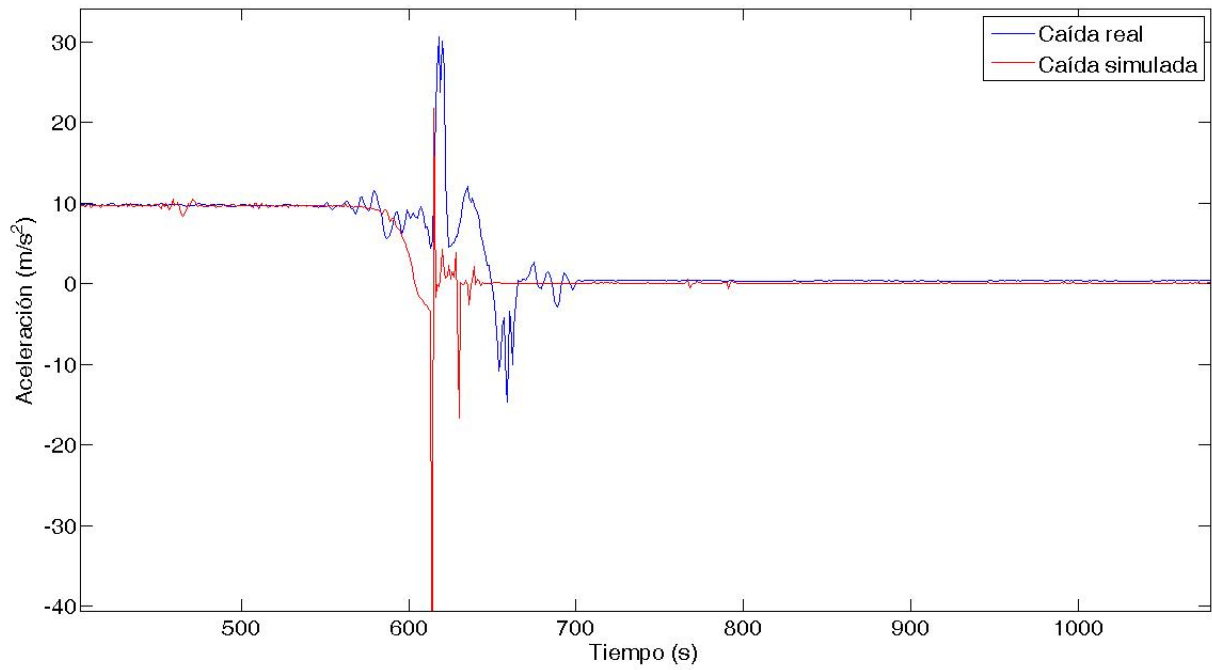


Las pruebas se realizaron con caídas aisladas, es decir, sin combinar con otras actividades. En la figura 10 puede verse una comparación entre las señales de aceleración y velocidad angular en una caída, tanto simuladas sobre la barra como para las simuladas con personas.

2.5. Resumen

En este capítulo se comenta todo lo relacionado con la creación de la base de datos de ADL y caídas, se da una descripción de los sensores utilizados y para la toma de datos se diseñó un protocolo de experi-

Figura 10: Señal de aceleración de una caída simulada ubicando sensores sobre una barra (roja) y simulada con una persona (azul)



mentación donde se involucran las diferentes variables como la ubicación de los sensores y la variación en los patrones de movimiento de diferentes usuarios.

Capítulo 3

Selección de Características y Diseño del Algoritmo de Detección de Caídas y ADL

3.1. Introducción

Para diseñar y entrenar un algoritmo de reconocimiento de patrones es necesario conocer el fenómeno que se enfrenta, esto permite ser más efectivo a la hora de seleccionar las características y el flujo de datos necesario para que funcione adecuadamente.

A partir de la base de datos creada se realiza la selección de características utilizando MATLAB como herramienta de simulación, se obtienen las características que ofrecen la mejor separabilidad entre las diferentes actividades. Este grupo de características sirven posteriormente para entrenar los clasificadores SVM necesarios para la implementación del algoritmo de reconocimiento de patrones a implementar.

Adicionalmente, luego de la revisión del comportamiento de las señales para cada actividad, se define el flujo de datos del algoritmo de reconocimiento de actividades y caídas, que es el implementado posteriormente en tiempo real.

3.2. Selección de Características

Siguiendo lo estudiado en la literatura [27] y basándose en la extracción de características realizada dentro del proyecto ADEAM, se extrajeron alrededor de 140 características que son utilizadas para la detección de actividades cotidianas, estas pueden consultarse en el anexo A. Las características son calculadas cada 2 segundos (lo que equivale a 100 muestras de cada señal) debido a que los cambios en características como el diferencial de presión son bastante lentos, y es necesario tomar un número considerable de muestras para obtener medidas que sean útiles. El algoritmo fue planteado originalmente para clasificar entre las actividades mostradas en la tabla 2, dicho algoritmo contiene dos máquinas de soporte vectorial para clasificación de actividades (Caminar - correr, caídas - otras actividades).

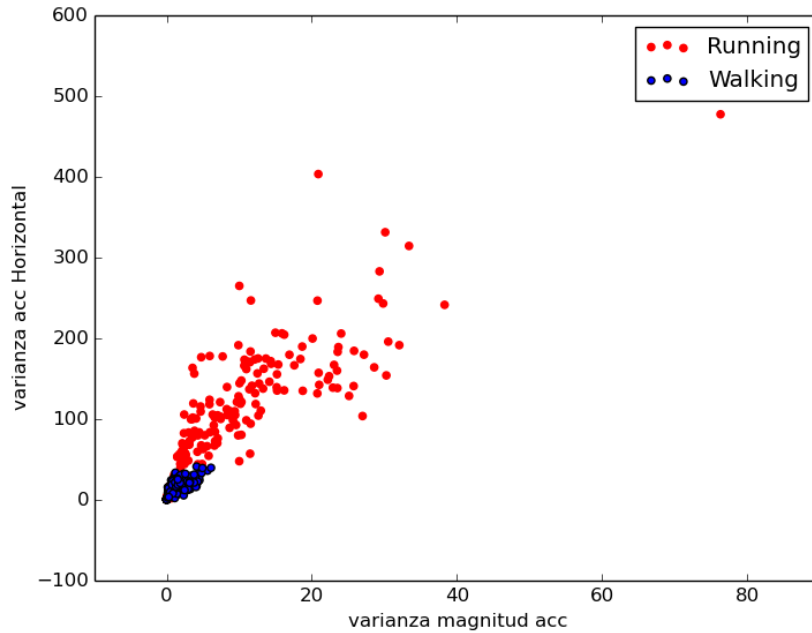
Para seleccionar las características que ofrecen la mejor separabilidad para estas actividades, se realiza una selección hacia adelante (*forward selection*) con ayuda de la función *sequentialfs* de MATLAB. Este método consiste en seleccionar características de forma secuencial hasta que se encuentra el conjunto con el mejor desempeño en la detección, cabe resaltar que se trató de mantener el conjunto de características más pequeño posible para cada grupo de actividades, las características seleccionadas pueden verse en la tabla 7 y la distribución de los datos para caminar-correr y caídas-otros pueden verse en las gráficas 11 y 12 respectivamente.

En el caso de la detección de desplazamiento en un ascensor, se usó el diferencial de presión el cual aumenta o disminuye de manera significativa cuando el usuario se mueve entre pisos de un edificio. Para

Tabla 7: Características obtenidas luego del proceso de selección usando MATLAB

Caminar - correr	Caídas - otros
Varianza de la magnitud de la aceleración (m/s^2)	Mínimo velocidad angular (rad/s)
Varianza de la aceleración horizontal (m/s^2)	Máximo de la velocidad angular (rad/s)
	Varianza de la magnitud de la aceleración (m/s^2)

Figura 11: Distribución de datos en el espacio de características para las actividades caminar y correr



la detección en escaleras se trata de usar también el diferencial de presión pero los cambios son pequeños y la señal fluctúa considerablemente provocando que no sea robusta la detección, sin embargo un posterior procesamiento puede mejorar la clasificación.

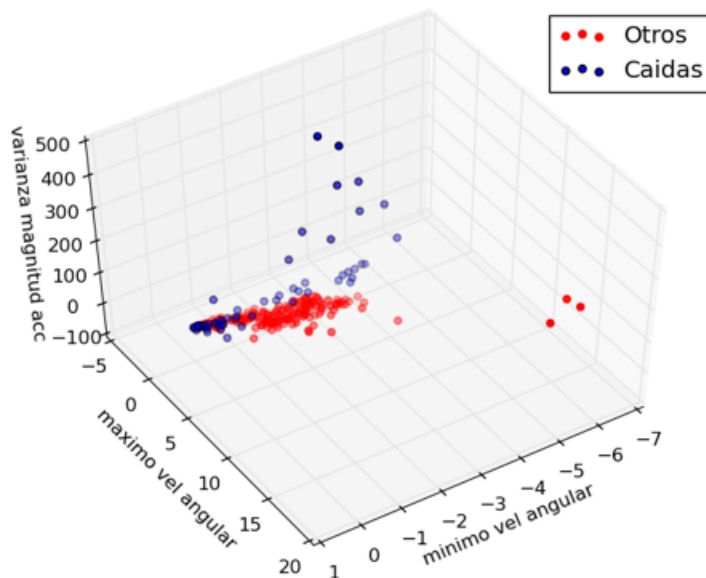
3.3. Diseño del Algoritmo

Para definir el flujo de datos del algoritmo de detección de caídas y ADL fue tomada en cuenta la dificultad para separar las diferentes clases, es decir, se clasifican primero las actividades que son fácilmente detectables a través de umbrales.

3.3.1. Adquisición y Pre-procesamiento de Datos. Inicialmente es necesario tomar los datos que se requieren para poder extraer las características seleccionadas. A diferencia de las pruebas realizadas para la creación de la base de datos, se tomarán solamente señales de aceleración, velocidad angular y presión, dado que estas señales son las necesarias para el cálculo de las características ya definidas.

En cuanto al preprocesamiento, es necesario filtrar la señal de presión debido a que esta está contaminada con ruido y puede llevar a errores en la clasificación. Los datos se almacenan para llevar un registro en

Figura 12: Distribución de datos en el espacio de características para caídas (azul) y Otras actividades (rojo).



caso de que sea necesario hacer alguna verificación posterior.

3.3.2. Clasificación de Actividades sin Desplazamiento. La primera clasificación se hace entre el estar quieto de pie, acostado y desplazarse en un ascensor, en estos casos se considera que el usuario se queda quieto, lo que conlleva a que no haya cambios significativos en la varianza, un valor de varianza menor a 1 obedece a este caso. Si se presenta una reducción de al menos la mitad en el valor medio de la aceleración vertical (el valor de la gravedad no se refleja en este eje) se estima que el usuario se encuentra acostado. Luego, si no se da una de estas dos situaciones se verifica el diferencial de presión, con lo que se puede determinar si el usuario está desplazándose en un ascensor, un cambio de al menos 7.5 Pa/s da un indicio de esta situación.

3.3.3. Clasificación de Actividades con Desplazamiento. Una vez se detecta movimiento, la clasificación se realiza entre las actividades “caminar” y “correr”. Para hacer esta separación se recurre a un clasificador SVM, dado que no se puede definir un umbral que sea adecuado para cualquier usuario, en este caso un método robusto como SVM ayuda a una detección mucho más precisa.

Se dedujo a partir de pruebas preliminares que la caminata en escaleras se confunde con correr, es por esto que si se registra como actividad detectada correr, se verifica el cambio en el diferencial de presión para estimar si hay un cambio de altura, similar a como se da con el desplazamiento en un ascensor, en el caso de la caminata en escaleras el umbral es de alrededor de 5 Pa en cada iteración del algoritmo.

3.3.4. Detección de Caídas. Por último, para detectar caídas se utilizan dos métodos complementarios: el primero es un clasificador de SVM que ayuda a detectar caídas aisladas, y para el caso de caídas combinadas con otras actividades se observó que existe una confusión con las actividades “correr” y “caminata en escaleras”, sabiendo que al darse una caída existe una transición entre una posición erguida y luego se yace acostado en el suelo, se agregó verificación de valor medio de la aceleración vertical cada vez que se

detecta una de estas dos clases, lo que evita la confusión. Inicialmente, se detecta una “posible caída”, para luego entrar en un proceso de verificación de 10 segundos, luego de esto se verifica la actividad que realiza el usuario. Si la persona se encuentra acostada o de pie e inmóvil, se dice que el usuario efectivamente cayó. Adicionalmente, si se detecta una “posible caída”, se verifica que la actividad anterior sea diferente a “acostado”, lo cual evita que se den falsos positivos en transiciones de “acostado” a “de pie”.

El flujo de datos del algoritmo se puede observar en las figuras 13 y 14, donde se muestran resaltados en rojo, los instantes en los que se diagnostica cada actividad, y en azul los bloques en los que se activa o se desactiva la bandera correspondiente a “posible caída”.

Figura 13: Primera parte del algoritmo de reconocimiento de actividades.

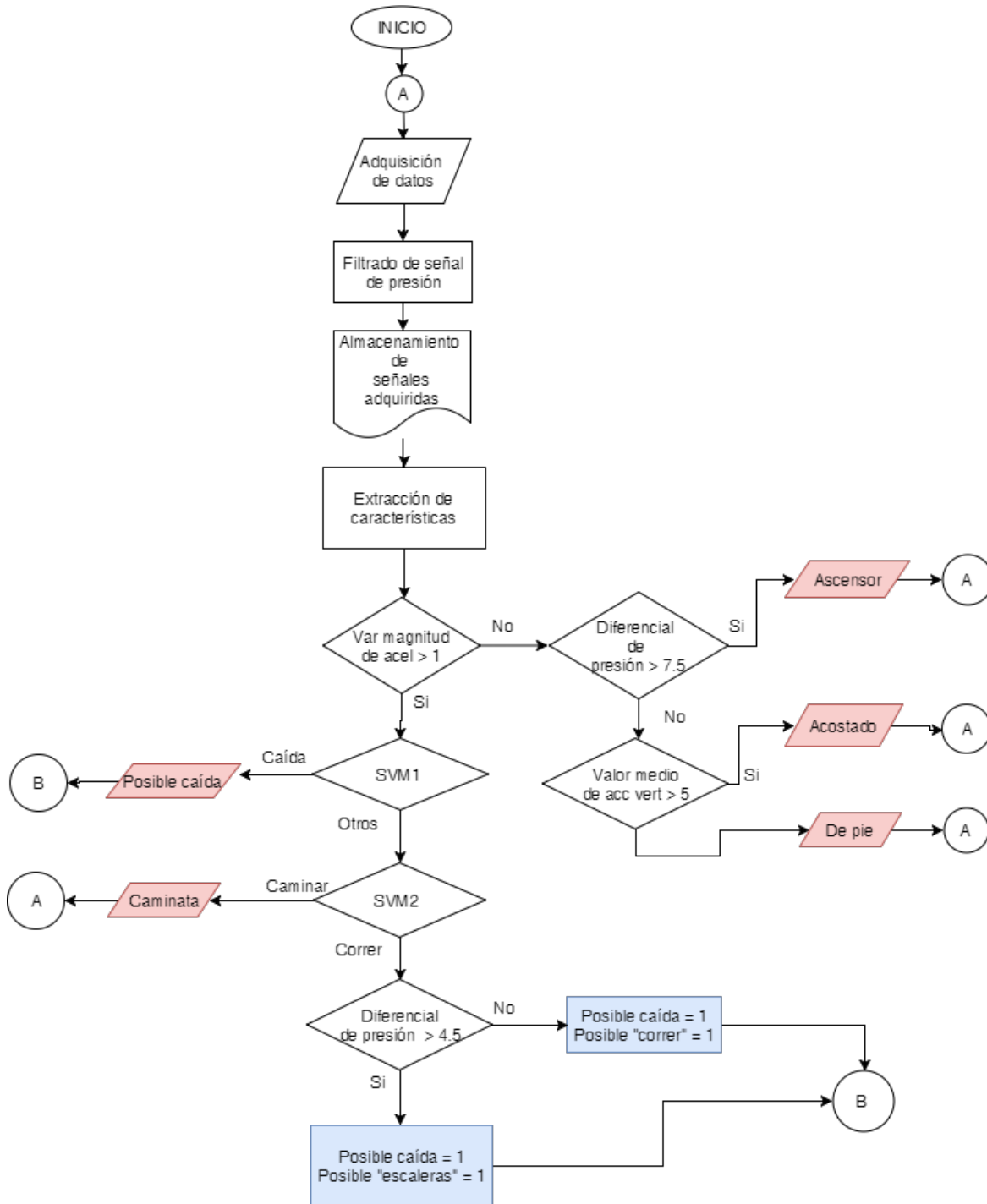
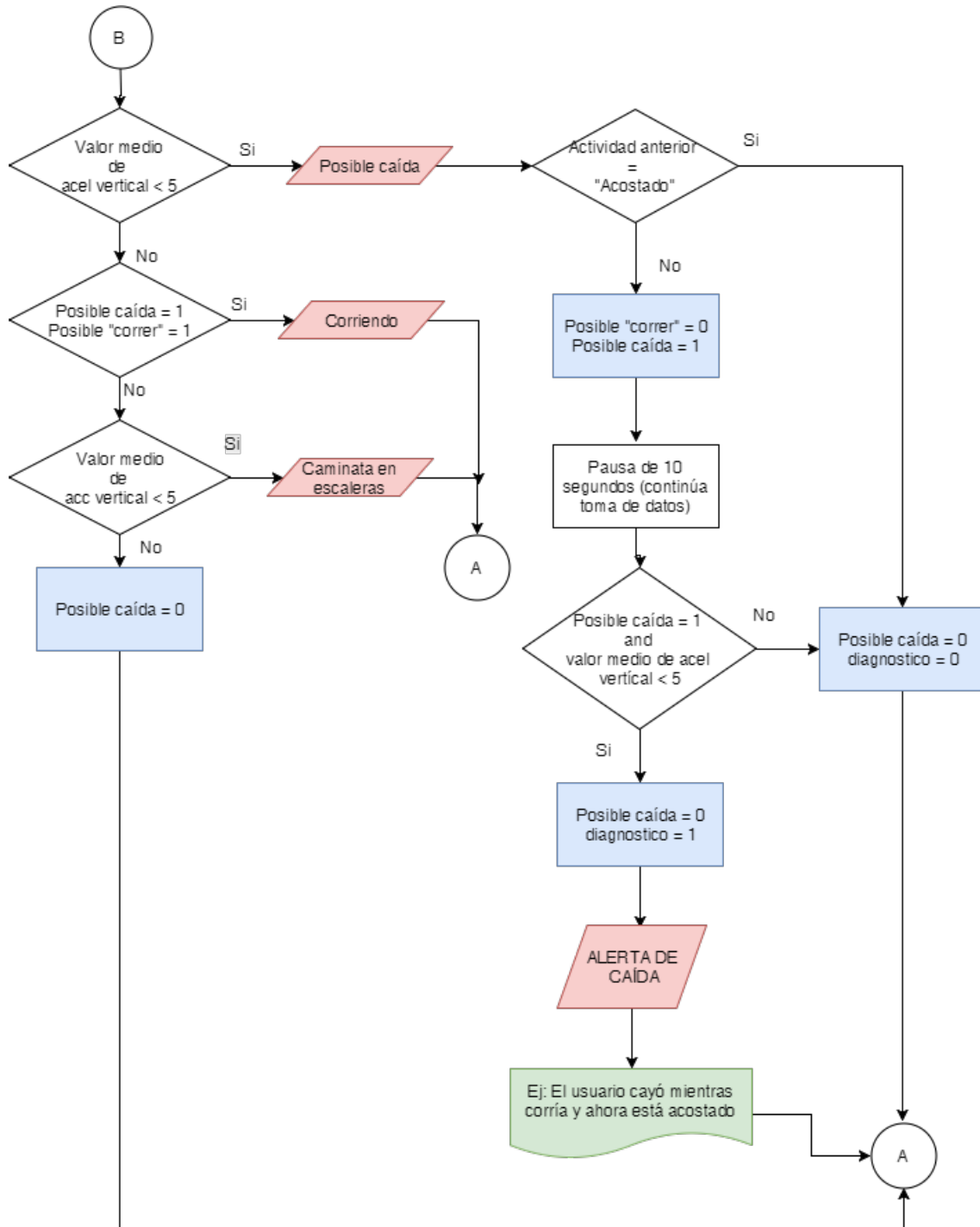


Figura 14: Segunda parte del algoritmo de reconocimiento de actividades.



3.4. Resumen

En esta sección se presentó el proceso de selección de características y diseño del algoritmo. Una vez se tuvo el conjunto de características para la detección de cada una de las actividades fue posible definir el flujo de datos y con base a esto diseñar el algoritmo de manera que se aproveche la relación y las diferencias entre las clases.

Una vez definida la estructura inicial del algoritmo se procede a realizar la implementación en MATLAB para comprobar su desempeño ante actividades no aisladas, es decir, señales que corresponden a más de una actividad.

Capítulo 4

Implementación de Algoritmos de Detección de ADL y Caídas

4.1. Introducción

Para desarrollar un clasificador en tiempo real, debe realizarse previamente la implementación *offline* del mismo. Esto para entrenarlo correctamente, calcular su desempeño y tener un estimado del posterior funcionamiento en tiempo real.

Es necesario comprobar el desempeño de los clasificadores SVM por separado y del algoritmo final, debido a que como se verá más adelante, se requiere de ajustes para que el funcionamiento del algoritmo completo sea el esperado.

En este capítulo se muestran las implementaciones *offline* y en tiempo real del algoritmo de detección de actividades y caídas, así como el análisis de desempeño en cada caso y una comparación de los mismos.

4.2. Algoritmo de Detección de Actividades y Caídas Offline

4.2.1. Clasificadores SVM. Se diseñaron dos clasificadores SVM usando MATLAB, el primero fue diseñado para separar las actividades caminar y correr. Este clasificador fue implementado usando kernel RBF (Gaussiano) con un margen de 0.6, debido a que se ajustó mejor a la distribución de los datos. Para verificar los parámetros de desempeño del clasificador se recurre a la implementación de validación cruzada de 10 grupos, a partir de la cual se obtiene la matriz de confusión y se extraen los parámetros de desempeño que se observan en la tabla 8. La matriz de confusión asociada a este clasificador es la mostrada a continuación:

$$\begin{pmatrix} 192 & 15 \\ 14 & 184 \end{pmatrix}$$

Tabla 8: Parámetros de desempeño para el clasificador SVM “caminar - correr”

Parámetro	Valor (%)
Precisión	90.6
Sensibilidad	92.75
Especificidad	92.92

Para el caso de la detección de caídas y su separación de otras actividades se implementa una SVM con tres características identificadas previamente en la sección de selección de características. Este clasi-

cador también utiliza el kernel RBF y se realiza una validación cruzada de 10 grupos para comprobar el desempeño del clasificador. Los parámetros de desempeño pueden observarse en la tabla 9 y la matriz de confusión se muestra a continuación:

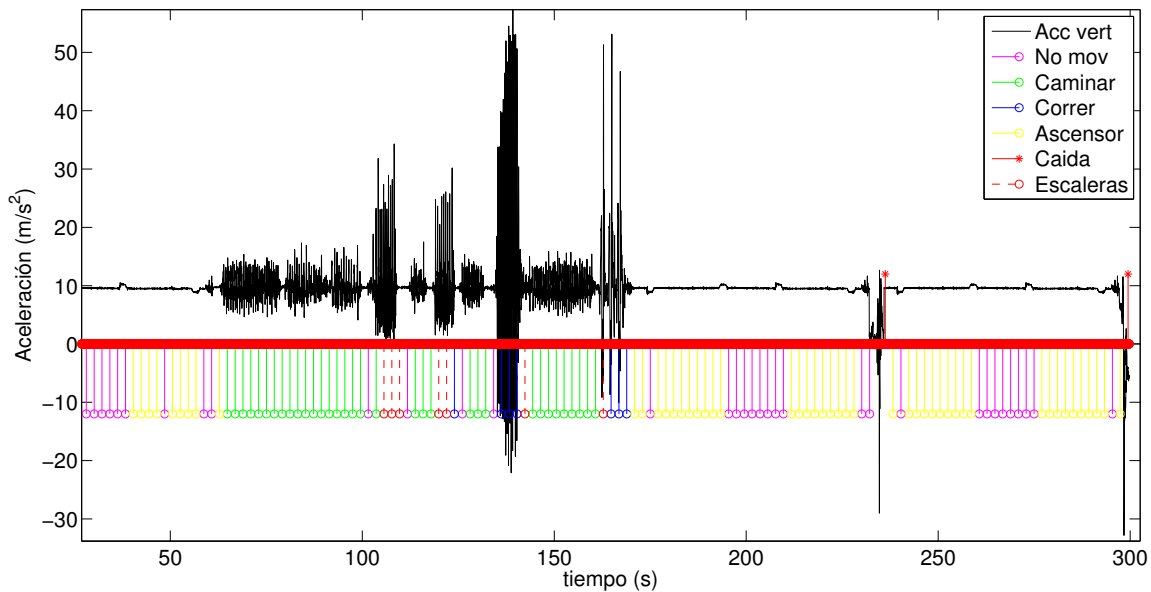
$$\begin{pmatrix} 48 & 14 \\ 41 & 153 \end{pmatrix}$$

Tabla 9: Parámetros de desempeño para el clasificador SVM “caídas - otros”

Parámetro	Valor (%)
Precisión	89
Sensibilidad	77
Especificidad	78

4.2.2. Validación del Algoritmo. Para validar el algoritmo completo de manera offline, se ingresaron al algoritmo diferentes rutinas de movimientos realizados previamente por voluntarios, donde se combinan las diferentes actividades que el algoritmo es capaz de identificar. Un ejemplo de estas rutinas puede verse en la figura 15.

Figura 15: Ejemplo de detección de actividades. Los marcadores indican la actividad detectada en cada iteración



El desempeño del algoritmo es evaluado teniendo en cuenta la tasa de acierto y errores en cada una de las pruebas realizadas, se toma como medida de desempeño el error promedio de todas las pruebas realizadas. El algoritmo mostró un desempeño del 90,8 %, teniendo en cuenta que la ventana de muestreo puede implicar transición entre actividades. En la tabla 10 se muestra el error obtenido para cada actividad y la tasa de error total para el algoritmo final.

Tabla 10: Parámetros de desempeño para el algoritmo completo de detección de caídas y ADL (offline)

Actividad	Error (%)
Acostado	0
Ascensor	0
Caminar	3.4
Correr	8
Caídas	0
De pie	0
Escaleras	15
Algoritmo final	9.2

4.3. Implementación en Tiempo Real del Algoritmo de Detección de Caídas y ADL

4.3.1. Selección de Python para la Implementación del Algoritmo. El lenguaje seleccionado para realizar la implementación es *Python 2.7*. La principal ventaja por la cual se seleccionó este lenguaje para la implementación es la existencia de la librería *Scikit Learn* [46], esta incluye un paquete completo para la implementación y validación de algoritmo relacionados con aprendizaje de máquina, de forma similar a como se maneja en MATLAB. Además, al ser Python orientado a objetos y contar con librerías para procesamiento de datos como *numpy*, permite implementar de forma más rápida y sencilla.

La mayor desventaja de la utilización de Python es la velocidad de procesamiento relacionada con la toma de datos. Dado que Python es un lenguaje de más alto nivel en comparación a C y C++, puede llegar a ser aproximadamente 10 veces más lento. Esta velocidad puede ser de gran importancia para algunas aplicaciones, en este caso, la toma de datos de los 10 sensores se limita a 50 Hz, sabiendo por pruebas realizadas previamente que a esta frecuencia se garantiza que no hay pérdidas de datos en la adquisición.

4.3.2. Implementación del Algoritmo. En las pruebas de toma de datos, se utilizaban los diez sensores disponibles dado que era necesario analizar todas las señales posibles en busca de las características para realizar la clasificación. Para el algoritmo en tiempo real solo se consideran medidas de aceleración y velocidad angular de la *IMU Xsens*, mientras que de la *altIMU* se toman los datos de presión. Es posible utilizar los acelerómetros y giroscopios de la *altIMU* y así realizar la implementación con un solo sensor, pero se optó por utilizar las medidas de la *IMU Xsens* debido al preprocesamiento que realiza la unidad.

La toma de datos se sigue realizando a 50 Hz, dado que los datos utilizados para el entrenamiento del algoritmo fueron tomados a esta frecuencia por las limitantes de velocidad relacionadas con el desempeño de Python. Según lo expuesto en [17][31], las señales relacionadas con el movimiento de una persona se encuentran entre los 0.5 y 20 Hz, por lo que muestrear a 50 Hz permite conservar los componentes de frecuencia necesarios cumpliendo el criterio de Nyquist.

Como se mencionó anteriormente la librería *scikit learn* permite la implementación de algoritmos de aprendizaje de máquina, para este caso se entrenaron los clasificadores diseñados previamente en MATLAB, es necesario realizar el entrenamiento y comprobar el funcionamiento dado que pueden existir diferencias en la implementación de las SVM en MATLAB y Python.

Inicialmente se realizó la implementación de los clasificadores SVM para “caminar - correr” y “caídas - otros” usando los *kernel*: Lineal, RBF y polinomial. Al igual que el caso de MATLAB, los mejores resultados se obtuvieron usando el kernel RBF.

Para el entrenamiento y validación de los clasificadores, se utilizaron los mismos datos y se realiza el mismo proceso de validación implementado para el caso de MATLAB. El desempeño de ambos clasificadores puede verse en las tablas 11 y 12.

Tabla 11: Parámetros de desempeño para el clasificador SVM “caminar - correr”

Parámetro	Valor (%)
Precisión	92.41
Sensibilidad	90.8
Especificidad	92.6

Tabla 12: Parámetros de desempeño para el clasificador SVM “caídas - otros”

Parámetro	Valor (%)
Precisión	90.6
Sensibilidad	92.75
Especificidad	92.92

Inicialmente se aplica un filtro butterworth pasabajas de 0.3 Hz a la señal de presión, dado que es necesario medir los cambios de presión para calcular el diferencial y los picos o sobresaltos en la señal afectan el valor del mismo. Dicho filtrado se había planteado inicialmente con un filtro de 1 Hz, pero se observa una señal más limpia al reducir la frecuencia de corte. Además, debido a la reducción en la frecuencia de corte, el filtro toma el doble de tiempo en establecerse (10 segundos aproximadamente). Esto puede realizarse debido a que se considera que el tiempo de inicialización del algoritmo no es tan elevado como para afectar el funcionamiento del mismo. En la imagen 16 puede verse la señal de presión filtrada a las dos frecuencias de corte ya mencionadas.

EL flujo de datos del algoritmo en tiempo real no difiere del algoritmo planteado para el funcionamiento *offline* (descrito en la sección 3). Los principales cambios son los mencionados en la cantidad de datos adquiridos y el filtro aplicado a la señal de presión.

En caso de darse una eventual caída, se muestra el mensaje “POSIBLE CAÍDA”, luego de esto se entrará a un proceso de espera de aproximadamente 10 segundos antes de que el sistema sea capaz de establecer si realmente hubo una caída. por ultimo, verifica la actividad que realiza el usuario inmediatamente después.

El sistema registra una caída en dos posibles escenarios: el primero corresponde a una alerta de posible caída en la que luego del tiempo de verificación el usuario permanece acostado en el suelo. En el otro caso, el usuario cae y logra levantarse pero no se registra desplazamiento por parte del mismo, lo cual puede ser indicador de que se encuentra de pie o sentado pero no está en condiciones de caminar.

El algoritmo se implementó completamente utilizando *Python* desde una *Beaglebone Black*. Una vez se ingresa a la BBB y se ejecuta el código, se puede observar en la consola lo mostrado en la figura 17

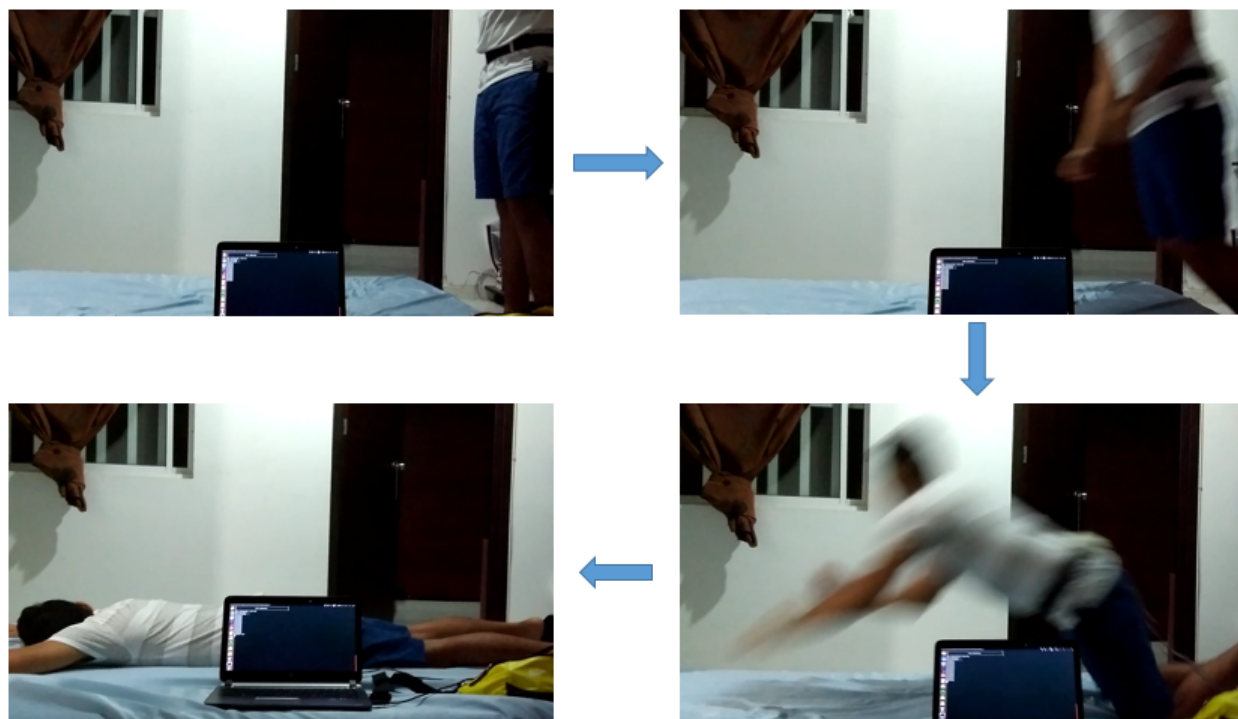
En la consola se muestra la actividad detectada por el algoritmo cada dos segundos como se había mencionado anteriormente. En caso de detectarse una posible caída el sistema lo mostrará como se ve en la imagen 17, si luego de la verificación se determina que hubo una caída se verá el mensaje “CAÍDA” y se informará la actividad que realizaba antes el usuario antes de la caída y su estado actual, de lo contrario se continuará mostrando la actividad que realiza el usuario en cada iteración.

- ❖ **Prueba 1:** el participante se mantiene de pie y quieto para luego caer sobre el colchón, debe mantenerse acostado hasta que se le solicite levantarse (figura 18). Luego se repite la misma prueba pero se le pide al participante que caiga y se levante de forma natural.
- ❖ **Prueba 2:** el participante camina hacia el colchón y luego cae sobre este, se mantiene recostado hasta que se le solicite que se levante. Posteriormente, se repite la prueba pero al final el participante debe levantarse de forma natural.
- ❖ **Prueba 3:** el participante camina hacia el colchón y luego cae sobre este, el participante debe levantarse de forma natural y caminar.

Con las pruebas 1 y 2 se pretende demostrar que el algoritmo es capaz de reconocer una caída, y enviar la alerta en situaciones en las que luego de caer el paciente se queda recostado en el suelo o puede levantarse pero se mantiene en el mismo lugar. Se consideran estos dos escenarios como los más comunes en el caso de una caída real que pueda tener efecto sobre la integridad del usuario.

La prueba 3 sirve para verificar que el algoritmo puede detectar una posible caída pero también descartar una alerta cuando no se requiere, para luego seguir la detección de actividades.

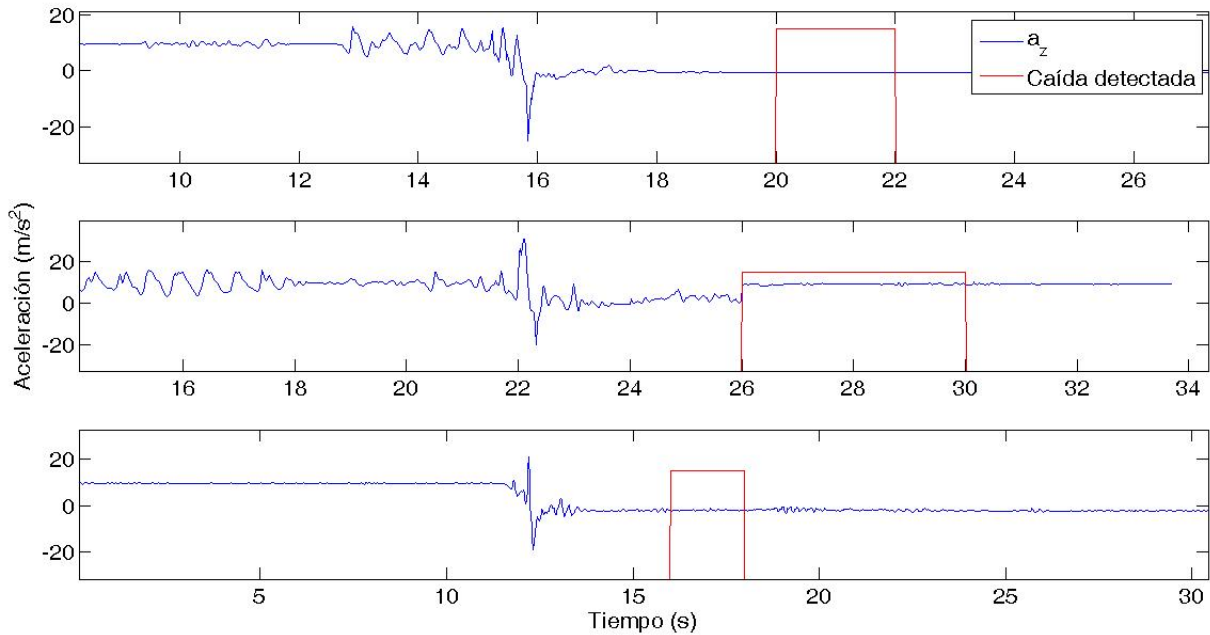
Figura 18: Ejemplo de pruebas realizadas para la validación de caídas realizada con voluntarios



En la figura 19 puede apreciarse la detección de caídas para 3 casos diferentes. En el primer caso el usuario cae luego de realizar una caminata corta, luego se mantiene acostado en el suelo hasta que se dá la alerta. En la segunda imagen, el usuario realiza una corta caminata, se detiene y luego cae. En este caso se levanta luego de la caída permaneciendo inmóvil, lo cual efectivamente es detectado como una caída. Para el ejemplo presentado en la tercera imagen, el usuario realiza una caída aislada, es decir, cae luego de estar quieto y se queda recostado luego de la caída, esto permite que se registre solamente la caída sin

que haya interferencia de otras actividades.

Figura 19: Recorrido realizado durante las pruebas de validación del algoritmo de detección de actividades



Como se puede observar, el algoritmo es capaz de identificar caídas en los casos especificados, mostrando que no se ve afectado por actividades previas y que funciona en los casos expuestos en la sección 5.3.

4.3.4. Pruebas de Validación para el Algoritmo de Detección de Actividades Para comprobar el funcionamiento del algoritmo en la detección de las actividades definidas (caminar, correr, caminar en escaleras y movimiento en ascensor) Se ejecutó la siguiente rutina:

- ❖ **Prueba1:** esta prueba se realiza en el segundo piso de Laboratorios pesados de la Universidad Industrial de Santander. Se le solicitó a los participantes caminar 15m y detenerse frente a las escaleras por 5 segundos. Luego, debe bajar las escaleras 15 escalones caminar 3 metros y bajar 15 escalones más y esperar 5 segundos. Se debe dar vuelta y subir las escaleras para luego correr los 15 m iniciales. El recorrido puede apreciarse en la imagen 20
- ❖ **Prueba2:** en esta prueba se le pide al participante que entre a un ascensor para descender y luego subir al interior del mismo. Esta prueba se realizó pocas veces dado que el movimiento de la persona no tiene mayor incidencia en la detección del movimiento del ascensor.

A partir de estas pruebas se establece el desempeño del algoritmo para clasificar las actividades realizadas por el usuario. Estas actividades se encuentran entre las más comunes en la vida cotidiana de una persona.

4.3.5. Desempeño del Algoritmo Completo en Tiempo Real Al tener el caso de un algoritmo multiclase, lo recomendado es basar el análisis de desempeño en las tasas de error y acierto en la detección. Teniendo esto en cuenta y luego de analizar los datos obtenidos de las pruebas mencionadas en la sección anterior, se calcula el error en la detección de caídas y el error en la detección de actividades.

glas que evitan falsos positivos en la clasificación. Se evidencia que con la ayuda de estas restricciones se disminuye el número de falsos positivos en la detección.

Para evaluar los clasificadores diseñados se realizó validación cruzada de 10 grupos, con lo cual se estima la precisión, sensibilidad y especificidad de ambos sistemas. Para el caso del algoritmo completo se evaluó basándose en la tasa de aciertos y errores del mismo. A partir de los resultados obtenidos es posible proceder a realizar la implementación del algoritmo en tiempo real.

La validación experimental del algoritmo en tiempo real permitió corroborar que su funcionamiento es bastante bueno dado que la tasa de error es aproximadamente del 8 %, lo que puede considerarse bajo. Con el algoritmo propuesto es posible clasificar actividades como: caminar, correr, caminata en escaleras, movimiento en ascensor, permanecer de pie y estar acostado. Además, la detección de caídas es posible con un alto grado de precisión.

En el caso de la detección de actividades, se ve que la actividad que registra el mayor error de detección es la caminata en escaleras. No obstante, se considera que esto puede ser solucionado al utilizar un barómetro de mejores especificaciones que no registre tantas fluctuaciones repentinas en la medida de presión.

Por otro lado, haber realizado la validación de la detección de caídas por separado permite asegurar la integridad de los participantes en la prueba, además de aislar el fenómeno y probarlo en los casos específicos necesarios. Con esto se consigue llegar a una estimación del comportamiento del algoritmo ante una situación real.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1. Conclusiones Generales

En este trabajo de investigación se diseñó e implementó usando *Python* un algoritmo para la detección de caídas contextualizando el evento, es decir, brindando información sobre las actividades que el usuario realiza antes y después de caer. Se propuso un algoritmo de reconocimiento de patrones basado en máquinas de soporte vectorial. Se utilizaron unidades de medida inercial para tomar señales de aceleración, velocidades angulares y presión con las cuales se realiza la detección de actividades. Una unidad *BeagleBone Black* se utilizó para la adquisición y procesamiento de los datos, esta se programó usando el lenguaje Python, el cual brinda la posibilidad de usar librerías especializadas en el procesamiento de datos y una mayor simpleza en la implementación en comparación con otros lenguajes de programación existentes. La selección de características se realizó luego de la creación de una base de datos, la cual se realizó con la participación de voluntarios en pruebas experimentales. Basándose en lo descrito en la literatura acerca de los aspectos que resaltan de cada actividad, se planteó la estructura y el flujo de datos del algoritmo, agrupando actividades y estableciendo diferentes reglas para su correcta detección. Se realizó la validación del algoritmo mediante pruebas experimentales en las que un grupo de voluntarios realizó las diferentes actividades que el algoritmo puede clasificar, incluyendo las caídas que se realizaron en un ambiente controlado, evitando así comprometer la integridad física de los participantes.

A partir de la implementación *offline* del algoritmo, fue posible establecer la estructura general del mismo. Al probar el desempeño del algoritmo usando una base de datos ajena ofreció buenos resultados, pero estos difirieron de lo observado en tiempo real, por lo que fue necesario agregar reglas para evitar la confusión entre las actividades correr y caminata en escaleras con las caídas, dado que algunos picos de aceleración asociados a estas actividades se asemejan a los que se dan cuando se registra una caída. Al realizar los cambios mencionados el desempeño en tiempo real mostró un desempeño del 100 % en la detección de caídas y del 92 % en la detección de las demás actividades. Una de las limitantes en cuanto a frecuencia de muestreo de características fue la detección del movimiento en escaleras, dado que al ser realizado usando la señal de presión con la cual se pueden determinar cambios de altura, se debe tener en cuanto que los cambios en el diferencial de presión solo pudieron ser observables a partir de un muestreo de características de al menos 100 muestras (2 segundos).

La selección de la ubicación del sensor sobre la cintura obedeció a que según la literatura y mediante experimentación, se encuentra que las caídas pueden ser mejor diferenciadas en esta posición. En trabajos previos como el presentado en [3], se muestra trabajo en la detección de algunas actividades ubicando el sensor en el pie, pero a la hora de diseñar el algoritmo completo se determinó que los resultados eran mejores ubicando el dispositivo en la cintura. Existe la posibilidad de trabajar con dos IMU, una en el tobillo y otra en la cintura. Sin embargo, una de las metas dentro de este proyecto era realizar la clasificación con el dispositivo en una sola ubicación.

El algoritmo propuesto, basado en máquinas de soporte vectorial, es capaz de clasificar las actividades “caminar”, “correr”, “caminar en escaleras”, “permanecer de pie”, “estar acostado” y por ultimo “caídas”. Estas son definidas como las clases sobre las cuales actúa el clasificador. Al realizar un proceso de validación *offline* de los clasificadores individuales y del algoritmo completo se logra sentar las bases para el desarrollo del algoritmo en tiempo real que debe ofrecer resultados similares al caso *offline*. Se decide realizar la toma de datos con ayuda de la IMU Xsens dado que los datos adquiridos con esta son de mayor calidad a los tomados con la altIMU, lo cual se refleja en la necesidad de un filtrado especial sobre la señal de presión que es tomada con el segundo sensor mencionado. Las fluctuaciones sobre la señal de presión conducen a errores en la detección de caminata en escaleras, dado que se altera el valor del diferencial.

A partir de los datos obtenidos en la toma experimental de datos, se puede decir que el muestreo a 50 Hz posibilita la detección de las diferentes actividades y el muestreo de características cada dos segundos no supone un inconveniente para la implementación en tiempo real, dado que se observa que los cambios de actividad son relativamente lentos y solo se requiere de una iteración para que el algoritmo sea capaz de captar el cambio entre actividades. Adicionalmente, el establecer reglas de verificación sobre las actividades que tienden a confundirse con caídas se evita la aparición de falsos positivos, logrando un algoritmo capaz de detectar caídas independientemente de la actividad previa que esté realizando el usuario.

La implementación de este algoritmo sobre un dispositivo que pueda ser utilizada por usuarios permanentemente, esta sujeta a la capacidad de procesamiento y el consumo de energía que conlleve su ejecución. La utilización de algoritmos de aprendizaje de máquina, ofrece una solución más robusta en comparación a la utilización de umbrales sobre la señal de aceleración. Lo anterior puede justificarse en que algunas actividades como correr pueden generar picos de aceleración similares produciendo errores en la clasificación, lo que puede ser evitado con un correcto entrenamiento de un algoritmo de aprendizaje automático.

5.2. Trabajo Futuro

El algoritmo desarrollado se podrá complementar con el diseño de un dispositivo inalámbrico sobre el cual pueda ser ejecutado. Además, puede estudiarse la implementación del algoritmo en teléfonos inteligentes, teniendo en cuenta que las especificaciones de algunos de estos dispositivos abren la puerta a ejecutar algoritmos cada vez más especializados. Cabe resaltar que sistemas operativos como *android* cuentan con librerías para la implementación de algoritmos de aprendizaje de máquina.

El algoritmo fue entrenado y validado con datos tomados de adultos jóvenes. Sería interesante poder probar su desempeño sobre adultos mayores para obtener información que se acerque más al desempeño real sobre la población objetivo.

La adquisición de un mejor barómetro podría mejorar el desempeño del algoritmo, dado que al no requerir un filtrado tan estricto, puede reducirse el tiempo de inicialización del algoritmo.

Para dar continuidad al desarrollo de algoritmos de detección de actividades podría trabajarse en sistemas para la detección de actividades más complejas, y así complementar el funcionamiento del algoritmo actual.

Bibliografía

- S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio. A smartphone-based fall detection system. *Pervasive and Mobile Computing*, 8(6):883–899, 2012.
- K. Altun, B. Barshan, and O. Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010.
- R. Alvarez, E. Pulido, and D. A. Sierra. *Climbing/Descending Stairs Detection Using Inertial Sensors and Implementing PCA and a SVM Classifier*, pages 581–584. Springer Singapore, Singapore, 2017.
- E. Andersson. *Motion Classification and Step Length Estimation for GPS/INS Pedestrian Navigation*. PhD thesis, KTH Vetenskap Och Konst, 2012.
- F. Bagalà, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PloS one*, 7(5):e37062, 2012.
- J. Bancroft, D. Garrett, and G. Lachapelle. Activity and environment classification using foot mounted navigation sensors. *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings*, (February), 2012.
- S. Beauregard and H. Haas. Pedestrian dead reckoning: A basis for personal positioning. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, pages 27–35, 2006.
- C. Bishop. *Pattern recognition and machine learning*. springer, 2006.
- P. Boissy, S. Choquette, M. Hamel, and N. Noury. User-based motion sensing and fuzzy logic for automated fall detection in older adults. *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*, 13(6):683–693, 2007.
- A. K. Bourke, P. van de Ven, M. Gamble, R. O’Connor, K. Murphy, E. Bogan, E. McQuade, P. Finucane, G. O’Laighin, and J. Nelson. Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities. *Journal of Biomechanics*, 43(15):3051–3057, 2010.
- A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.
- W. Cheng and D. Jhan. Triaxial accelerometer-based fall detection method using a self-constructing cascade-adaboost-svm classifier. *IEEE journal of biomedical and health informatics*, 17(2):411–419, 2013.
- A. I. Cuesta-vargas and J. M. Williams. The use of inertial sensors system for human motion analysis. *Physical Therapy Reviews*, 3196(May):462–473, 2016.
- C. Doukas, I. Maglogiannis, P. Tragas, D. Liapis, and G. Yovanof. Patient fall detection using support vector machines. *Artificial Intelligence and Innovations 2007: from Theory to Applications*, pages 147–156, 2007.

- H. Gjoreski, M. Lustrek, and M. Gams. Accelerometer placement for posture recognition and fall detection. In *Intelligent environments (IE), 2011 7th international conference on*, pages 47–54. IEEE, 2011.
- J. Han, H. Jeon, B. Jeon, and K. Park. Gait detection from three dimensional acceleration signals of ankles for the patients with parkinsons disease. In *Proceedings of the IEEE The International Special Topic Conference on Information Technology in Biomedicine, Ioannina, Epirus, Greece*, volume 2628, 2006.
- R. Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys and Tutorials*, 15(3):1281–1293, 2013.
- J. Hwang, J. Kang, Y. Jang, and H. Kim. Development of Novel Algorithm and Real-time Monitoring Ambulatory System Using Bluetooth Module for Fall Detection in the Elderly. *International conference of the IEEE EMBS*, 1:2204–2207, 2004.
- R. Igual, C. Medrano, and I. Plaza. Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine*, 12(1):1, 2013.
- P. Jantaraprim, P. Phukpattaranont, C. Limsakul, and B. Wongkittisuksa. Improving the accuracy of a fall detection algorithm using free fall characteristics. In *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, pages 501–504. IEEE, 2010.
- A. R. Jiménez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. *WISP 2009 - 6th IEEE International Symposium on Intelligent Signal Processing - Proceedings*, (August):37–42, 2009.
- M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & posture*, 28(2):285–291, 2008.
- M. Kangas, I. Vikman, J. Wiklander, P. Lindgren, L. Nyberg, and T. Jämsä. Sensitivity and specificity of fall detection in people aged 40 years and over. *Gait & posture*, 29(4):571–574, 2009.
- D. M. Karantonis, M. Narayanan, M. Mathie, N. H Lovell, and B. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, 10(1):156–167, 2006.
- J. Kim, H. Jang, D. Hwang, and C Park. A Step, Stride and Heading Determination for the Pedestrian Navigation System. *Journal of Global Positioning Systems*, 3(1&2):273–279, 2004.
- F. Korbinian, M. Nadales, P. Robertson, M. Angermann, and V. Nadales. Reliable real-time recognition of motion related human activities using MEMS inertial sensors. *Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 2906–2912, 2010.
- F. Korbinian, P. Robertson, and M. Angermann. Reliable Real-Time Recognition of Motion Related Human Activities Using MEMS Inertial Sensors. 2006.
- M. Kouroggi, T. Ishikawa, and T. Kurata. A method of pedestrian dead reckoning using action recognition. In *Record - IEEE PLANS, Position Location and Navigation Symposium*, number February 2016, pages 85–89, 2010.
- P. Kumar. An inertial sensing module for movement and posture monitoring for assisted living. *M. Tech. dissertation, Biomedical Engineering, Indian Institute of Technology Bombay*, 2013.
- B. Kwolek and M. Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117:489–501, 2014.

- Q. Ladetto. On foot navigation : continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. In *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation*, volume 2000, pages 1735–1740, Salt Lake City, USA, 2000.
- C. Lai, S. Chang, H. Chao, and Y. Huang. Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling. *IEEE Sensors Journal*, 11(3):763–770, 2011.
- C. Lan, Y. Hsueh, and R. Hu. Real-time fall detecting system using a tri-axial accelerometer for home care. In *Biomedical Engineering and Biotechnology (iCBEB), 2012 International Conference on*, pages 1077–1080. IEEE, 2012.
- M. Lee, C. Park, and C. Shim. A Movement-Classification Algorithm for Pedestrian using Foot-Mounted IMU. In *Proceedings of the 2012 International Technical Meeting of The Institute of Navigation*, pages 922–927, Newport Beach, USA, 2012.
- Q. Li, J. Stankovic, M. Hanson, A. Barth, J. Lach, and G. Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pages 138–143. IEEE, 2009.
- S. Lowe and G. ÓLaighin. Monitoring human health behaviour in one’s living environment: A technological review. *Medical Engineering and Physics*, 36(2):147–168, 2014.
- S. Luo and Q. Hu. A dynamic motion pattern analysis approach to fall detection. In *Biomedical Circuits and Systems, 2004 IEEE International Workshop on*, pages 1–5. IEEE, 2004.
- S. Madansingh, T. Trasher, C. Layne, and B. Lee. Smartphone based fall detection system. *International conference on Control, Automation and Systems*, (369), 2015.
- M. Mubashir, L. Shao, and L. Seed. A survey on fall detection Principles and approaches. *Neurocomputing*, 100:144–152, 2013.
- United Nations. WHO Global Report on Falls Prevention in Older Age. *Community Health*, page 53, 2007.
- N Noury, A Fleury, P Rumeau, A K Bourke, G O Laighin, V Rialle, and J E Lundy. Fall detection - principles and methods. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1663–1666, 2007.
- Y. Ohtaki, M. Susumago, A. Suzuki, K. Sagawa, R. Nagatomi, and H. Inooka. Automatic classification of ambulatory movements and evaluation of energy consumptions utilizing accelerometers and a barometer. *Microsystem Technologies*, 11(8-10):1034–1040, 2005.
- L. Ojeda and J. Borenstein. Personal dead-reckoning system for gps-denied environments. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6. IEEE, 2007.
- G. Panahandeh, N. Mohammadiha, A. Leijon, and P. Händel. Continuous hidden markov model for pedestrian activity classification and gait analysis. *IEEE Transactions on Instrumentation and Measurement*, 62(5):1073–1083, 2013.
- Y. Pao. *Statistical Pattern Recognition. Second edition*, volume 35. 1993.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

- P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti. A High Reliability Wearable Device for Elderly Fall Detection. *15(8):4544–4553*, 2015.
- P. Rashidi and A. Mihailidis. A Survey on Ambient Assisted Living Tools for Older Adults. *Biomedical and Health Informatics, IEEE Journal of*, PP(99):1, 2013.
- A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In *Proceedings - International Symposium on Wearable Computers, ISWC*, number June 2012, pages 108–109, Newcastle, UK, 2012.
- F. Riva, M. Toebes, M. Pijnappels, R. Stagni, and J. H. van Dieën. Estimating fall risk with inertial sensors using gait stability measures that do not require step detection. *Gait and Posture*, 38(2):170–174, 2013.
- A. Sabatini, G. Ligorio, A. Mannini, V. Genovese, and L. Pinna. Prior-to- and Post-Impact Fall Detection Using Inertial and Barometric Altimeter Measurements. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 4320(c):1–1, 2015.
- G. Shi, C. Chan, W. Li, K. Leung, Y. Zou, and Y. Jin. Mobile human airbag system for fall protection using mems sensors and embedded svm classifier. *IEEE Sensors Journal*, 9(5):495–503, 2009.
- M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australian conference on artificial intelligence*, volume 4304, pages 1015–1021, 2006.
- A. Sorvala, E. Alasaarela, H. Sorvoja, and R. Myllylä. A two-threshold fall detection algorithm for reducing false alarms. In *Medical Information and Communication Technology (ISMICT), 2012 6th International Symposium on*, pages 1–4. IEEE, 2012.
- Z. Sun, X. Mao, W. Tian, and X. Zhang. Activity classification and dead reckoning for pedestrian navigation with wearable sensors. *Measurement Science and Technology*, 20(1):015203, 2009.
- Y. Syed, D. Brown, D. Garrity, and A. Mackinnon. A hidden Markov model-based activity classifier for indoor tracking of first responders. *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, (January):1–6, 2015.
- M. Tolkiehn, L. Atallah, B. Lo, and G. Yang. Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 369–372. IEEE, 2011.
- W. Wan, H. Liu, L. Wang, G. Shi, and W. Li. A hybrid hmm/svm classifier for motion recognition using μ imu data. In *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pages 115–120. IEEE, 2007.
- C. Wang, C. Chiang, P. Lin, Y. Chou, I. Kuo, C. Huang, and C. Chan. Development of a fall detecting system for the elderly residents. In *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on*, pages 1359–1362. IEEE, 2008.
- A. Webb. *Statistical pattern recognition*. John Wiley & Sons, 2003.
- H. Weinberg. Using the ADXL202 in pedometer and personal navigation applications. *Analog devices AN-602 application note*, pages 1–8, 2002.
- H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek. Automatic Step Detection in the Accelerometer Signal. In *4th International Workshop on Wearable and Implantable Body Sensor Networks*, volume 13, pages 80–85, Aachen, Germany, 2007.

T. Zhang, J. Wang, P. Liu, and J. Hou. Fall detection by embedding an accelerometer in cellphone and using kfd algorithm. *International Journal of Computer Science and Network Security*, 6(10):277–284, 2006.

T. Zhang, J. Wang, L. Xu, and P. Liu. Fall detection by wearable sensor and one-class SVM algorithm. In *Lecture Notes in Control and Information Sciences*, volume 345, pages 858–863, 2006.

Referencias

- [1] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio. A smartphone-based fall detection system. *Pervasive and Mobile Computing*, 8(6):883–899, 2012.
- [2] K. Altun, B. Barshan, and O. Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010.
- [3] R. Alvarez, E. Pulido, and D. A. Sierra. *Climbing/Descending Stairs Detection Using Inertial Sensors and Implementing PCA and a SVM Classifier*, pages 581–584. Springer Singapore, Singapore, 2017.
- [4] E. Andersson. *Motion Classification and Step Length Estimation for GPS/INS Pedestrian Navigation*. PhD thesis, KTH Vetenskap Och Konst, 2012.
- [5] F. Bagalà, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PloS one*, 7(5):e37062, 2012.
- [6] J. Bancroft, D. Garrett, and G. Lachapelle. Activity and environment classification using foot mounted navigation sensors. *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings*, (February), 2012.
- [7] S. Beauregard and H. Haas. Pedestrian dead reckoning: A basis for personal positioning. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, pages 27–35, 2006.
- [8] C. Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [9] P. Boissy, S. Choquette, M. Hamel, and N. Noury. User-based motion sensing and fuzzy logic for automated fall detection in older adults. *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*, 13(6):683–693, 2007.
- [10] A. K. Bourke, P. van de Ven, M. Gamble, R. O’Connor, K. Murphy, E. Bogan, E. McQuade, P. Finucane, G. O’Laighin, and J. Nelson. Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities. *Journal of Biomechanics*, 43(15):3051–3057, 2010.
- [11] A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.
- [12] W. Cheng and D. Jhan. Triaxial accelerometer-based fall detection method using a self-constructing cascade-adaboost-svm classifier. *IEEE journal of biomedical and health informatics*, 17(2):411–419, 2013.
- [13] A. I. Cuesta-vargas and J. M. Williams. The use of inertial sensors system for human motion analysis. *Physical Therapy Reviews*, 3196(May):462–473, 2016.

- [14] C. Doukas, I. Maglogiannis, P. Tragas, D. Liapis, and G. Yovanof. Patient fall detection using support vector machines. *Artificial Intelligence and Innovations 2007: from Theory to Applications*, pages 147–156, 2007.
- [15] H. Gjoreski, M. Lustrek, and M. Gams. Accelerometer placement for posture recognition and fall detection. In *Intelligent environments (IE), 2011 7th international conference on*, pages 47–54. IEEE, 2011.
- [16] J. Han, H. Jeon, B. Jeon, and K. Park. Gait detection from three dimensional acceleration signals of ankles for the patients with parkinsons disease. In *Proceedings of the IEEE The International Special Topic Conference on Information Technology in Biomedicine, Ioannina, Epirus, Greece*, volume 2628, 2006.
- [17] R. Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys and Tutorials*, 15(3):1281–1293, 2013.
- [18] J. Hwang, J. Kang, Y. Jang, and H. Kim. Development of Novel Algorithm and Real-time Monitoring Ambulatory System Using Bluetooth Module for Fall Detection in the Elderly. *International conference of the IEEE EMBS*, 1:2204–2207, 2004.
- [19] R. Igual, C. Medrano, and I. Plaza. Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine*, 12(1):1, 2013.
- [20] P. Jantaraprim, P. Phukpattaranont, C. Limsakul, and B. Wongkittisuksa. Improving the accuracy of a fall detection algorithm using free fall characteristics. In *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, pages 501–504. IEEE, 2010.
- [21] A. R. Jiménez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. *WISP 2009 - 6th IEEE International Symposium on Intelligent Signal Processing - Proceedings*, (August):37–42, 2009.
- [22] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & posture*, 28(2):285–291, 2008.
- [23] M. Kangas, I. Vikman, J. Wiklander, P. Lindgren, L. Nyberg, and T. Jämsä. Sensitivity and specificity of fall detection in people aged 40 years and over. *Gait & posture*, 29(4):571–574, 2009.
- [24] D. M. Karantonis, M. Narayanan, M. Mathie, N. H Lovell, and B. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, 10(1):156–167, 2006.
- [25] J. Kim, H. Jang, D. Hwang, and C Park. A Step, Stride and Heading Determination for the Pedestrian Navigation System. *Journal of Global Positioning Systems*, 3(1&2):273–279, 2004.
- [26] F. Korbinian, M. Nadales, P. Robertson, M. Angermann, and V. Nadales. Reliable real-time recognition of motion related human activities using MEMS inertial sensors. *Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 2906–2912, 2010.
- [27] F. Korbinian, P. Robertson, and M. Angermann. Reliable Real-Time Recognition of Motion Related Human Activities Using MEMS Inertial Sensors. 2006.
- [28] M. Kouroggi, T. Ishikawa, and T. Kurata. A method of pedestrian dead reckoning using action recognition. In *Record - IEEE PLANS, Position Location and Navigation Symposium*, number February 2016, pages 85–89, 2010.

- [29] P. Kumar. An inertial sensing module for movement and posture monitoring for assisted living. *M. Tech. dissertation, Biomedical Engineering, Indian Institute of Technology Bombay*, 2013.
- [30] B. Kwolek and M. Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117:489–501, 2014.
- [31] Q. Ladetto. On foot navigation : continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. In *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation*, volume 2000, pages 1735–1740, Salt Lake City, USA, 2000.
- [32] C. Lai, S. Chang, H. Chao, and Y. Huang. Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling. *IEEE Sensors Journal*, 11(3):763–770, 2011.
- [33] C. Lan, Y. Hsueh, and R. Hu. Real-time fall detecting system using a tri-axial accelerometer for home care. In *Biomedical Engineering and Biotechnology (iCBEB), 2012 International Conference on*, pages 1077–1080. IEEE, 2012.
- [34] M. Lee, C. Park, and C. Shim. A Movement-Classification Algorithm for Pedestrian using Foot-Mounted IMU. In *Proceedings of the 2012 International Technical Meeting of The Institute of Navigation*, pages 922–927, Newport Beach, USA, 2012.
- [35] Q. Li, J. Stankovic, M. Hanson, A. Barth, J. Lach, and G. Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pages 138–143. IEEE, 2009.
- [36] S. Lowe and G. ÓLaighin. Monitoring human health behaviour in one’s living environment: A technological review. *Medical Engineering and Physics*, 36(2):147–168, 2014.
- [37] S. Luo and Q. Hu. A dynamic motion pattern analysis approach to fall detection. In *Biomedical Circuits and Systems, 2004 IEEE International Workshop on*, pages 1–5. IEEE, 2004.
- [38] S. Madansingh, T. Trasher, C. Layne, and B. Lee. Smartphone based fall detection system. *International conference on Control, Automation and Systems*, (369), 2015.
- [39] M. Mubashir, L. Shao, and L. Seed. A survey on fall detection Principles and approaches. *Neurocomputing*, 100:144–152, 2013.
- [40] United Nations. WHO Global Report on Falls Prevention in Older Age. *Community Health*, page 53, 2007.
- [41] N Noury, A Fleury, P Rumeau, A K Bourke, G O Laighin, V Rialle, and J E Lundy. Fall detection - principles and methods. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1663–1666, 2007.
- [42] Y. Ohtaki, M. Susumago, A. Suzuki, K. Sagawa, R. Nagatomi, and H. Inooka. Automatic classification of ambulatory movements and evaluation of energy consumptions utilizing accelerometers and a barometer. *Microsystem Technologies*, 11(8-10):1034–1040, 2005.
- [43] L. Ojeda and J. Borenstein. Personal dead-reckoning system for gps-denied environments. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6. IEEE, 2007.
- [44] G. Panahandeh, N. Mohammadiha, A. Leijon, and P. Händel. Continuous hidden markov model for pedestrian activity classification and gait analysis. *IEEE Transactions on Instrumentation and Measurement*, 62(5):1073–1083, 2013.

- [45] Y. Pao. *Statistical Pattern Recognition. Second edition*, volume 35. 1993.
- [46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [47] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti. A High Reliability Wearable Device for Elderly Fall Detection. 15(8):4544–4553, 2015.
- [48] P. Rashidi and A. Mihailidis. A Survey on Ambient Assisted Living Tools for Older Adults. *Biomedical and Health Informatics, IEEE Journal of*, PP(99):1, 2013.
- [49] A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In *Proceedings - International Symposium on Wearable Computers, ISWC*, number June 2012, pages 108–109, Newcastle, UK, 2012.
- [50] F. Riva, M. Toebe, M. Pijnappels, R. Stagni, and J. H. van Dieën. Estimating fall risk with inertial sensors using gait stability measures that do not require step detection. *Gait and Posture*, 38(2):170–174, 2013.
- [51] A. Sabatini, G. Ligorio, A. Mannini, V. Genovese, and L. Pinna. Prior-to- and Post-Impact Fall Detection Using Inertial and Barometric Altimeter Measurements. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 4320(c):1–1, 2015.
- [52] G. Shi, C. Chan, W. Li, K. Leung, Y. Zou, and Y. Jin. Mobile human airbag system for fall protection using mems sensors and embedded svm classifier. *IEEE Sensors Journal*, 9(5):495–503, 2009.
- [53] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australian conference on artificial intelligence*, volume 4304, pages 1015–1021, 2006.
- [54] A. Sorvala, E. Alasaarela, H. Sorvoja, and R. Myllylä. A two-threshold fall detection algorithm for reducing false alarms. In *Medical Information and Communication Technology (ISMICT), 2012 6th International Symposium on*, pages 1–4. IEEE, 2012.
- [55] Z. Sun, X. Mao, W. Tian, and X. Zhang. Activity classification and dead reckoning for pedestrian navigation with wearable sensors. *Measurement Science and Technology*, 20(1):015203, 2009.
- [56] Y. Syed, D. Brown, D. Garrity, and A. Mackinnon. A hidden Markov model-based activity classifier for indoor tracking of first responders. *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, (January):1–6, 2015.
- [57] M. Tolkiehn, L. Atallah, B. Lo, and G. Yang. Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 369–372. IEEE, 2011.
- [58] W. Wan, H. Liu, L. Wang, G. Shi, and W. Li. A hybrid hmm/svm classifier for motion recognition using μ imu data. In *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, pages 115–120. IEEE, 2007.
- [59] C. Wang, C. Chiang, P. Lin, Y. Chou, I. Kuo, C. Huang, and C. Chan. Development of a fall detecting system for the elderly residents. In *Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on*, pages 1359–1362. IEEE, 2008.
- [60] A. Webb. *Statistical pattern recognition*. John Wiley & Sons, 2003.

- [61] H. Weinberg. Using the ADXL202 in pedometer and personal navigation applications. *Analog devices AN-602 application note*, pages 1–8, 2002.
- [62] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek. Automatic Step Detection in the Accelerometer Signal. In *4th International Workshop on Wearable and Implantable Body Sensor Networks*, volume 13, pages 80–85, Aachen, Germany, 2007.
- [63] T. Zhang, J. Wang, P. Liu, and J. Hou. Fall detection by embedding an accelerometer in cellphone and using kfd algorithm. *International Journal of Computer Science and Network Security*, 6(10):277–284, 2006.
- [64] T. Zhang, J. Wang, L. Xu, and P. Liu. Fall detection by wearable sensor and one-class SVM algorithm. In *Lecture Notes in Control and Information Sciences*, volume 345, pages 858–863, 2006.

Apéndices

Apéndice A

Listado de Características

Para la detección de ADL, diferentes conjuntos de características han sido propuestos en la literatura. Luego de una revisión del estado del arte, se hace un análisis de las características mostradas en la tabla 14, donde a_i es la aceleración en el eje i , w_i es la velocidad angular en el eje i y m_i es el campo magnético en el eje i .

Estas características son en su mayoría medidas estadísticas, aunque también se analizan valores máximos, mínimos y diferenciales. Este listado de características se basó en el trabajo presentado por Korbinan et al. en [26]. Las Características analizadas son descritas a continuación:

❖ **Valor máximo:** Valor más alto de la señal dado en la ventana de observación.

❖ **Valor mínimo:** Valor más bajo dado en la ventana de observación.

❖ **Promedio:** Valor medio de la señal dado por la ecuación:

$$Prom = \frac{1}{n} \sum_{i=1}^n x_i \quad (A.1)$$

❖ **Varianza:** Es una medida de dispersión de los datos alrededor de la media, esta definida por la ecuación:

$$Var = \frac{1}{n} \sum_{i=1}^n (x_i - Prom)^2 \quad (A.2)$$

❖ **Desviación estándar:** Esta medida estadística es la raíz de la varianza, y se da en las unidades de la variable analizada. Está dada por la ecuación:

$$STD = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - Prom)^2} \quad (A.3)$$

❖ **Indice intercuartílico (IQR):** Medida de variabilidad definida como la diferencia entre el tercer y primer cuartil. es decir:

$$IQR = Q3 - Q1 \quad (A.4)$$

- ❖ **Valor cuadrático medio (RMS):** Esta es una medida estadística de la magnitud de una variable. Se describe como la raíz cuadrada del promedio del cuadrado de un conjunto de valores. Se calcula con la ecuación:

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \quad (A.5)$$

- ❖ **Correlación:** indica la fuerza de la relación lineal entre dos variables, puede tomar valores entre -1 y 1. Se calcula mediante la ecuación:

$$CORR = \frac{\sum_{i=1}^n (x_i - Prom_x) \cdot (y_i - Prom_y)}{STD_x \cdot STD_y} \quad (A.6)$$

- ❖ **Componente principal de frecuencia (MFC):** Es la frecuencia en la que se encuentra más energía, esta medida puede encontrarse mediante la transformada rápida de Fourier.

- ❖ **Energía baja frecuencia (EBPF):** Energía de la señal en la banda de frecuencias comprendida de 0 a 2 Hz.

- ❖ **Energía frecuencia media (ELPF):** Energía de la señal en la banda de frecuencias comprendida de 1.5 a 4 Hz.

- ❖ **Diferencial:** Es la primera diferencia de la señal, es útil para detectar cambios bruscos en los datos. Se obtiene mediante la ecuación:

$$x' = (x[n] - x[n - 1]) / \Delta T \quad (A.7)$$

- ❖ **Integral:** La integral de la señal puede brindar información sobre desplazamiento y velocidad al aplicar esta operación sobre la señal de aceleración. Se calcula mediante la ecuación:

$$INT = x_0 + \Delta T \sum_{i=1}^n x_i \quad (A.8)$$

Tabla 14: Listado de características estudiadas

Característica	Señal
Valor máximo	a_x, a_y, a_z
	w_x, w_y, w_z
	m_x, m_y, m_z
	$ a , w , z $
Valor mínimo	a_x, a_y, a_z
	w_x, w_y, w_z
	m_x, m_y, m_z
	$ a , w , z $
Valor promedio	a_x, a_y, a_z
	w_x, w_y, w_z
	m_x, m_y, m_z
	$ a , w , z $
Desviación estándar	a_x, a_y, a_z
	w_x, w_y, w_z
	m_x, m_y, m_z
	$ a , w , z $
Varianza	a_x, a_y, a_z
	w_x, w_y, w_z
	$ a , w , z $
IQR	a_x, a_y, a_z
	w_x, w_y, w_z
	$ a , w , z $
RMS	a_x, a_y, a_z
	w_x, w_y, w_z
	$ a , w , z $
MFC	a_x, a_y, a_z
	w_x, w_y, w_z
	$ a , w , z $
ELPF	a_x, a_y, a_z
	w_x, w_y, w_z
	m_x, m_y, m_z
EBPF	a_x, a_y, a_z
	w_x, w_y, w_z
	m_x, m_y, m_z
Diferencial	Presión
Integral	a_z
Segunda Integral	a_z

Apéndice B

Máquinas de Soporte Vectorial (SVM)

Para este proyecto, las máquinas de soporte vectorial son la técnica utilizada para clasificar las actividades realizadas por una persona. Las SVM plantean llevar los datos a un espacio de características de mayor dimensión, donde un plano de separación es construido [45].

SVM se plantea partiendo desde la clasificación binaria, suponiendo que se tienen datos linealmente separables (x) asignados a dos clases w_1 y w_2 , entonces la función discriminante estará definida por:

$$g(x) = W^T x + w_0 \quad (\text{B.1})$$

donde W indica la inclinación del discriminante formado.

Para considerar que un dato pertenece a una clase u otra, las SVM toman un margen de 1 para la clasificación, es decir:

$$y_i(W^T x + w_0) \geq 1 \quad (\text{B.2})$$

$$y_i(W^T x + w_0) \leq -1 \quad (\text{B.3})$$

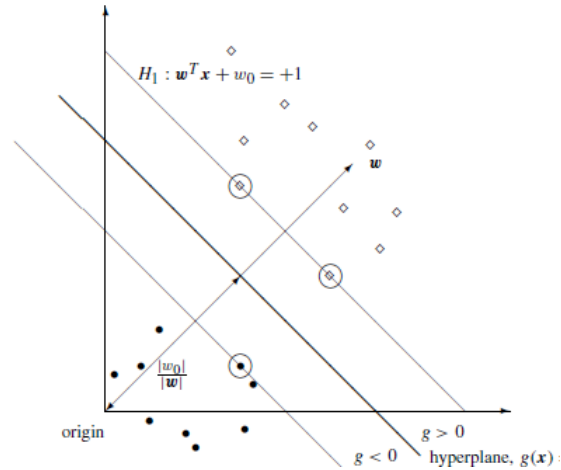
Si el discriminante es mayor que 1, entonces el dato pertenecerá a la clase w_1 . Mientras que si dicho discriminante es menor o igual a -1, el dato pertenecerá a la clase w_2 . Esto representará una separación del hiperplano de $1/w$, y será tomado como el margen de separación, mientras que los datos que se ubiquen sobre el margen serán considerados vectores de soporte, como puede observarse en el ejemplo de la figura 21.

Para extender esta idea a la clasificación de datos no lineales hay todo un planteamiento que permite llevar los datos a otro espacio de características[60] e incluso se han planteado alternativas no lineales de SVM. En el caso de que se necesite realizar clasificación entre varias clases o multiclase, pueden ser implementadas las técnicas de “uno contra el resto” o “uno contra uno”, teniendo cada una sus ventajas y desventajas dependiendo del caso [45].

En resumen, las máquinas de soporte vectorial son efectivas en espacios de características grandes, son funcionales aún en casos donde el número de características es mayor al número de muestras disponibles. Al usar solo una porción de los datos de entrenamiento como vectores de soporte es posible también afirmar que este método de clasificación es eficiente en memoria[60].

Transformación de Kernel. Las máquinas de soporte vectorial brindan la alternativa de crear planos no lineales para la separación de clases. Esto se logra representando los datos en un nuevo espacio de características y se conoce como transformación de Kernel [8]. Durante la ejecución de este proyecto

Figura 21: Clasificador basado en máquinas de soporte vectorial. Tomado de [45]



se realizan implementaciones usando MATLAB y Python, donde se definen principalmente los siguientes kernels:

- ❖ **Kernel lineal:** El kernel lineal se usa para casos en donde las clases muestran una separación lineal clara, se realiza como un producto punto. Este se considera como el caso ideal y está representado en la ecuación:

$$Linear = \langle x, x' \rangle \quad (B.4)$$

- ❖ **Kernel polinomial:** El kernel polinomial crea un plano que sigue la forma de una función polinómica del grado que se especifique, la ecuación puede verse a continuación donde la variable d representa el grado de la ecuación, mientras que r define un término independiente y γ es la pendiente del polinomio.

$$Poly = (\gamma \langle x, x' \rangle + r)^d \quad (B.5)$$

- ❖ **Kernel RBF:** Esta función kernel es de base gaussiana radial, que depende del valor del parámetro γ que debe ser mayor que 0. Se basa en la distribución de probabilidad gaussiana creada a partir de los datos de entrenamiento para cada clase.

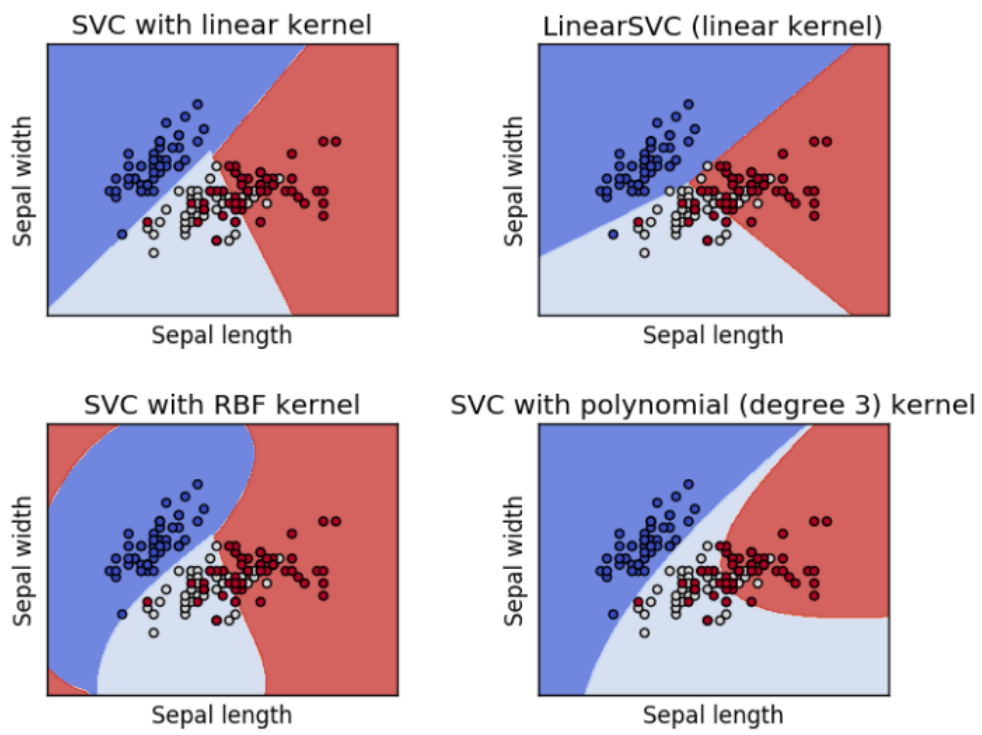
$$RBF = \exp(-\gamma \langle x, x' \rangle^2) \quad (B.6)$$

- ❖ **Kernel sigmoideo:** En este caso el kernel es la representación del kernel polinomial de grado 1 en términos de la tangente hiperbólica.

$$sigmoid = \tanh(\gamma \langle x, x' \rangle + r) \quad (B.7)$$

El kernel seleccionado dependerá de la distribución de los datos en el espacio de características, un ejemplo de la clasificación usando los diferentes kernels puede verse en la imagen 22.

Figura 22: Ejemplo de clasificación usando los diferentes kernels disponibles en Python. Tomado de [46]



Apéndice C

Algoritmo de Ajuste de Longitud del Paso

Detección del Paso Para calcular la longitud del paso mediante modelos de estimación, es necesario detectar la ocurrencia de pasos. Para este fin, se han planteado diversos algoritmos que buscan evitar falsos positivos o pasos no detectados durante una caminata. La forma más sencilla de detectar pasos se da mediante la detección de umbrales sobre la señal de aceleración pero esto puede llevar a errores en la detección dado que los picos de aceleración que se producen al caminar pueden variar para cada usuario [62].

Se han planteado diversos algoritmos para detección del paso que involucran parámetros estadísticos como la varianza y valor medio de la aceleración [21] [16] [62], estos algoritmos muestran buenos resultados *offline* pero presentan inconvenientes a la hora de implementarse en tiempo real, dado que requieren segmentos de señal largos para obtener los parámetros estadísticos.

En este trabajo, la detección del paso se realiza mediante la identificación del ciclo del paso, en el cual debe haber un pico positivo, un cruce por cero y por último un pico negativo que indica la presencia de un paso. Para la identificación de este patrón es necesario filtrar la señal a 4Hz, considerando que esto permite conservar los principales componentes de frecuencia cuando se camina o corre. Para realizar el filtrado se utiliza un filtro butterworth pasabanda de 0.05 - 4Hz (ver ecuación C.1), lo que facilita la identificación de los cruces por cero.

$$H(z) = 0,0028 \left(\frac{(1 - z^{-2})^2}{1 - 3,8410z^{-1} + 5,5382z^{-2} - 3,5532z^{-3} + 0,8560z^{-4}} \right) \quad (\text{C.1})$$

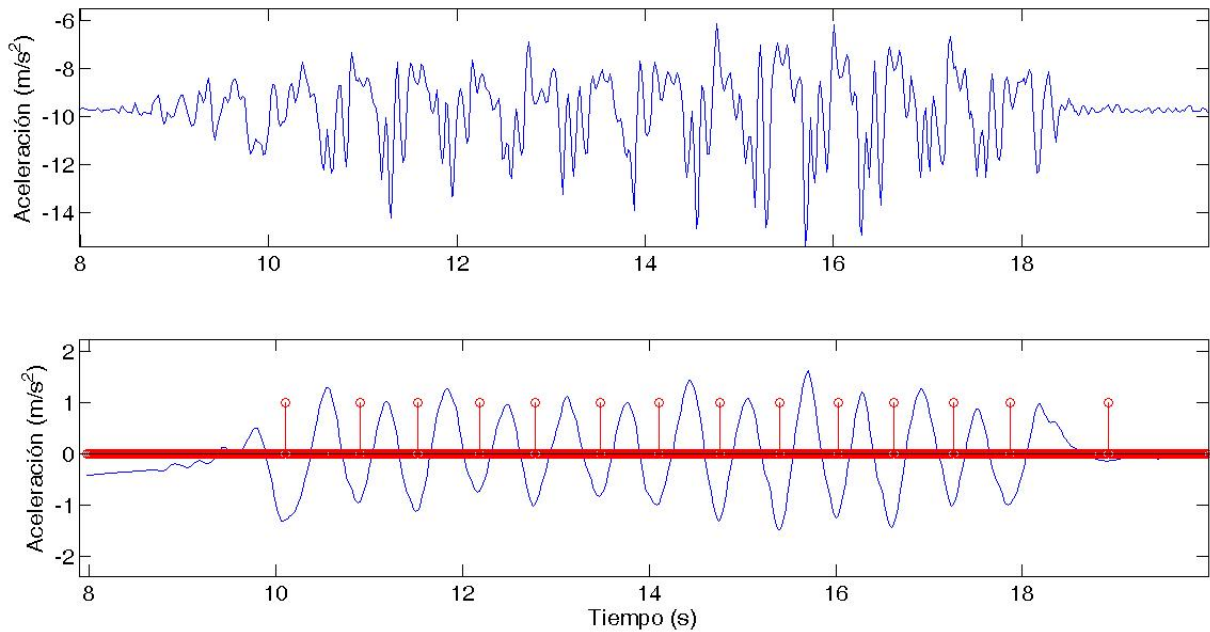
En la imagen 23 puede verse la señal de aceleración vertical antes y después de ser filtrada, así como los pasos detectados durante la caminata.

Métodos para la Estimación de Longitud del Paso El problema de la estimación de longitud del paso ha sido abordado de dos formas principalmente: Doble integración de la aceleración e implementación de modelos.

La doble integración de la aceleración puede ofrecer buenos resultados, pero presenta problemas relacionados al error que se acumula de forma cuadrática con el tiempo debido a la operación de integración [43].

Por otro lado los modelos de estimación de longitud del paso, ofrecen ventajas en cuanto a la precisión pero requieren de ajustes personalizados para cada usuario, dado que los parámetros de cada modelo pueden ser diferentes para cada persona dependiendo de variables como su estatura [31] [7].

Figura 23: Señal de aceleración vertical no filtrada (arriba) y señal filtrada con pasos detectados (abajo)



Se han presentado trabajos para estimar parámetros de regresión sobre el modelo lineal presentado en [31] y los no lineales implementados en [61] y [25] que pueden verse en la tabla 15.

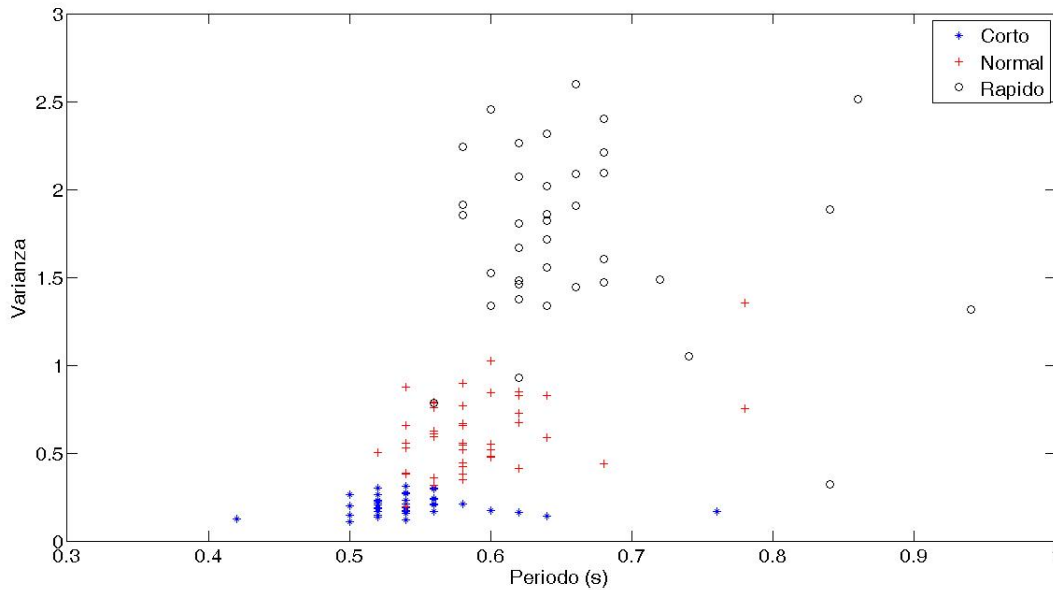
Tabla 15: Modelos de estimación de longitud del paso populares en la literatura

Modelo	Ecuación	Descripción
Lineal	$SL = A * SF + B * Var + C$	A, B y C = parámetros de regresión. SF = frecuencia del paso. Var = varianza de la aceleración.
Kim	$SL = K * \sqrt[3]{\frac{\sum_{k=1}^n a_k}{N}}$	a_k = muestras de aceleración en el intervalo un paso. N = número de muestras durante un paso. K = parámetro ajustable.
Weiberg	$SL = K * \sqrt[4]{a_{max} - a_{min}}$	a_{max} = Aceleración máxima en el intervalo de un paso. a_{min} = Aceleración mínima en el intervalo de un paso. K = parámetro ajustable.

Metodología para el Autoajuste de Longitud del Paso En este trabajo se optó por utilizar el modelo no lineal planteado por Kim [25], el cual depende de un solo parámetro como se puede observar en la tabla 15.

Para lograr que el parámetro sea auto ajustable, se hace uso del principio utilizado en el modelo lineal. En este caso se utiliza una aproximación lineal para estimar el parámetro K , en vez de calcular directamente la longitud del paso. De acuerdo a lo expuesto por Ladetto [31] para el planteamiento del modelo lineal, la longitud del paso está relacionada con la varianza y el periodo del paso, es por esto que conociendo la longitud del paso se calculan estas variables. La relación obtenida puede verse en la imagen 24.

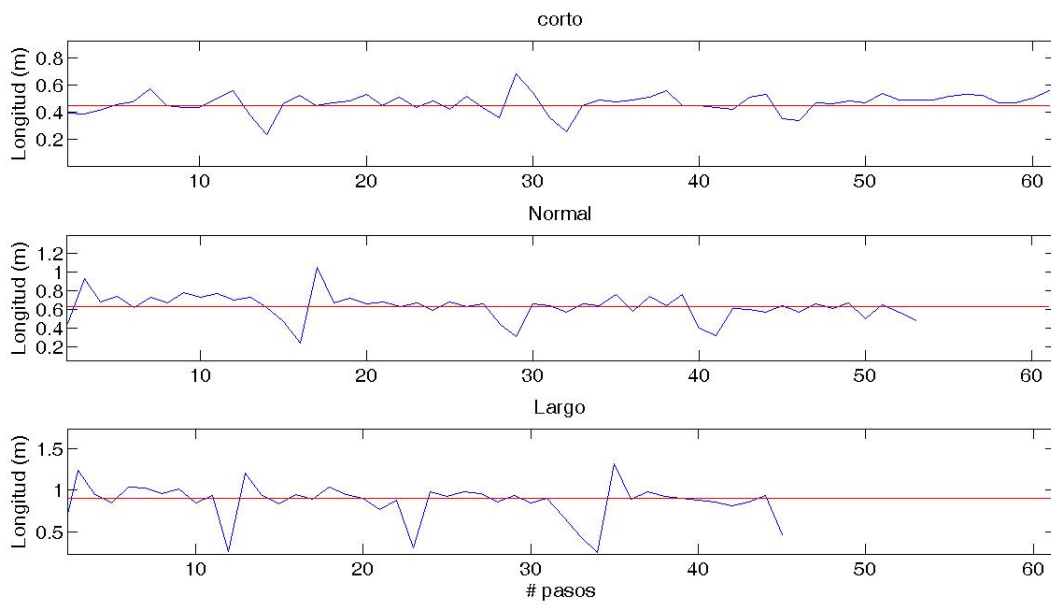
Figura 24: Relación entre varianza y periodo del paso



Una vez ajustado el valor de K para obtener las longitudes de paso deseadas es posible realizar una regresión lineal para obtener K a partir de SF y la varianza de la aceleración. A partir de este procedimiento se llega a una ecuación para estimar automáticamente K y por consiguiente la longitud del paso.

En la imagen 25 puede observarse la variación de la longitud del paso con respecto al valor real. Cabe aclarar que los picos que se presentan en algunos puntos de la señal son debido a transiciones entre quietud y movimiento; estos picos son propios del algoritmo original. Al comparar el valor promedio estimado con el valor real de la longitud del paso se encuentra que los errores de estimación son del 4.3 %, 0.7 % y 8.4 % para pasos cortos (45 cm), normales (62.5 cm) y largos (90 cm) respectivamente.

Figura 25: Valor estimado (azul) y valor real (rojo) de la longitud del paso estimado usando el algoritmo planteado



Apéndice D

Validación de Algoritmos de Reconocimiento de Patrones

Existen diferentes parámetros que se utilizan para evaluar el desempeño de un clasificador. Estos parámetros toman como criterio los aciertos y errores que se dan en la clasificación. Si se definen las clases A y B como positiva y negativa respectivamente, se manejan cuatro tipos de resultados que son [53]:

- ❖ **Verdaderos positivos (TP):** Los verdaderos positivos son aquellos datos que pertenecen a la clase A y son clasificados dentro de esta.
- ❖ **Falsos positivos (FP):** Son los datos pertenecientes a la clase B que son clasificados dentro de la clase A.
- ❖ **Verdaderos negativos (TN):** Son los datos pertenecientes a la clase B que son clasificados dentro de esta.
- ❖ **Falsos negativos (FN):** Son aquellos datos pertenecientes a la clase A que son clasificados como pertenecientes a la clase B.

A partir de las definiciones de TP, FN, FP, y TN se plantea la matriz de confusión la cual es una representación que permite visualizar el desempeño de un clasificador. Las columnas representan el número de predicciones de cada clase y las filas representan las instancias en la clase real. La matriz de confusión queda definida así:

$$\begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix}$$

A partir de la matriz de confusión es posible calcular los parámetros de desempeño más populares que son:

- ❖ **Precisión:** Esta medida indica el porcentaje de efectividad del clasificador y se describe como se muestra en la ecuación D.1.

$$Precision = \frac{(TP + TN)}{P + N} \tag{D.1}$$

- ❖ **Sensibilidad:** LA sensibilidad mide el desempeño en la clasificación de verdaderos positivos, es decir miembros de la clase considerada verdadera. Se describe en la ecuación D.2.

$$Sensibilidad = \frac{TP}{P} \tag{D.2}$$

- ❖ **Especificidad:** Esta medida también es conocida como la tasa de falsos positivos, y esta descrita por la ecuación D.3.

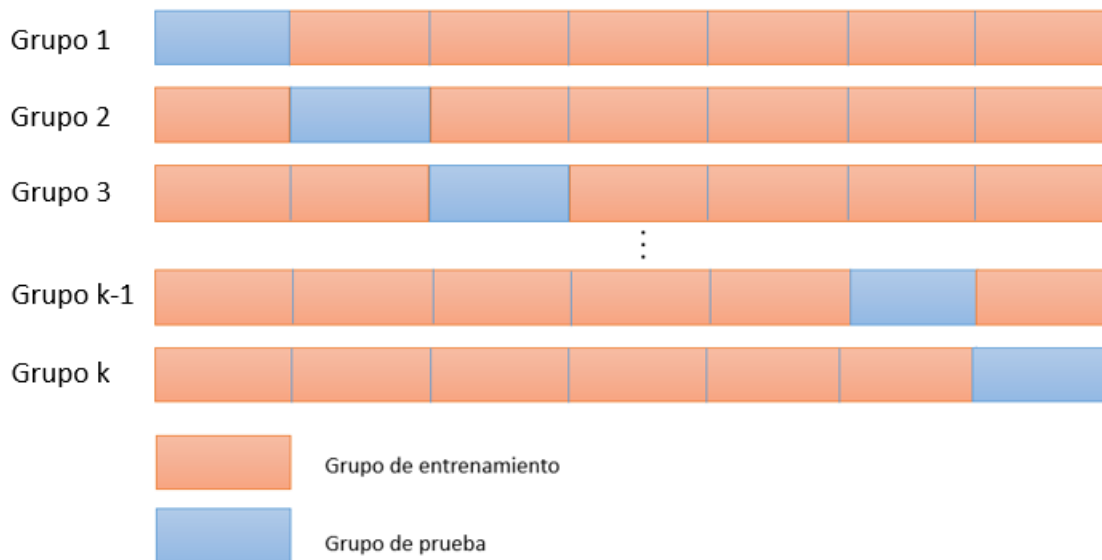
$$Precision = \frac{FP}{N} \tag{D.3}$$

De acuerdo a lo mencionado en [53] en el caso de clasificadores binarios (dos clases) es necesario usar las medidas de desempeño mencionadas, en el caso de algoritmos multiclase se suele apelar al uso de la precisión y la tasa de errores del clasificador.

D.0.1. Validación Cruzada Para la validación de un clasificador no basta con seleccionar conjuntos de datos de entrenamiento y validación únicos, considerando que los resultados de clasificación pueden variar dependiendo de los datos que se seleccionen. Es por esto que se presentan alternativas como la validación cruzada.

En la validación cruzada, los conjuntos de datos se dividen en k grupos y en cada iteración, un grupo es reservado para las pruebas mientras los demás son para entrenar el clasificador [60]. Un ejemplo de esto puede verse en la imagen 26.

Figura 26: Ejemplo de validación cruzada de k grupos. Adaptado de [53]



Existen algunas variaciones en la validación cruzada como lo son:

- ❖ **Validación estratificada:** Es útil cuando se sesga la distribución de los datos de la clase. se asegura que la distribución se respeta en los sets de entrenamiento y prueba creados en cada *fold*. Este no es el caso si se usa un proceso puramente aleatorio.

- ❖ **Leave-One-out:** En la validación cruzada cada grupo contiene m/k puntos de datos donde m es el tamaño total del set de datos. EN este caso $k = m$ y por lo tanto, cada conjunto tiene solo un punto de datos.

La técnica más conocida y mas utilizada para validación es la validación cruzada, y es menos costosa computacionalmente que realizar el método *Leave One Out*.

Los conjuntos de prueba deben ser independientes uno de otro, como se requiere por métodos de prueba estadística. Esto para asegurar que el clasificador se desempeñe bien independientemente de la distribución de los datos y evitar sobre-entrenamiento.