

**DISEÑO Y DESARROLLO DE UN PROTOTIPO PARA INFERIR UBICACIÓN  
MEDIANTE EL ALGORITMO *DEAD RECKONING* APLICANDO MÉTODOS  
DE CORRECCIÓN DE ERRORES**

Juan Camilo Bravo Pérez  
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

**DISEÑO Y DESARROLLO DE UN PROTOTIPO PARA INFERIR UBICACIÓN  
MEDIANTE EL ALGORITMO *DEAD RECKONING* APLICANDO MÉTODOS  
DE CORRECCIÓN DE ERRORES**

Juan Camilo Bravo Pérez  
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 Eléctrica

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

*En primer lugar agradezco a Dios por darme sabiduría y fuerza en los momentos que estuve a punto de desfallecer.*

*Agradezco a mis papás y mi hermano, fueron sin duda alguna el apoyo incondicional en todo momento.*

*Agradezco a mis directores de tesis: Daniel Sierra y Edith Pulido, personas profesionales que fueron mi guía en este periodo de formación profesional.*

*Finalmente agradezco a mis compañeros del proyecto ADEAM: Roberth Álvarez, Yeny Trujillo y Clara Cañon; con los que compartí momentos únicos.*

*Juan Camilo Bravo Pérez*

# ÍNDICE GENERAL

<b>INTRODUCCIÓN</b>	<b>13</b>
<b>1 MARCO TEÓRICO Y REVISIÓN DEL ESTADO DEL ARTE</b>	<b>17</b>
1.1 Introducción . . . . .	17
1.2 Sensores utilizados en localización de personas . . . . .	17
1.2.1 Sensores inerciales . . . . .	17
1.2.2 Sistema de Posicionamiento Global . . . . .	19
1.2.3 Identificación por Radiofrecuencia . . . . .	19
1.2.4 Banda Ultra Ancha . . . . .	20
1.2.5 Sensores visuales . . . . .	20
1.3 Sistemas de navegación . . . . .	20
1.3.1 Sistema de Navegación Inercial . . . . .	20
1.3.2 Navegación por Estima ( <i>Dead Reckoning</i> ) . . . . .	21
1.4 Filtro de Kalman . . . . .	22
1.4.1 Filtro Extendido de Kalman . . . . .	23
1.4.2 Proceso Gauss - Markov . . . . .	23
1.5 Errores y modelos de medida de los sensores inerciales . . . . .	24
1.5.1 Errores en los sensores inerciales . . . . .	24
1.5.2 Modelos matemáticos para modelar errores en los sensores inerciales . . . . .	25
1.6 Métodos de corrección de errores . . . . .	25
1.6.1 Corrección del error en la estimación de longitud del paso . . . . .	26
1.6.2 Corrección del error en la orientación . . . . .	27
<b>2 DISEÑO E IMPLEMENTACIÓN DEL HARDWARE</b>	<b>29</b>
2.1 Introducción . . . . .	29
2.2 Diseño del prototipo de localización . . . . .	29
2.2.1 Sensores seleccionados . . . . .	29
2.2.2 Unidad de Procesamiento . . . . .	30
2.2.3 Protocolos de comunicación . . . . .	32
2.2.4 Configuración de los sensores . . . . .	32
2.2.5 Configuración del prototipo . . . . .	33
2.2.6 Integración componentes del prototipo . . . . .	35
<b>3 IMPLEMENTACIÓN DE LOS ALGORITMOS DE LOCALIZACIÓN</b>	<b>37</b>

3.1	Introducción . . . . .	37
3.2	Estimación de posición aplicando PDR . . . . .	37
3.3	Métodos de corrección de errores para mejorar la estimación de parámetros del PDR . . . . .	38
3.3.1	Métodos de detección del paso . . . . .	39
3.3.2	Métodos de estimación de longitud del paso . . . . .	40
3.3.3	Estimación de orientación y métodos de corrección de errores . . . . .	44
3.4	Estimación de errores utilizando Filtro de Kalman . . . . .	46
3.4.1	Filtro de Kalman Extendido . . . . .	46
3.4.2	Ajuste del modelamiento de errores de la matriz de transición (modelo Gauss-Markov) . . . . .	47
3.5	Validación de los métodos de corrección de errores y estimación de parámetros del PDR . . . . .	49
3.5.1	Desempeño de la detección del paso . . . . .	50
3.5.2	Desempeño de los métodos de estimación de longitud del paso . . . . .	51
3.5.3	Desempeño de los métodos de corrección de error de la orientación . . . . .	51
3.6	Algoritmo de localización propuesto . . . . .	52
	Consideraciones del algoritmo . . . . .	53
3.7	Desempeño en interiores y exteriores . . . . .	53
3.7.1	Localización en interiores . . . . .	54
<b>4</b>	<b>VALIDACIÓN DEL PROTOTIPO</b>	<b>56</b>
4.1	Resultados en localización . . . . .	56
4.2	Desempeño en tiempo real . . . . .	58
4.2.1	Retrasos . . . . .	58
4.2.2	Consideraciones de implementación . . . . .	59
4.3	Desempeño del hardware . . . . .	59
4.3.1	Potencia consumida . . . . .	59
<b>5</b>	<b>CONCLUSIONES Y TRABAJO FUTURO</b>	<b>61</b>
5.1	Conclusiones . . . . .	61
5.2	Trabajo a futuro . . . . .	62
	<b>BIBLIOGRAFIA</b>	<b>62</b>
	<b>REFERENCIAS</b>	<b>69</b>
	<b>APÉNDICES</b>	<b>75</b>

# LISTA DE FIGURAS

Figura 1	Estimación de posición mediante PDR sin métodos de corrección de errores. [9] . . .	15
Figura 2	Unidad de Medida Inercial de 6 DOF. [18]. . . . .	18
Figura 3	Principio de funcionamiento del GPS. Adaptado de [33]. . . . .	19
Figura 4	Diagrama de bloques del funcionamiento de un Sistema de Navegación Inercial. . . . .	20
Figura 5	Sujeto con una unidad de medida inercial ubicada en la parte baja de la espalda. . . . .	21
Figura 6	Cálculo de posición aplicando DR. . . . .	21
Figura 7	Diagrama de bloques del Filtro de Kalman [6]. . . . .	22
Figura 8	Auto correlación y función de densidad espectral de un proceso Gauss-Markov [6] . . . .	24
Figura 9	Señal de aceleración horizontal filtrada con un filtro pasa bajas. . . . .	26
Figura 10	<i>Xsens MTi 1-series Development Board with MTi-3-8A7G6 module.</i> . . . .	30
Figura 11	<i>uBlox EVK-M8.</i> . . . .	30
Figura 12	Tarjeta de desarrollo <i>Beaglebone Black.</i> . . . .	31
Figura 13	Mensaje de protocolo UART. . . . .	32
Figura 14	Estructura de una trama de datos con el protocolo UBX. . . . .	32
Figura 15	Estructura de una trama de datos con el protocolo XS . . . . .	33
Figura 16	Circuito del prototipo terminado e implementado. . . . .	36
Figura 17	Patrón de las tres aceleraciones de una persona al caminar. . . . .	38
Figura 18	Gráfica de estimación de posición de personas con el algoritmo PDR. . . . .	38
Figura 19	Ciclo del paso. Adaptada de [18]. . . . .	39
Figura 20	Transformada de Fourier de la señal de aceleración vertical. . . . .	40
Figura 21	FFT para determinar la frecuencia del paso. . . . .	41
Figura 22	Diagrama de flujo de la estimación de longitud del paso basado en la frecuencia. . . . .	42
Figura 23	a) Señal de velocidad perturbada por el <i>drift</i> . b) Señal de velocidad corregida por ZUPT. .	42
Figura 24	Diagrama de flujo de la estimación de longitud del paso aplicando ZUPT. . . . .	43
Figura 25	Diagrama de flujo de la estimación de longitud del paso basado en un parámetro de pro- porcionalidad. . . . .	43
Figura 26	Diagrama de flujo del método de corrección de error de la orientación HHR. . . . .	44
Figura 27	Diagrama de flujo del método de corrección de error de la orientación ZARU. . . . .	45
Figura 28	Ruta de referencia. . . . .	47
Figura 29	Campos magnéticos para los tres escenarios planteados. . . . .	48
Figura 30	Resultados de los tres métodos de corrección de error en la orientación de la primera prueba.	52
Figura 31	Orientación de los cuatro giros con el modelo <i>yaw</i> . . . . .	53

Figura 32	Diagrama de bloques del algoritmo de localización para interiores. . . . .	54
Figura 33	Diagrama de bloques del algoritmo de localización con filtro de Kalman. . . . .	54
Figura 34	Resultados de los dos algoritmos de localización bajo la misma ruta de referencia. . . . .	55
Figura 35	Comparación de resultados del algoritmo en interiores y exteriores. . . . .	55
Figura 36	Diagrama de bloques del algoritmo con los respectivos retardos. . . . .	58

# LISTA DE TABLAS

Tabla 1	Especificaciones técnicas de la <i>Xsens MTi 1-series Development Board with MTi-3-8A7G6 module</i> . . . . .	29
Tabla 2	Especificaciones técnicas del receptor <i>uBlox EVK-M8</i> . . . . .	31
Tabla 3	Especificaciones técnicas del receptor <i>uBlox EVK-M8</i> . . . . .	31
Tabla 4	Pines utilizados de la tarjeta de desarrollo. . . . .	34
Tabla 5	Medidas tomadas de la IMU Xsens. . . . .	34
Tabla 6	Medidas tomadas del receptor GPS uBlox. . . . .	34
Tabla 7	Especificaciones técnicas del regulador 7805. . . . .	35
Tabla 8	Estimación de longitud del paso y frecuencia del paso. . . . .	41
Tabla 9	Varianzas en las señales de campos magnéticos del tercer tramo para cada escenario. . . . .	48
Tabla 10	Banco de constantes de tiempo para el modelo Gauss-Markov del error en el bias de la orientación. . . . .	48
Tabla 11	Resumen de longitudes del paso a una frecuencia normal de caminata para usuarios de diferente estatura. . . . .	49
Tabla 12	Banco de constantes de tiempo para el modelo Gauss-Markov del error en la longitud del paso. . . . .	49
Tabla 13	Resultados de detección del paso a diferentes frecuencias de caminata detectando la fase de apoyo. . . . .	50
Tabla 14	Resultados de detección del paso a diferentes frecuencias con la señal de aceleración horizontal. . . . .	50
Tabla 15	Resultados de la estimación de longitud del paso para cada método implementado. . . . .	51
Tabla 16	Resultados de los métodos de corrección de error en la orientación de la primera prueba. . . . .	52
Tabla 17	Resultados de la prueba de cuatro giros. . . . .	53
Tabla 18	Resultados obtenidos en localización en tiempo real con constantes de tiempo dinámicas. . . . .	56
Tabla 19	Resultados obtenidos en localización en tiempo real con constantes de tiempo fijas. . . . .	57
Tabla 20	Resultados obtenidos en localización en tiempo real de la orientación por tramos con constantes de tiempo dinámicas. . . . .	57
Tabla 21	Resultados obtenidos en localización en tiempo real de la orientación por tramos con constantes de tiempo fijas. . . . .	57
Tabla 22	Tabla con el consumo de potencia de cada componente del prototipo. . . . .	60

# LISTA DE APÉNDICES

<b>Appendices</b>	<b>75</b>
<b>Apéndice A Protocolos y direcciones</b>	<b>76</b>
A.1 Protocolo XS . . . . .	76
A.2 Protocolo UBX . . . . .	77
<b>Apéndice B Esquemático del circuito impreso</b>	<b>78</b>
B.1 Esquemático . . . . .	78
B.2 Circuito impreso . . . . .	79
<b>Apéndice C Resultados, productos e impactos</b>	<b>80</b>
C.1 Resultados . . . . .	80
C.2 Productos . . . . .	80
C.3 Impactos . . . . .	80

# RESUMEN

**TÍTULO:** DISEÑO Y DESARROLLO DE UN PROTOTIPO PARA INFERIR UBICACIÓN MEDIANTE EL ALGORITMO *DEAD RECKONING* APLICANDO MÉTODOS DE CORRECCIÓN DE ERRORES <sup>1</sup>

**AUTOR:** Juan Camilo Bravo Pérez<sup>2</sup>

**PALABRAS CLAVES:** Navegación por estima, Sistema de Posicionamiento Global, Unidad de Medida Inerciales, sensores inerciales.

## DESCRIPCIÓN:

Este trabajo de investigación presenta el diseño y la implementación de un prototipo capaz de inferir ubicación, basándose en medidas de sensores inerciales y un receptor GPS, aplicando el algoritmo *Pedestrian Dead Reckoning* (PDR). Además presenta una comparación de métodos de corrección de errores de la longitud del paso y de la orientación propuestos por algunos autores en la literatura. Finalmente se hace la implementación de un algoritmo final con base en los resultados obtenidos.

Se presenta el proceso de diseño del *hardware*, los sensores se comunicaron con el protocolo de comunicación UART, y se utilizó una Unidad de Medida Inercial (IMU), un receptor GPS y una unidad de procesamiento. La implementación en tiempo real se realizó a través de *threads* para paralelizar la lectura de los sensores de manera paralela. Los resultados obtenidos en la estimación de longitud del paso fueron menores al 1% y en la orientación menores a 3 grados en giros de 90.

Los errores obtenidos son menores al 5% en la ubicación final. Los errores en interiores están directamente relacionados con la duración que el prototipo no pueda obtener información del GPS. El prototipo puede llegar a estimar una posición con errores menores al 5% hasta inclusive por dos minutos desde que la persona no realice giros mayores a 20 grados, de lo contrario el bias aumenta e inclusive puede llegar a divergir. Adicionalmente los errores tanto de orientación como en la estimación de longitud del paso se lograron reducir hasta 0.5% utilizando un banco de constantes de tiempo para el modelo Gauss-Markov de primer orden con el que se modelaron estos dos errores.

---

<sup>1</sup>Trabajo de grado de maestría

<sup>2</sup>Facultad de Ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Directora: Edith Pulido Herrera. Ph.D., Codirector: Daniel Alfonso Sierra Bueno. Ph.D.

# ABSTRACT

**TITLE:** DESIGN AND IMPLEMENTATION OF A PROTOTYPE TO INFER LOCATION USING THE DEAD RECKONING ALGORITHM APPLYING ERROR CORRECTION METHODS<sup>3</sup>

**AUTHOR:** Juan Camilo Bravo Pérez<sup>4</sup>

**KEYWORDS:** Pedestrian Dead Reckoning, Global Position System, Inertial Measurement Unit, inertial sensors.

## **DESCRIPTION:**

This work presents the design and implementation of a prototype for inferring location, based on inertial sensors measurements and a GPS receiver, applying the Pedestrian Dead Reckoning algorithm (PDR). In addition, a comparison of step length and heading correction methods proposed by some authors in the literature is presented. Finally, the implementation of a final algorithm is done based on the results obtained.

The work also presents the hardware design process, the communication protocol used was UART to communicate the sensors and the main unit. There was used a Inertial Measurement Unit (IMU), a GPS receptor and a main unit. The real time implementation was made using threads. The obtained results in the step length estimation have errors under 1 % and of the heading are less than 3 degrees in turns of 90.

Errors obtained were less than 5 % in final position. The errors in indoors are directly related to the duration that the prototype can not obtain information from the GPS. The prototype could estimate position with errors below 5 % up to two minutes only if the person don't make turns over 20 degrees, otherwise the bias is high and even could make the system diverge. Additionally, the errors of step length and heading estimation methods were reduced up to 0.5 % using a set of time constants for the first order Gauss-Markov model with which the errors were modeled.

---

<sup>3</sup>Master thesis: Master on electronic engineering.

<sup>4</sup>Facultad de Ingenierías Físico-mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Directora: Edith Pulido Herera. Ph.D. Codirector: Daniel Alfonso Sierra Bueno. Ph.D.

# INTRODUCCIÓN

## Motivación

El uso de tecnologías tradicionales para la localización de personas como el Sistema de Posicionamiento Global (GPS) es altamente usado en la actualidad. Sin embargo, esta tecnología es funcional en ambientes externos y bajo ciertos criterios, condiciones que no son aceptables para muchas aplicaciones de servicios basados en localización. En el caso particular de localización de personas, el ambiente cotidiano de estas puede variar entre exteriores e interiores en varias ocasiones, por lo que sistemas adicionales deben ser implementados. Este trabajo tiene como fin poder integrar medidas de sensores inerciales y un GPS para permitir no solo la localización en interiores, sino además mejorar la exactitud de localización en ambientes externos en comparación con sistemas como el GPS.

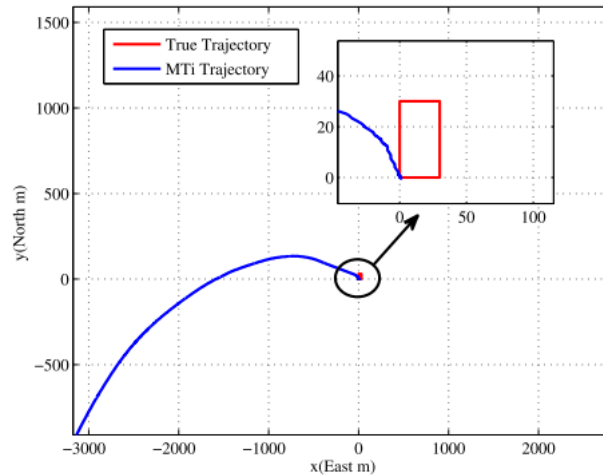
## Planteamiento del problema

En la actualidad la localización de personas en aplicaciones comerciales se realiza a través del Sistema de Posicionamiento Global (GPS). Sin embargo, este sistema es funcional bajo ciertas condiciones como cielo abierto, no interferencia por estructuras, entre otros. Por esta razón, algoritmos como *Pedestrian Dead Reckoning* (PDR) y Sistemas de Navegación Inercial (INS por sus siglas en inglés) han sido propuestos e implementados por varios autores. En el caso específico del PDR, se deben conocer tres aspectos fundamentales: la longitud del paso del usuario, la orientación y la posición inicial. Comúnmente, se utilizan medidas de sensores inerciales para estimar estos parámetros, sin embargo cualquier sensor que brinde información para estimarlos puede ser usado. La principal ventaja de este algoritmo es que permite estimar la ubicación de las personas en entornos donde el GPS no puede.

Sin embargo, los sensores inerciales que suelen utilizarse son de tecnología micro electromecánica (MEMS por sus siglas en inglés), y estos al ser tecnología de bajo costo trae consigo errores en la medición. Procesos de calibración deben ser implementados para mejorar la medición de estos. Por esta razón, métodos de corrección de errores deben ser aplicados con el fin de evitar que la estimación de posición diverja, como se observa en la figura 1 [9]. En este caso se evidencia que la estimación diverge de manera casi que instantánea, lo que concluye en la necesidad de aplicar métodos de corrección de errores.

Este proyecto pretender diseñar e implementar un prototipo para localización de personas, haciendo previamente una comparación de métodos de corrección de errores para seleccionar los que mejor desempeño tengan. Finalmente se muestran los resultados obtenidos con un análisis de los errores y del desempeño. Algunos productos de este trabajo fueron realizados bajo el proyecto: “Análisis de desorientación espacial

Figura 1: Estimación de posición mediante PDR sin métodos de corrección de errores. [9]



a través de los individuos. Caso de estudio: Adulto mayor”, cuya investigadora principal es la Dra. Edith Pulido Herrera en colaboración con el profesor Daniel Sierra.

## Objetivo general

Diseñar e implementar un prototipo que permita estimar la posición de una persona en interiores y exteriores, utilizando métodos de corrección de error de bajo costo computacional que permita la estimación en tiempo real.

## Conclusiones

Para el algoritmo de localización, los métodos de detección del paso probados en este trabajo evidenciaron un buen comportamiento a frecuencias normales y altas, a frecuencias bajas su error es mayor de hasta 3%. Además el prototipo puede estimar la posición en ambientes internos hasta por dos minutos siempre y cuando el usuario no realice giros mayores a 20 grados, de lo contrario el bias se incrementa y puede llegar hacer que el sistema diverja. También hay que mencionar que la implementación se realizó en tiempo real, por lo que un análisis de tiempos llegó a concluir que se presenta un retraso de hasta 50ms, tiempo que es bajo para este tipo de aplicaciones.

Con relación al *hardware*, teniendo en cuenta el consumo de potencia de la IMU, el receptor GPS y la unidad de procesamiento, además de la capacidad de la batería con la que se está energizando el circuito, se concluyó que el prototipo tiene una duración útil de no más de 45 minutos, tiempo que para este tipo de aplicaciones es muy baja por lo que para trabajos futuros se plantea un mejor diseño del sistema de polarización.

Finalmente, los errores se lograron reducir hasta un 0.5% en la longitud del paso y orientación utilizando un banco de constantes de tiempo para el modelo Gauss-Markov de primer orden utilizado para modelar el error en la longitud del paso y el bias de la orientación. Este modelo se utiliza en la matriz de transición de un filtro de Kalman Extendido (EKF).

## Organización del documento

Este documento está dividido por capítulos como se describe a continuación:

- ❖ **Capítulo 2:** Descripción del marco teórico y conceptos teóricos fundamentales que se deben entender. Además se presenta un estado del arte de trabajos que han realizado otros autores respecto a localización de personas utilizando otros sensores.
- ❖ **Capítulo 3:** Se hace una descripción del *hardware* y su implementación. Se mencionan los sensores utilizados, los protocolos de comunicación implementados y un análisis de su configuración. Finalmente se menciona el diseño del sistema de polarización.
- ❖ **Capítulo 4:** En este capítulo se describen los algoritmos implementados para la estimación de longitud del paso y la corrección en la orientación. Además se hacen pruebas preliminares en interiores para determinar su desempeño. También se presenta el algoritmo de localización final con su implementación del Filtro de Kalman Extendido. Finalmente se menciona el proceso de como se implementó el banco de constantes de tiempo para el modelo Gauss-Markov de primer orden con el cual fueron modelados los errores en la longitud del paso y el bias en la orientación.
- ❖ **Capítulo 5:** Una validación del prototipo se lleva a cabo. Primero se hace una validación del *hardware*, desde el consumo de potencia hasta la duración del prototipo. Finalmente se realiza una validación de los algoritmos, presentando un análisis del desempeño en tiempo real y de los errores obtenidos.
- ❖ **Capítulo 6:** Por último se presentan las conclusiones y el trabajo a futuro.

# Capítulo 1

## MARCO TEÓRICO Y REVISIÓN DEL ESTADO DEL ARTE

### Introducción

En el campo de localización de personas, el Sistema de Posicionamiento Global (GPS, por sus siglas en inglés) es comúnmente usado para estimación de posición. Sin embargo, esta tecnología es funcional únicamente en ambientes externos, y su exactitud depende de ciertas condiciones como cielo abierto, zonas despejadas, entre otras [33]. Alternativas como Sistemas de Navegación Inercial (INS por sus siglas en inglés) o Navegación por Estima (DR por sus siglas en inglés) han sido probados como complemento al GPS. La ventaja de estos métodos es que pueden ser utilizados en exteriores e interiores, incorporando sensores inerciales como acelerómetros y giroscopios. Por su forma de operar, los sensores inerciales presentan errores que terminan en una divergencia en la estimación de posición, por lo que métodos de corrección de errores deben ser implementados.

Como alternativa al problema identificado, se han planteado diferentes métodos de corrección de errores que permitan una estimación de posición más exacta. En este capítulo se mencionan los sensores comúnmente utilizados en la localización de personas, su funcionamiento y los algoritmos matemáticos utilizados para estimar posición. Se describen los errores y su modelado, al igual que algunos métodos de corrección de errores encontrados en la literatura.

### Sensores utilizados en localización de personas

Los sensores más utilizados para localización de personas son receptores GPS y sensores inerciales. Sin embargo algunos autores han utilizado otros como tecnología de radiofrecuencia (RFID por sus siglas en inglés), WiFi, *Ultra Wide Band* (UWB), sensores ópticos, entre otros [54]. Una descripción de estos sensores se presenta a continuación:

#### Sensores inerciales

Los sensores inerciales han sido ampliamente usados en sistemas de navegación inercial. Los acelerómetros y giroscopios son sensores inerciales que suelen complementarse entre si para obtener posición y orientación de una persona con un procesamiento matemático previo. A esta combinación de acelerómetros

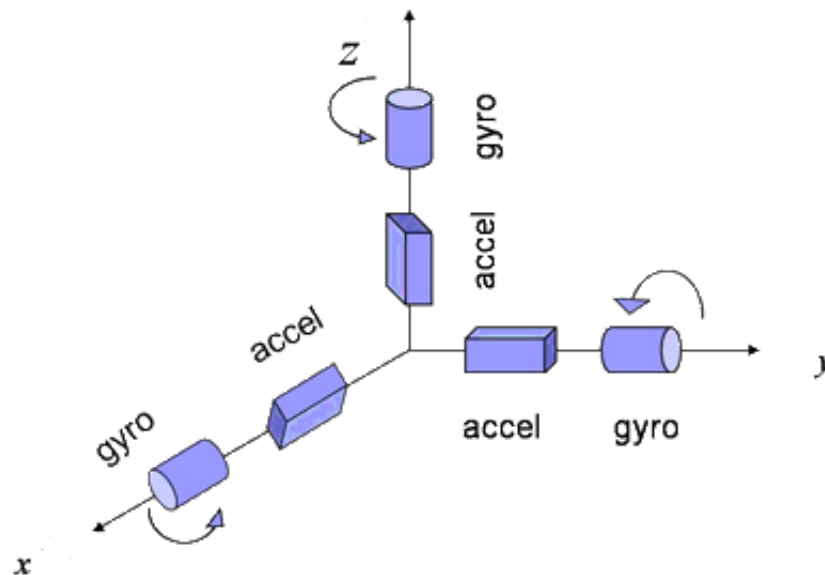
y giroscopios se le conoce como Unidad de Medida Inercial (IMU por sus siglas en inglés). A continuación se explica el funcionamiento de estos sensores.

**Acelerómetros.** Un acelerómetro es un sensor que se basa en la Segunda Ley de Newton ( $F = ma$ ) para medir la aceleración lineal. Por lo general, una Unidad de Medida Inercial contiene tres acelerómetros ortogonales para medir las aceleraciones lineales en tres dimensiones. Con estas medidas se pueden estimar algunas características del movimiento de la persona como detectar pasos, estimar la longitud del paso, calcular su velocidad, entre otros.

**Giroscopios.** Los giroscopios son sensores que miden velocidad angular basados en el efecto *coriolis*. Una IMU suele tener tres giroscopios de forma ortogonal para medir velocidad angular en tres ejes. Integrando estas mediciones se puede obtener la orientación de la persona (*Pitch, Roll, Yaw*). También se puede utilizar para estimar características del movimiento de la persona como detección del paso [9].

**Unidad de medida inercial.** Una Unidad de Medida Inercial, en su forma más básica, está compuesta por tres acelerómetros y tres giroscopios ubicados de forma ortogonal, (Figura 2). Esta sería una IMU de 6 grados de libertad (6DOF), ya que mide aceleración lineal en tres ejes ortogonales al igual que la velocidad angular [46].

Figura 2: Unidad de Medida Inercial de 6 DOF. [18].



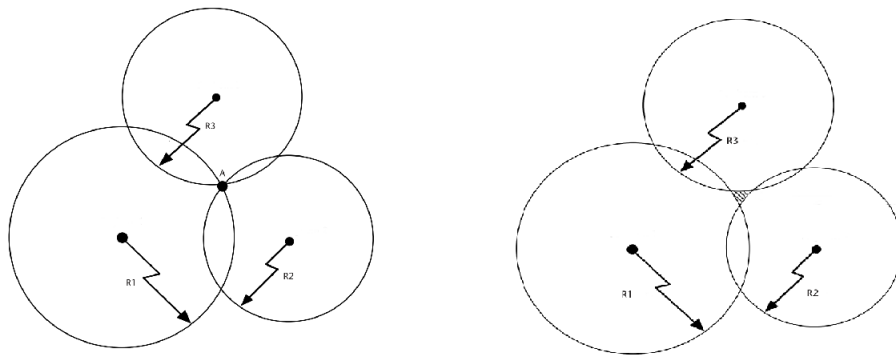
El uso de IMUs ha sido muy común en aplicaciones como robótica o navegación de vehículos no tripulados [66], ya que permiten calcular orientación y velocidad. Además, algunos incorporan magnetómetros que dan información acerca del norte magnético. Esto es utilizado como complemento para estimar la orientación ya que son lo suficientemente sensibles para medir los campos magnéticos de la tierra. Algunos autores utilizan los magnetómetros como única fuente de información para determinar la orientación [30] [37] [38], utilizando métodos matemáticos como filtro de Kalman como método de estimación de errores. Otros trabajos presentan una combinación de giroscopios y magnetómetros para determinar las perturbaciones magnéticas de las medidas de campo magnético [57] [29]. Algunas IMUs también traen consigo un

barómetro para medir la presión, y así poder determinar la altitud del usuario y lograr una estimación de posición en tres dimensiones [40] [39].

### Sistema de Posicionamiento Global

El GPS es un sistema que permite obtener en tiempo real la velocidad y la posición en tres dimensiones de un usuario que porte un receptor. Su funcionamiento se basa en triangulación de señales provenientes de 24 satélites que lo conforman como se observa en la figura 3. Uno de los errores se presenta en la sincronización de los satélites. El tiempo de desfase en el envío de las señales produce un error en la estimación de la posición[4]. Otros errores se presentan debido al desfase de los relojes del satélite y el receptor [33].

Figura 3: Principio de funcionamiento del GPS. Adaptado de [33].



Al ser un sistema para estimar la posición, este debe estar referenciado a un sistema de coordenadas. Algunos de los que se han usado son ECI (*Earth-Centered Inertial System*) y J2000. Actualmente, suele usarse el sistema ECEF (*Earth - Centered Earth-Fixed*) y coordenadas geodésicas (latitud, longitud y altitud).

Para interiores se han extendido dos sistemas basados en GPS: HS-GPS y A-GPS[54]. El primero es un sistema lo suficientemente sensible para detectar señales satelitales débiles y con esto estimar una posición [14]. El segundo comúnmente está conformado por un receptor tradicional GPS que es capaz de recibir datos de una red de telecomunicaciones. Esta red puede tener acceso a información de los satélites desde su red de referencia y compartirla con el receptor GPS [47].

### Identificación por Radiofrecuencia

La tecnología de RFID es un sistema utilizado para localizar objetos usando señales de radio [58]. Por esta razón ha sido ampliamente utilizada por diferentes autores como complementos a los sensores inerciales para la localización de personas especialmente en interiores [31]. Sin embargo, en [41] es utilizado en combinación con un GPS para mitigar los errores en la transición de interiores a exteriores.

## Banda Ultra Ancha

La localización basada en UWB se basa en transmitir una serie de pulsos de muy corta duración para asegurar una estimación de posición muy exacta [54]. En [34] se presenta un método de localización para interiores utilizando únicamente tres receptores y conociendo algunas características del entorno como techos y paredes. Otros autores han optado por hacer fusión de datos con medidas de otros sensores. En [52] utilizan sensores inerciales de tecnología Micro electromecánica (MEMS por sus siglas en inglés) y un filtro de Kalman extendido como algoritmo de fusión de datos.

## Sensores visuales

La localización de personas usando sensores visuales ha sido ampliamente usado en especial en ambientes internos. Su funcionamiento se basa en capturar imágenes de donde se encuentra el sujeto y compararlas con una base de datos. Esta base de datos debe ser construida previamente tomando imágenes del entorno de prueba [64]. Este método es conocido como "*fingerprinting*".

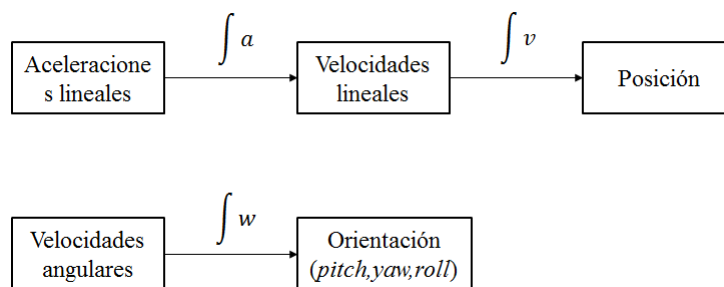
## Sistemas de navegación

El GPS es un sistema de localización ampliamente utilizado para localización, pero debido a sus limitaciones de ser funcional bajo ciertas condiciones ya mencionadas, es de interés presentar algunas alternativas. A continuación se menciona algunos sistemas de localización que han sido implementados por varios autores [3] [61] [70].

### Sistema de Navegación Inercial

El Sistema de Navegación Inercial se basa en las medidas de aceleración y velocidad angular de una IMU ubicada en el sujeto para determinar la posición y la orientación de una personas (Figura 4).

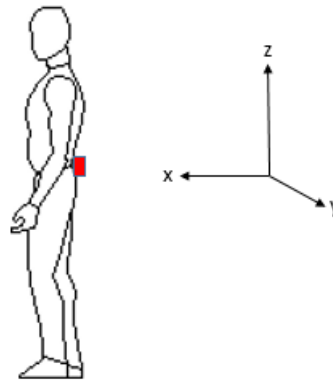
Figura 4: Diagrama de bloques del funcionamiento de un Sistema de Navegación Inercial.



La IMU es ubicada usualmente en la parte baja de la espalda del sujeto como se muestra en la figura 5. En este ejemplo haciendo una doble integración de la aceleración en los ejes  $x$  y  $y$  se obtiene la posición. De igual forma integrando la velocidad angular sobre el eje  $x$  se obtiene la orientación.

Otros autores ubican el sensor en otras partes del cuerpo como pie [50], pecho [17], entre otros. Al estar la IMU ubicada en un sistema de referencia diferente al sistema de referencia del cuerpo, un cambio de referencia debe hacerse. Algunos métodos comunes es el uso de ángulos de Euler [36], cuaterniones [56] y matrices de cosenos directores [49].

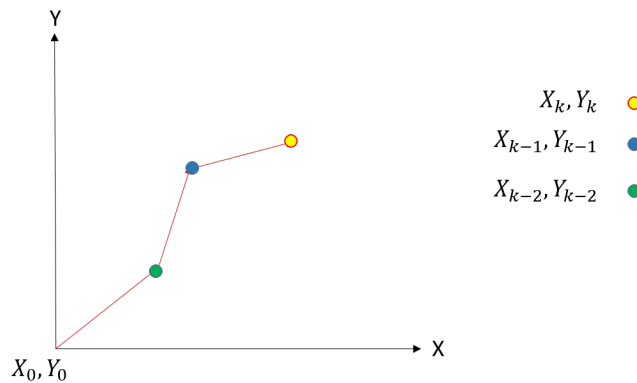
Figura 5: Sujeto con una unidad de medida inercial ubicada en la parte baja de la espalda.



### Navegación por Estima (*Dead Reckoning*)

Navegación por estima (DR), es un algoritmo que ha sido ampliamente utilizado en el campo de la robótica, pero ha sido extendido a la localización de personas por varios autores [15] [24] [23]. Este algoritmo se basa en tres aspectos fundamentales: La distancia recorrida entre dos periodos de tiempo consecutivos, la orientación y la posición del instante de tiempo anterior. En cada iteración se tendrá una nueva posición (Figura 6).

Figura 6: Cálculo de posición aplicando DR.



La estimación de posición se hace en un eje coordenado cartesiano de dos dimensiones aplicando las siguientes ecuaciones:

$$X_k = X_{k-1} + d_k * \cos(\Psi_k) \quad (1.1)$$

$$Y_k = Y_{k-1} + d_k * \sen(\Psi_k) \quad (1.2)$$

Donde  $X_k$  y  $Y_k$  es la posición X y Y en la iteración  $k$ ,  $X_{k-1}$  y  $Y_{k-1}$  es la posición del instante anterior,  $d_k$  es el desplazamiento entre los dos instantes de tiempo y  $\Psi_k$  es la dirección del desplazamiento. De las ecuaciones 1.1 y 1.2 se puede inferir la posición final [22]:

$$X = X_0 + \sum_{k=0}^N d_k * \cos(\Psi_k) \quad (1.3)$$

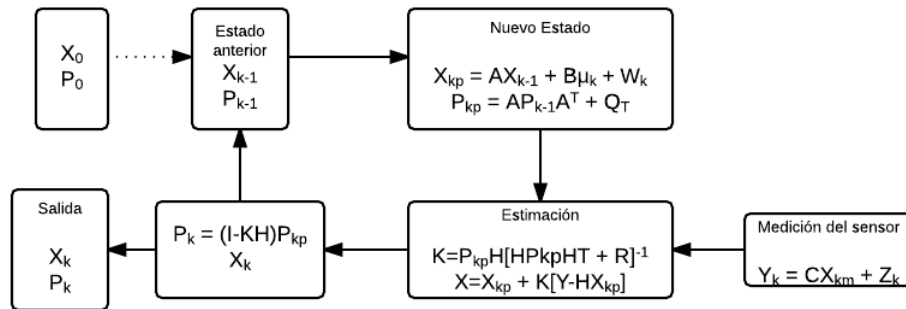
$$Y = Y_0 + \sum_{k=0}^N d_k * \sen(\Psi_k) \quad (1.4)$$

Donde  $X$  y  $Y$  es la posición final,  $X_0$  y  $Y_0$  es la posición inicial,  $d_k$  y  $\Psi_k$  es el desplazamiento y dirección del instante de tiempo  $k$ .

## Filtro de Kalman

El filtro de Kalman es un algoritmo recursivo que hace uso de la dinámica del sistema y de las propiedades estadísticas de los errores en las medidas para estimar el error de un número de estados que son linealmente relacionados con las mediciones [24]. Un diagrama de bloques del Filtro de Kalman se puede observar en la Figura 7.

Figura 7: Diagrama de bloques del Filtro de Kalman [6].



Para definir un filtro de Kalman, debe definirse su modelo de observación y su modelo de estado. El modelo de estado se puede modelar asumiendo un proceso aleatorio [6]:

$$X_{k-1} = \Phi_k X_k + w_k \quad (1.5)$$

Y el modelo de observación que se asume ocurre en instantes de tiempo discreto:

$$z_k = H_k X_k + v_k \quad (1.6)$$

Asumiendo que se conoce  $\Phi_k$ ,  $H_k$  y las covarianzas  $w_k$  y  $v_k$ . Algunas de las variables utilizadas en el filtro de Kalman se mencionan a continuación:

$x_k$  = Vector de estados del proceso en el instante  $k$

$\Phi_k$  = Matriz de transición

$w_k$  = Vector de ruido blanco con covarianzas conocidas

$z_k$  = Vector de medida en el instante de tiempo  $k$

$H_k$  = Matriz que relaciona el vector de medida con el vector de estado

$v_k$  = Vector de error de medida con covarianza conocida y correlación cruzada de cero con el vector  $w_k$

Otro elemento de importancia es la ganancia del filtro. Este parámetro es el que decide que tanto peso se le da a las mediciones de los sensores y que tanto a las predicciones del filtro, y es recalculado en cada iteración.

## Filtro Extendido de Kalman

El Filtro de Kalman tradicional es aplicado a sistemas lineales de fusión de datos. Sin embargo, algunos fenómenos no pueden ser modelados de forma lineal y al aplicar el filtro de Kalman tradicional se introducen perturbaciones al sistema debido a la linealización [70]. Debido a esto se utiliza una extensión del filtro de Kalman conocida como el Filtro de Kalman Extendido (EKF). Este predice el siguiente vector de estados del sistema asumiendo que su modelo de estado y su modelo de observación son lineales en un intervalo de medición (Series de Taylor). Los modelos del filtro se expanden usando Jacobianos [36]. El nuevo modelo de estado y de observación se pueden representar de la siguiente forma:

$$X_{k-1} = f(X_k) + w_k \quad (1.7)$$

$$z_k = h(X_k) + v_k \quad (1.8)$$

Finalmente la matriz de relación  $H_k$  se modela como el Jacobiano de  $h$  evaluado en  $X_k$ :

$$H = \frac{\delta h}{\delta X} \quad (1.9)$$

## Proceso Gauss - Markov

Un proceso Gaussiano estacionario que tiene una auto correlación exponencial se le conoce como un proceso Gauss-Markov. La auto correlación y la función espectral de este proceso son de la forma [6]:

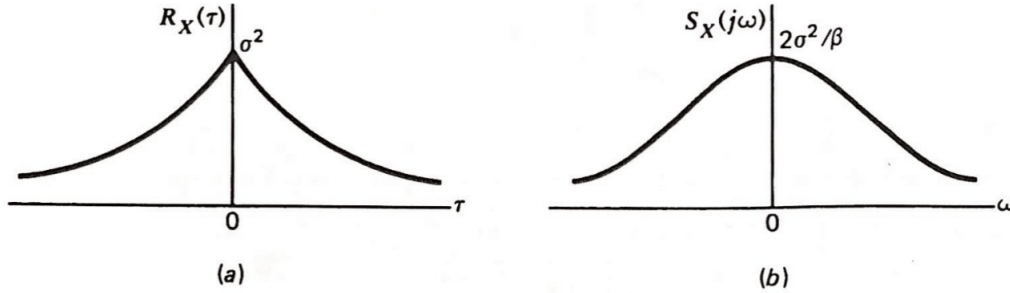
$$R_x(\tau) = \sigma^2 \exp^{-\beta|\tau|} \quad (1.10)$$

$$S_x(j\omega) = \frac{2\sigma\beta}{\omega^2 + \beta^2} \left[ \text{or } S_x(s) = \frac{2\sigma^2\beta}{-s^2 + \beta^2} \right] \quad (1.11)$$

Las gráficas de estas funciones se observan en la figura 8. El valor medio cuadrático y la constante de tiempo del proceso están dadas por los parámetros  $\sigma^2$  y  $1/\beta$  respectivamente.

El proceso Gauss-Markov ha sido utilizado por algunos autores para modelar algunas variables del vector de estado como la inestabilidad del bias del giroscopio [15], variaciones del bias [24], error del bias y error de longitud del paso [53], entre otros.

Figura 8: Auto correlación y función de densidad espectral de un proceso Gauss-Markov [6]



## Errores y modelos de medida de los sensores inerciales

Los sensores inerciales usualmente usados para localización de personas suelen ser de tecnología MEMS, que en su mayoría son de bajo costo y tamaño reducido para diseñar prototipos portables. Sin embargo por sus características se ven afectados por errores de calibración, incertidumbre, rango de medida, entre otros. Por esta razón algunos autores se han dedicado a proponer modelos matemáticos que permitan modelar estos errores y así evitar la propagación de estos por el sistema. A continuación algunos de los errores presentes en este tipo de sensores.

### Errores en los sensores inerciales

**Bias.** El bias hace referencia a un nivel en continua que suele tener los valores medidos por los sensores. Un modelo matemático para el bias es el propuesto en [12], donde se propone que el bias depende de factores como temperatura y pequeñas variaciones:

$$b = b_0 + b_1 t + b_{c1} + (b_{T1} + b_{T2} \Delta T) \Delta T \quad (1.12)$$

Donde  $b$  es el bias contenido en la medida,  $b_0$  es el bias dependiente de la gravedad,  $b_1$  es la variación del bias,  $b_{Tn}$  es el bias de orden  $n$  dependiente de la temperatura y  $b_{c1}$  es el término de sensibilidad entre ejes.

**Factor de escala.** El factor de escala hace referencia a la diferencia entre la entrada del sensor y su salida. Un modelo para este error es el propuesto en [25], el cual modela el factor de escala de la siguiente manera:

$$S = S_1 + S_2 l \quad (1.13)$$

Donde  $S_n$  hace referencia a una matriz de orden  $n$  de los errores de los factores de escala (matriz diagonal de los ejes  $x$ ,  $y$  y  $z$ ).

**No ortogonalidad.** El desalineamiento se produce por las imperfecciones en la manufactura del sensor. Esto resulta en un error aleatorio en aplicaciones que involucran cinemática [24].

**Ruido del sensor.** Este ruido por el que se ve afectada la salida del sensor se debe a factores como la calibración y el ruido propio del convertidor A-D. Este error suele modelarse como un ruido blanco con desviación estándar especificada en los archivos de control.

### Modelos matemáticos para modelar errores en los sensores inerciales

Teniendo en cuenta los errores recién presentados, a continuación se presentan algunos modelos matemáticos que pretenden cuantificar el error para poder tener el valor real de la medición. Un modelo común es el presentado en [24], donde el valor verdadero se presenta como:

$$\hat{l} = l + b + Sl + Nl + \epsilon \quad (1.14)$$

Donde  $\hat{l}$  es el valor obtenido del sensor en el marco de referencia del cuerpo,  $l$  es el valor actual de la medida,  $b$  es el vector del bias,  $S$  es la matriz de factores de escala,  $N$  es la matriz de desalineamiento y  $\epsilon$  es el vector de ruido blanco.

En sistemas donde se aplica INS, debido a la integración aparece otro error conocido como el *drift*. Este error aparece debido a la integración de medidas perturbadas por otros errores [15]:

**Bias** ( $\delta\omega$ ). Cuando se integra causa un error de crecimiento constante  $\Phi(t) = \delta\omega t$ .

**Ruido blanco del sensor** ( $\eta(t)$ ). Al integrar se conduce a un movimiento aleatorio (proceso *Brownian Motion*)  $\Phi(t) = \int_0^t \eta(\tau) d\tau$ , el cual tiene valor esperado de cero ya que es igualmente probable que se dirija en cualquier dirección, pero con un error cuadrático medio creciendo linealmente en el tiempo.

**Errores de calibración:** Desalineamiento, factores de escala generando un aumento en el *drift* con el tiempo.

En general, el error por el drift puede reducirse eliminando el valor promedio de la ventana antes de la integración [13]. Además se plantea que el error se propaga en cada paso, es decir que crece de forma lineal en cada paso y es independiente del tiempo [55].

## Métodos de corrección de errores

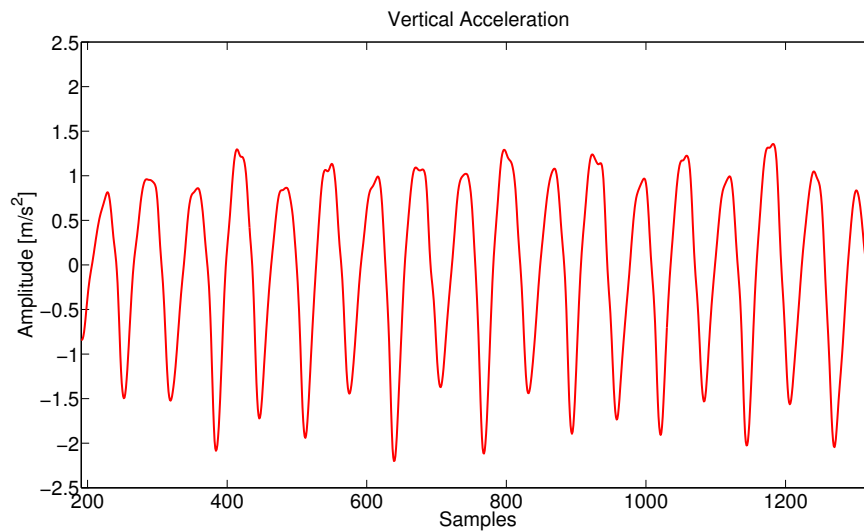
Como ya se mencionó, la mayoría de sistemas de localización que se basan en el algoritmo DR utilizan las medidas de sensores inerciales para estimar la posición. Retomando las ecuaciones 1.1 y 1.2, se evidencia que la posición se calcula basándose en dos factores: la orientación y la distancia recorrida entre dos iteraciones consecutivas. Cuando se extiende el algoritmo DR a la localización de personas, la distancia recorrida entre dos iteraciones se convierte en la longitud del paso o la longitud de zancada. Por esta razón, la investigación en este campo se ha enfocado a corregir estos dos factores y aumentar la exactitud de la estimación de posición. A continuación se mencionan algunos métodos propuestos por algunos autores para cada una de las fuentes de error.

## Corrección del error en la estimación de longitud del paso

Para la estimación de la longitud del paso, la investigación se ha enfocado a proponer métodos que presenten bajo error más que a métodos que corrijan dicha estimación. Esto es debido a que los métodos encontrados en la literatura tienen errores bajos (menores al 1%). Lo primero para estimar la longitud del paso o zancada es detectar esta. Analizando el patrón de la marcha de una persona, se evidencia un ciclo en cada paso [26]. Principalmente se destacan dos fases: la fase de apoyo y la fase de balanceo. Cada vez que se detecta una fase de apoyo, se detecta una zancada. Un método común para la detección de la fase de apoyo es comparar la señal de aceleración con un umbral previamente establecido, si la aceleración medida es mayor se detecta una fase de apoyo [69]. Otros autores han presentado métodos más robustos. En [27], se detecta la fase de apoyo con tres condiciones: a) Comparación de la aceleración con un umbral b) Comparación de la varianza local de la aceleración con un umbral c) La magnitud de la velocidad angular con cierto umbral.

Otra forma es detectando los picos negativos de la aceleración horizontal [68]. Tomando la señal de aceleración y filtrando con un filtro pasa bajas se obtiene una señal suavizada como se observa en la figura 9.

Figura 9: Señal de aceleración horizontal filtrada con un filtro pasa bajas.



Luego aplicando una compuerta de cuatro condiciones se detecta un paso: a) Aceleración tiene un valor superior a un umbral b) Cruce positivo por cero c) Cruce negativo por cero d) Tiene un pico negativo.

Luego que el paso haya sido detectado, se procede a estimar su longitud. Algunos métodos se han basado en técnicas de inteligencia artificial como son redes neuronales [10] [5] [21], lógica difusa [7] [59], entre otros. Otros autores han propuesto métodos menos robustos. Uno conocido es el propuesto en [42], donde se plantea que la longitud del paso es directamente proporcional a la frecuencia del paso:

$$SL = S_0 + m * (f_{step} - f_0) \quad (1.15)$$

A pesar que es un método que se ha probado y es exacto, su implementación en aplicaciones en tiempo

real demandan importantes recursos computacionales, por lo que otros autores han investigado métodos menos robustos pero con una exactitud aceptable. En [35] proponen un modelo donde la longitud del paso tiene una relación lineal con la raíz cúbica de un promedio de las medidas de la aceleración:

$$SL = M * \sqrt[3]{\frac{\sum_{k=1}^N |A_k|}{N}} \quad (1.16)$$

Sin embargo este método depende de un parámetro  $M$  que debe ser ajustado para cada usuario, algo poco deseable para muchas aplicaciones. Por esto, algunos autores han propuesto calcular la velocidad del pie en cada paso con el fin de poder calcular la distancia recorrida [21]. Integrando la aceleración se puede obtener la velocidad promedio en la fase de balanceo y conociendo el tiempo de duración (número de muestras) se puede calcular la distancia recorrida. La principal ventaja de este método es que no se necesita ajustar para cada usuario lo que lo hace más independiente del usuario. Pero como ya se mencionó, al integrar la aceleración se introduce un *drift* que debe ser corregido.

El método más común es conocido como Actualización de Velocidad Cero (ZUPT por sus siglas en inglés) [65] [51] [1]. Durante la fase de apoyo, la velocidad del pie es idealmente cero, por lo que el valor estimado se considera como error. Este error suele corregirse con una interpolación lineal de pesos [2] [28]. De igual forma se ha propuesto el método de Actualización de Velocidad Constante (CUPT por sus siglas en inglés), método similar al ZUPT pero tiene en cuenta que durante la fase de balanceo el pie tiene un intervalo de velocidad constante [67] [43].

Aunque la mayoría de los autores han propuesto métodos de estimación de longitud con bajo error, algunos autores lo complementan con un método de corrección de error. En [53] y [32] plantean un filtro de Kalman extendido donde una variable del vector de estado es el error en la longitud del paso y su modelo matemático es el siguiente:

$$S = \hat{S} + \epsilon_s \quad (1.17)$$

Donde  $S$  es el valor verdadero de la longitud del paso,  $\hat{S}$  es el valor calculado y  $\epsilon_s$  es el error estimado por el filtro de Kalman.

### Corrección del error en la orientación

El error en la orientación es la fuente de error crítica para corregir, pues puede causar que el sistema diverja de manera rápida y debe ser corregida permanentemente. Muchos autores han optado por usar IMUs que contengan magnetómetros y combinarlos con las medidas de velocidad angular para predecir cuando hay perturbaciones magnéticas [30] [57]. Durante el andar de una persona en línea recta, se puede evidenciar un bias que afecta su trayectoria y es de interés corregirlo. En [29] proponen un método heurístico que compara la orientación del paso actual con el de los dos pasos anteriores y si la diferencia es menor a cierto umbral, se toma esta diferencia como el error bias y se corrige:

$$\Delta\Psi_k = \Psi_k - \frac{\Psi_{k-(k_s-k_{s-1})} + \Psi_{k-(k_s-k_{s-2})}}{2} \quad (1.18)$$

Sin embargo, debido a la falta de calibración y al error que se va acumulando con el tiempo, el sistema puede llegar a divergir en aplicaciones de larga duración. Un método común es aplicar Actualización de Velocidad Angular Cero (ZUPT por sus siglas en inglés) cuando se calcula la orientación integrando la velocidad angular [16] [19]. Al igual que el ZUPT, durante la fase de apoyo la velocidad angular debe ser cero, por lo que el valor calculado se considera el error y es corregido. Sin embargo, después de la corrección se integra lo que hace que se introduzca un nuevo *drift*, el cual puede ser corregido eliminando el valor promedio de la última ventana de integración [62].

Otros métodos utilizan el *yaw* como orientación de la persona, sin embargo al ser calculado con medidas de aceleración y campos magnéticos, también se ven afectados por errores y deben ser corregidos. Una manera sencilla es mediante una interpolación lineal [45]. Además si se tiene un conocimiento a priori del valor del error, se puede aplicar una interpolación en cada paso y estimar la orientación con el siguiente modelo [53]:

$$\Psi = \hat{\Psi} - (B + \delta) \quad (1.19)$$

Donde  $\Psi$  es el valor verdadero de la orientación,  $\hat{\Psi}$  es el valor calculado por el sensor,  $B$  es el valor verdadero del bias y  $\delta$  es ruido Gaussiano. El valor verdadero del bias es modelado de la siguiente manera:

$$B = B_{sensor} + \epsilon_B \quad (1.20)$$

Donde  $B$  es el valor verdadero del bias,  $B_{sensor}$  es el bias propio del sensor y  $\epsilon_B$  es el error de la orientación estimado en cada iteración.

Aunque estos métodos pueden proporcionar una corrección aceptable, para ciertas aplicaciones no es suficiente, por lo que varios autores han aplicado filtros de Kalman para estimar el error en la orientación [60] [48] [20]. Algunos autores estiman directamente el bias de la orientación, otros estiman el error del bias [53] [32]. Otra forma común es utilizar el filtro de Kalman como fusión de datos de un sensor externo como es el GPS. Se ha demostrado que para ambientes externos se puede mejorar la estimación de posición usando sensores inerciales y fusión de datos [11].

# Capítulo 2

## DISEÑO E IMPLEMENTACIÓN DEL HARDWARE

### Introducción

Como se mencionó en el anterior capítulo, el objetivo principal de este trabajo es diseñar e implementar un prototipo que estime la posición utilizando el algoritmo PDR, aplicando un filtro de Kalman extendido como fusión de datos entre un receptor GPS y una Unidad de Medida Inercial.

En este capítulo se mencionarán los sensores seleccionados y sus características técnicas. Se explicarán los protocolos de comunicación utilizados y su principio base de funcionamiento. Finalmente se presenta el diseño desde el esquemático del circuito hasta la implementación del circuito impreso.

### Diseño del prototipo de localización

El prototipo está formado por una Unidad de Medida Inercial, un receptor GPS y una unidad de procesamiento. Para la selección de estos sensores se tuvieron en cuenta características como rango de medida, protocolos de comunicación, sensibilidad, potencia consumida, frecuencia de muestreo, entre otros.

### Sensores seleccionados

**Unidad de Medida Inercial.** Como Unidad de Medida Inercial se escogió la "*Xsens MTi 1-series Development Board with MTi-3-8A7G6 module*". Este sensor se observa en la figura 10.

Esta unidad está compuesta por tres acelerómetros, tres giroscopios y tres magnetómetros ortogonales entre sí (9 grados de libertad). Al ser una tarjeta de desarrollo, cuenta con algunos periféricos externos como un conmutador de seis niveles, luces indicadoras VDD y VDDIO, cabecero de 24 pines, entre otros. En la tabla 1 se observan las principales características de esta unidad.

Tabla 1: Especificaciones técnicas de la *Xsens MTi 1-series Development Board with MTi-3-8A7G6 module*.

Sensor	Rango	Frecuencia de muestreo	Ancho de banda
Acelerómetros	$\pm 16$ [g]	1 [KHz]	180 [Hz]
Giroscopios	$\pm 2000$ [deg/s]	1 [KHz]	180 [Hz]
Magnetómetros	$\pm 0,8$ [Gauss]	100 [Hz]	NA

Figura 10: *Xsens MTi 1-series Development Board with MTi-3-8A7G6 module.*

La unidad se polariza a 5 [V]. Los niveles lógicos están en el rango de 0 – 3,3 [V] por defecto pero pueden configurarse a 5 [V] por el pin VDDIO.

**Receptor GPS.** El receptor GPS seleccionado es el "*uBlox EVK-M8*". Este receptor se puede observar en la figura 11.

Figura 11: *uBlox EVK-M8.*

Este receptor es un kit de desarrollo, lo que permite tener conexión directo por USB al *software üCenter*". Esto permite la toma directa de datos y la configuración del mismo por medio de interfaz gráfica. Adicionalmente cuenta con un cabecero de 14 pines que permite conectarlo a una unidad de procesamiento o a otros periféricos externos adicionales. Una lista de las especificaciones técnicas se puede observar en la tabla 2.

Este sensor soporta el protocolo de mensajes NMEA que es común para los receptores GPS. Además se puede configurar para un protocolo propio de la marca conocido como protocolo UBX.

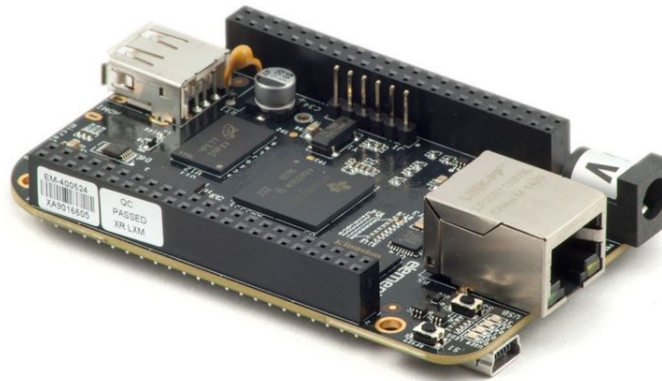
## Unidad de Procesamiento

Como unidad de procesamiento se escogió la tarjeta de desarrollo *Beaglebone Black*. Esta tarjeta se puede observar en la figura 12.

Esta unidad de procesamiento es una tarjeta de desarrollo con su propio sistema operativo (Linux).

Tabla 2: Especificaciones técnicas del receptor *uBlox EVK-M8*.

Característica	Descripción
Nivel de polarización	5 [V]
Protocolos de comunicación	I2C, SPI, UART y Dúplex UART
Dimensiones	105 x 64 x 26 [mm]
Temperatura normal de operación	-40°C a +65°C
Niveles lógicos de polarización	0 – 3,3 [V]

Figura 12: Tarjeta de desarrollo *Beaglebone Black*.

Normalmente viene configurada con la imagen de Debian aunque puede instalarse Ubuntu. Tiene comunicación por consola o por interfaz gráfica. Además cuenta con pines GPIOs que permiten conectar diferentes periféricos externos como displays, ADC, entre otros. Además soporta protocolos de comunicación seriales como SPI, I2C, USB, UART y UART duplex. Una lista con algunas especificaciones técnicas se observa en la tabla 3.

Tabla 3: Especificaciones técnicas del receptor *uBlox EVK-M8*.

Característica	Descripción
Procesador	AM335x 1[GHz] ARM® CORTEX – A8
Memoria SDRAM	512MB DDR3L 800MHz
Memoria FLASH	4GB 8bit MMC embebido
Nivel de polarización	5 [V]
Dimensiones	120 x 67 [mm]
Sistema operativo	Linux, soporta Debian y Ubuntu
Conectividad	USB client, USB host, Ethernet, HDMI

La tarjeta permite programación en lenguajes de python, C, C++, Arduino, entre otros. Además cuenta con algunos pines pre configurables, lo cual permite que funcionen como PWM, GPIO, entre otros.

## Protocolos de comunicación

Para comunicación con estos sensores, protocolos seriales son comúnmente usados. Entre los protocolos más destacados se encuentra SPI, I2C y UART. En este trabajo la comunicación entre la unidad de procesamiento, el receptor GPS y la unidad de medida inercial se realizó mediante UART. A continuación se explica en detalle el funcionamiento de este.

**Protocolo UART.** Este protocolo es uno de los más comunes para intercambio de datos. En su forma más usada se envían mensajes de 8 bits y uno adicional para describir la dirección (Ver figura 13). Además es un protocolo punto a punto, es decir solo dos elementos intercambian información entre si. Físicamente consiste únicamente de dos hilos de datos, el primero para la transmisión de información (Tx) y el otro para la recepción (Rx).

Figura 13: Mensaje de protocolo UART.

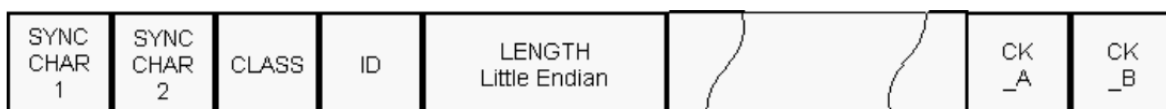


Los bits de inicio y parada indican el comienzo y el final de una trama de datos. El bit de paridad es opcional y es usado para revisar si algún tipo de error ocurrió durante la transmisión del mensaje. Además se caracteriza por ser una comunicación asíncrona pues se necesita del cambio de un nivel alto lógico de tensión a un nivel bajo para comenzar la comunicación entre partes.

## Configuración de los sensores

**Protocolos uBlox y Xsens.** La unidad de procesamiento debe tener comunicación con el receptor GPS y la unidad de medida inercial. Sin embargo, los fabricantes de ambos dispositivos seleccionados en este proyecto utilizan un protocolo de envío de mensajes propio. Para el receptor uBlox se utiliza el protocolo UBX. Este se utiliza para transmitir datos GNSS. Principalmente tiene tres características: a) Utiliza mensajes compactos de 8 bits b) Tiene detección de errores con el campo *checksum* c) Es modular pues la trama se divide en dos partes (clase y mensaje ID). La estructura de una trama puede observarse en la figura 14.

Figura 14: Estructura de una trama de datos con el protocolo UBX.

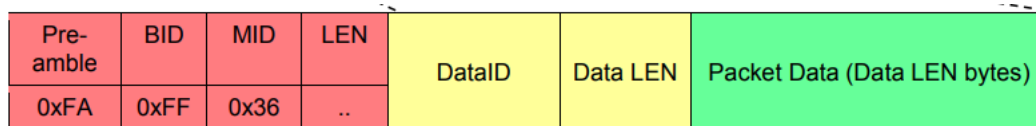


A continuación una explicación de cada uno de los campos de la trama:

- ❖ **Caracter de sincronización 1:** Siempre vale 0xB5 (1 byte)
- ❖ **Caracter de sincronización 2:** Siempre vale 0x62 (1 byte)
- ❖ **Clase:** Define la clase del mensaje (1 byte)
- ❖ **ID:** Indica el ID del mensaje (1 byte)
- ❖ **Longitud:** Tamaño de la trama del mensaje que se va a recibir o a enviar. (2 bytes).
- ❖ **Mensaje:** Contiene la información que se desea enviar o recibir (N bytes).
- ❖ **Checksum A:** Verificación de mensaje 1 (1 byte)
- ❖ **Checksum B:** Verificación de mensaje 2 (1 byte)

Para la Unidad de medida inercial, el protocolo para envío y recepción de mensajes se llama "*Low-level communication protocol*". Este protocolo es propio de Xsens y puede ser implementado en cualquier protocolo de comunicación serial. Un diagrama que muestra la estructura de un mensaje se puede observar en la figura 15.

Figura 15: Estructura de una trama de datos con el protocolo XS



A continuación se explica cada uno de los campos dentro de una trama:

- ❖ **Preámbulo :** Indicador de paquete. Siempre es 0xFA.
- ❖ **BID :** Identificador del bus. Siempre es 0xFF.
- ❖ **MID :** Identificador del mensaje a enviar o recibir.
- ❖ **LEN :** Longitud del mensaje en bytes.
- ❖ **Datos :** Mensaje que se desea enviar o recibir por parte de la unidad.
- ❖ **Checksum :** Verificación de errores en el mensaje.

El listado completo de los identificadores de mensajes e identificadores de clases de ambos protocolos se encuentran en el apéndice A.

## Configuración del prototipo

**Unidad de procesamiento.** Para la comunicación con los sensores, la unidad de procesamiento se configuró para que algunos pines específicos trabajaran bajo el protocolo de comunicación UART. En la tabla 4 se encuentra la lista de pines utilizados de la tarjeta de desarrollo.

Estos pines corresponden a la fila de pines P9.

Tabla 4: Pines utilizados de la tarjeta de desarrollo.

Pin	Descripción
1	GND GPS
2	GND Xsens
3	Polarización VDDIO Xsens
7	Polarización 5 [V] GPS
8	Polarización 5 [V] Xsens
21	UART Tx GPS
22	UART Rx GPS
24	UART Tx Xsens
26	UART Rx Xsens

**Unidad de Medida Inercial.** La unidad fue configurada a una frecuencia de muestreo de 100 [Hz]. El formato de los datos a leer es punto flotante IEEE de 32 bits de precisión simple. En la tabla 5 se mencionan las medidas tomadas de cada uno de los sensores.

Tabla 5: Medidas tomadas de la IMU Xsens.

Fuente	Medidas
Acelerómetros	$a_x, a_y, a_z$
Giroscopios	$w_x, w_y, w_z$
Magnetómetros	$h_x, h_y, h_z$
Ángulos de Euler	$yaw, pitch, roll$
Time stamp	$TS$

Los ángulos de Euler y el *time stamp* son datos que el sensor entrega pero que no son proporcionados por ningún sensor, es decir que corresponden a procesamientos internos de la unidad. No todas las IMUs comerciales brindan esta información. Esto es ventajoso ya que permite un ahorro en el algoritmo implementado lo que se evidencia en eficiencia, menor tiempo de ejecución y por ende menor consumo de potencia.

**Receptor GPS.** La lectura del receptor GPS se basa en la lectura de dos líneas: RMC y GSA. En la primera se encuentran los datos de latitud, longitud, UTC, entre otros. En la segunda línea se encuentran los errores de pseudo rango. En la tabla XX se muestra la lecturas de cada una de estas líneas.

Tabla 6: Medidas tomadas del receptor GPS uBlox.

Fuente	Medidas
RMC	$RMC ID, UTC, vlido, lat, Norte/Sur, Lon, Este/Oeste, velocidad, curso, fecha, checksum$
GSA	$GSA ID, mododeoperacin, satlites, PDOP, HDOP, VDOP$

El receptor entrega los valores de latitud y longitud en formato de horas, minutos y fracciones de

minutos. Para poder hacer el procesamiento planteado en este trabajo es necesario convertir estas medidas a formato decimal. La frecuencia de muestreo seleccionada es de 1 [Hz], sin embargo hay que aclarar que este receptor puede configurarse hasta 5 [Hz], algo que solo algunos receptores traen.

## Integración componentes del prototipo

Teniendo en cuenta que se desea un prototipo portable, el diseño de un circuito impreso que permita la conexión física entre los pines de la unidad de procesamiento y los sensores es necesario. Además se requiere de un sistema de polarización que permita energizar la unidad de procesamiento y por ende los demás periféricos. A continuación se hace una explicación del diseño completo del *hardware*.

**Sistema de polarización.** Para energizar el circuito se utilizó una batería de 9V. De las características técnicas de la tarjeta de desarrollo, la tensión de polarización de ser de 5V. Por esta razón se utilizó un regulador de tensión LM7805 que permite regular la tensión de entrada a una tensión de salida de 5V y 1A. Algunas características de este elemento se observan en la tabla 7.

Tabla 7: Especificaciones técnicas del regulador 7805.

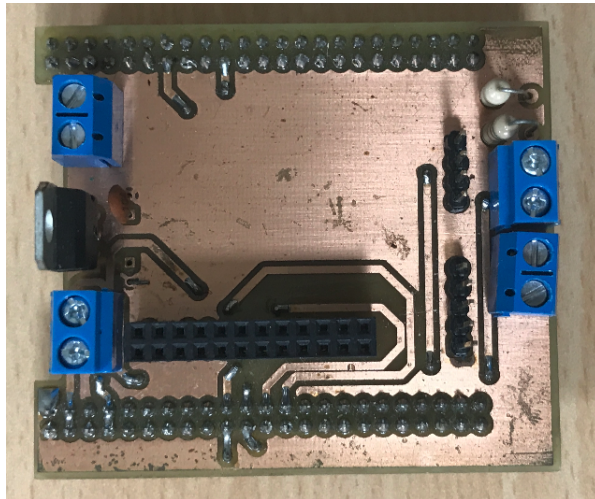
Característica	Descripción
Tensión de salida	min: 4,8 [V] max: 5,2 [V]
Drift de la tensión de salida	-0,8 [mV/°C]
Resistencia de salida	15 [mΩ]

Entre las precauciones más importantes de este elemento es no superar los 35 [V] en el pin de entrada, que para este prototipo es suficiente pues la batería para energizar es mucho menor. Además la conexión física cuenta con dos capacitores en la entrada y la salida para evitar ruidos y perturbaciones en los niveles de tensión.

**Circuito impreso.** En el diseño del circuito impreso se tuvieron que tener en cuenta algunos aspectos de los sensores. El primero es que el receptor GPS es grande por lo que debería estar ubicado en otro lugar. Una segunda observación es que la unidad de medida inercial para algunas pruebas irá en la cintura y para otras en el pie. Finalmente tener en cuenta que la tarjeta de desarrollo puede generar interferencia magnética con los magnetómetros de la unidad. Por esta razón se decidió dejar pines de conexión sobre la PCB y conectar por cable con el fin de reducir la perturbación magnética y lograr ubicar los sensores donde se desee. El circuito terminado e implementado se puede observar en la figura 16.

Como observación final se menciona que las pruebas realizadas no se realizaron con el sistema de polarización. Esto porque se energizó por medio de conexión USB con un computador con el fin de ir observando en pantalla los resultados obtenidos. Sin embargo el sistema fue probado y sus resultados fueron los deseados.

Figura 16: Circuito del prototipo terminado e implementado.



## Capítulo 3

# IMPLEMENTACIÓN DE LOS ALGORITMOS DE LOCALIZACIÓN

### Introducción

El objetivo principal de este trabajo de investigación es implementar un sistema de localización para personas que funcione en tiempo real, con sensores inerciales y un módulo receptor GPS. Además, deben seleccionarse e implementarse los métodos de corrección de errores necesarios para evitar que la estimación de posición diverja. Para ello, es necesario identificar las principales fuentes de error que van desde los errores producidos por los sensores hasta los introducidos por los modelos matemáticos.

En este capítulo se presentan las ecuaciones utilizadas para la estimación de posición utilizando Navegación por Estima para Personas (PDR, por sus siglas en inglés), y se realiza una comparación de métodos de corrección de errores para la detección del paso, longitud del paso y orientación. Se describen además las ventajas y desventajas de cada uno de estos métodos. Finalmente se muestran los resultados comparativos y se describe el algoritmo final de localización implementado en tiempo real.

### Estimación de posición aplicando PDR

Como ya se mencionó en el capítulo 2, una forma de estimación de posición es utilizando el algoritmo DR. Teniendo en cuenta el hecho que una persona al caminar evidencia un patrón en las señales obtenidas de los sensores inerciales, se pueden detectar pasos e incluso la distancia recorrida paso a paso. Un ejemplo de este patrón se evidencia en la figura 17. Estas señales corresponden a las aceleraciones en tres ejes evidenciadas durante la caminata de un sujeto, de las cuales se puede extraer diferentes características como frecuencia del paso, longitud del paso, detección del paso, entre otras.

Extendiendo el algoritmo DR para personas se puede plantear que la distancia recorrida en dos intervalos de tiempo consecutivos (frecuencia del paso o zancada) es la longitud del paso o de zancada de la persona, y conociendo su orientación se puede plantear el algoritmo PDR. Un gráfico de esto se puede ver en la figura 18.

La posición final puede calcularse aplicando las ecuaciones 3.1 y 3.2.

Figura 17: Patrón de las tres aceleraciones de una persona al caminar.

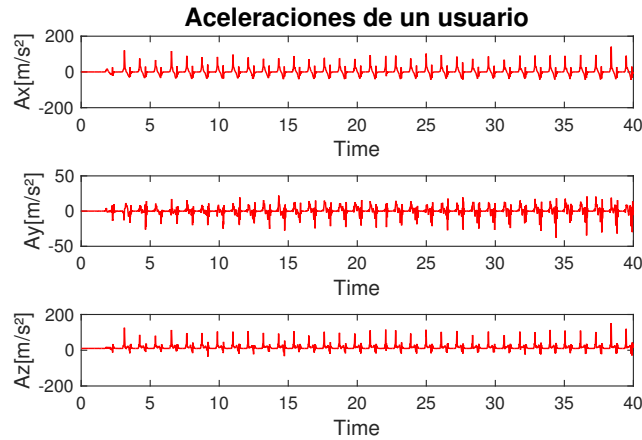
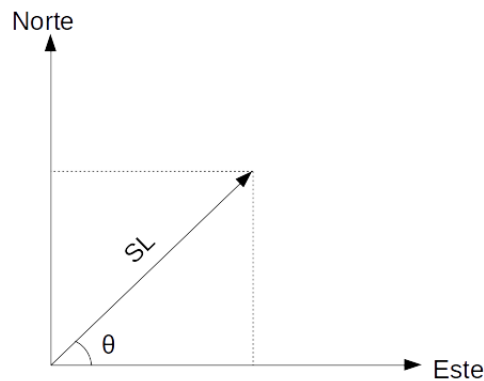


Figura 18: Gráfica de estimación de posición de personas con el algoritmo PDR.



$$X = X_0 + \sum_{k=0}^N SL_k * \cos(\Psi_k) \quad (3.1)$$

$$Y = Y_0 + \sum_{k=0}^N SL_k * \sen(\Psi_k) \quad (3.2)$$

Donde  $X_0$  y  $Y_0$  son la posición inicial, y  $\Psi_k$  y  $SL_k$  son la orientación y la longitud del paso  $k$  respectivamente. Sin embargo, si la estimación de la longitud del paso y la orientación no se corrige, la posición calculada va a divergir. Por esto, métodos de corrección de errores de la longitud del paso y de la orientación deben ser aplicados.

## Métodos de corrección de errores para mejorar la estimación de parámetros del PDR

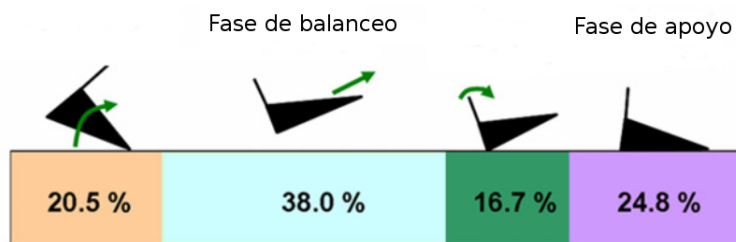
De las ecuaciones 3.1 y 3.2, los dos parámetros que deben ser calculados en cada paso son la longitud y la orientación de este. Sin embargo, para estimar la longitud del paso, primero debe ser este detectado. A continuación se presentan los métodos de detección del paso, estimación de longitud del paso y orien-

tación implementados en este trabajo junto a su comparación. Para la estimación de longitud del paso se implementaron métodos que presentan bajo error. Por su parte para la orientación si hubo necesidad de implementar métodos de corrección de error.

### Métodos de detección del paso

El primer método implementado para detectar el paso es la detección de la fase de apoyo del ciclo del paso [8] [14] [44]. En la figura 19, se observa el ciclo del paso de una persona. Hay una etapa donde el pie se encuentra completamente quieto en el suelo, la cual se conoce como la fase de apoyo. Detectar esta fase es detectar que la persona dio un paso. Para esto, la unidad debe ir ubicada en el pie del sujeto, por lo que cada vez que se detecte una fase de apoyo, se detecta una zancada.

Figura 19: Ciclo del paso. Adaptada de [18].

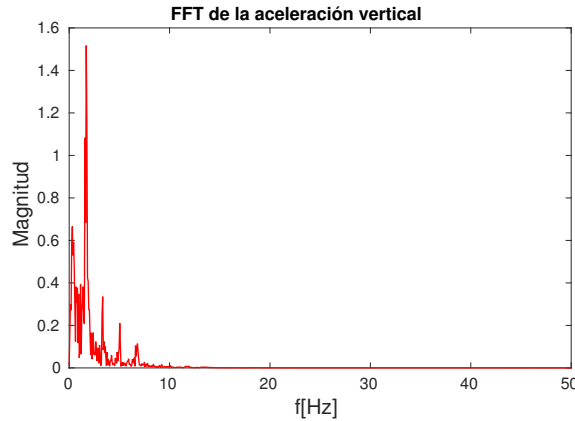


Para la detección de la fase de apoyo se tomaron las medidas de aceleración en los tres ejes, se calculó su magnitud y se comparó con un umbral ( $3,2 [m/s]$ ). Si hay 15 muestras consecutivas que tienen un valor por debajo del umbral se considera que el ciclo del paso se encuentra en la fase de apoyo y un paso es detectado justo en la primera muestra que sobrepase el umbral.

El segundo método de detección del paso es basándose en la aceleración vertical. Cada vez que la persona da un paso, el impacto del pie con el suelo genera un pico negativo que puede ser detectado. Para esto el sensor debe estar ubicado en la espalda baja del sujeto. Sin embargo, el ruido con el que viene la señal puede hacer que el algoritmo detecte pasos que en verdad no se han dado. Para evitar esto, se analizó la señal en el dominio de la frecuencia y así determinar el componente principal. En la figura 20 se puede observar la transformada de Fourier. Es evidente que el componente principal de frecuencia se encuentra a una frecuencia menor de  $4 [Hz]$ . Por esta razón, la señal es suavizada a través de un filtro pasa bajas de frecuencia de corte de  $4 [Hz]$ .

Adicionalmente, analizando la señal de aceleración horizontal, el comportamiento es similar e inclusive su patrón es más notorio. Por esta razón se optó por trabajar con la aceleración horizontal (misma dirección de movimiento del sujeto) y no con la vertical. La señal de aceleración horizontal filtrada es la misma mostrada en el capítulo 2 (Figura 9). Finalmente para detectar un paso se implementó un operador lógico AND de cuatro condiciones: 1) Detectar cruce positivo por cero b) Detectar cruce negativo por cero c) Aceleración debe superar cierto umbral d) Detectar pico negativo. Si estas cuatro condiciones se cumplen, un paso es detectado. A diferencia del primer método, este al tener la unidad ubicada en la cintura detecta pasos y no zancadas.

Figura 20: Transformada de Fourier de la señal de aceleración vertical.



### Métodos de estimación de longitud del paso

Tres métodos de estimación de longitud del paso fueron implementados para ser comparados entre sí: a) Método basado en la frecuencia del paso b) ZUPT c) Método basado en un parámetro de proporcionalidad.

**Método basado en la frecuencia del paso.** El primero es el propuesto en [42]. Ellos plantean un modelo donde la longitud del paso depende linealmente de la frecuencia del paso. La relación se muestra en la ecuación 3.3:

$$SL = S_0 + m(f_{step} - f_0) \quad (3.3)$$

Donde  $SL$  es la longitud del paso,  $m$  es la pendiente de la recta o el coeficiente de proporcionalidad,  $f_{step}$  es la frecuencia del paso y  $S_0$  es la longitud del paso para una frecuencia  $f_0$ . Para poder determinar esta recta, se realizó una prueba preliminar con cada participante. Se le solicitó al sujeto que caminara 30 pasos en línea recta. Lo primero fue calcular la longitud del paso utilizando la ecuación 3.4.

$$SL = \frac{steps}{d} \quad (3.4)$$

Donde  $SL$  es la longitud del paso,  $steps$  es el número de pasos que para esta prueba siempre fue 30 y  $d$  fue la distancia recorrida. Esta prueba se repitió dos veces para cada frecuencia de caminata (lenta, normal y rápida). Una recopilación de estos datos se observan en la tabla 8.

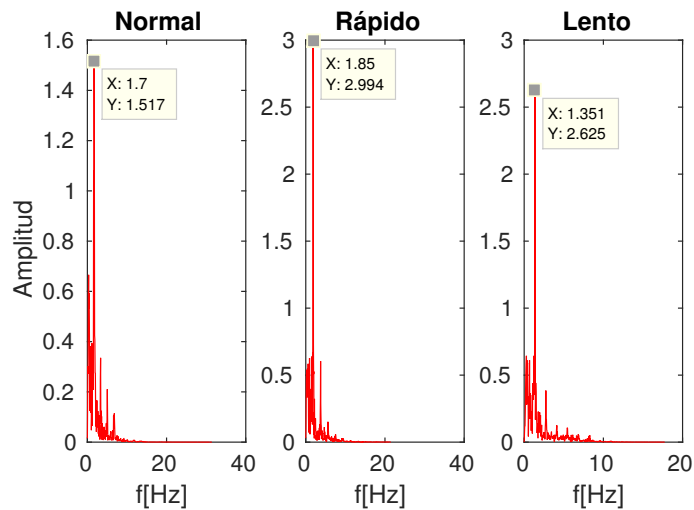
Para determinar la frecuencia de caminata, se almacenaron los datos de aceleración y de manera *offline* se calculó la Transformada Rápida de Fourier (FFT por sus siglas en inglés). Una gráfica de esto se observa en la figura 21.

Teniendo la frecuencia de caminata y la longitud del paso, se procedió a establecer la ecuación de la recta (Ecuación 3.5) mediante una interpolación lineal.

Tabla 8: Estimación de longitud del paso y frecuencia del paso.

Frecuencia	Pasos	Distancia [m]	Longitud del paso [m]	Frecuencia del paso [Hz]
Lento	30	14,77	0,492	1,351
Lento	30	15,32	0,511	1,432
Normal	30	17,39	0,58	1,7
Normal	30	17,56	0,585	1,717
Rápido	30	20,26	0,676	1,85
Rápido	30	20,22	0,674	1,958

Figura 21: FFT para determinar la frecuencia del paso.



$$SL = 0,049 + 0,322f_{step} \quad (3.5)$$

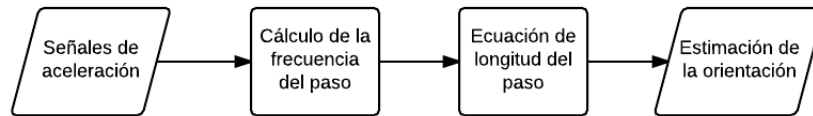
Finalmente reemplazando para una frecuencia de  $1,8Hz$  se obtiene una longitud del paso de  $0,6286$ , y la ecuación implementada para estimación del paso en tiempo real es:

$$SL = 0,6286 + 0,322(f_{step} - 1,8) \quad (3.6)$$

Hay que aclarar que esta ecuación es únicamente para un participante. Este procedimiento se llevó a cabo para cada sujeto que participó en las pruebas. Un diagrama de flujo de su implementación se observa en la figura 22.

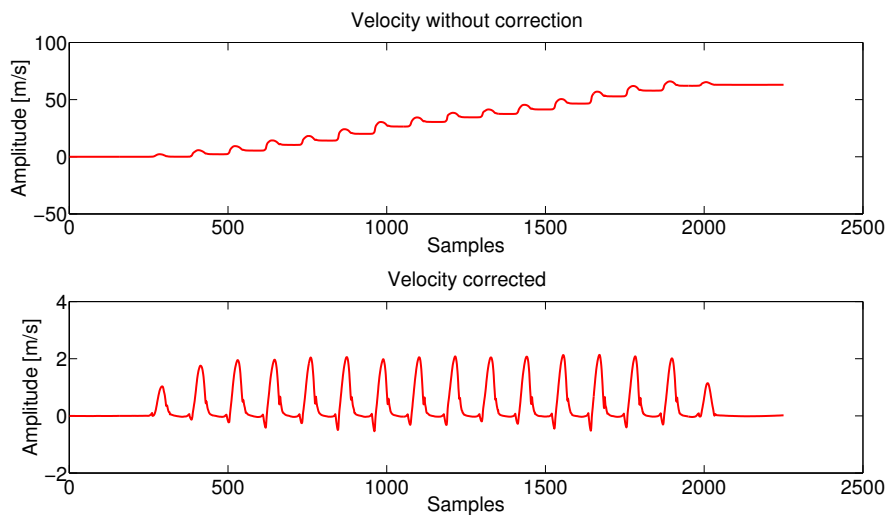
**Actualización de Velocidad Cero.** El segundo método implementado consiste en conocer la velocidad promedio del pie en la fase de balanceo, y conociendo el tiempo de duración (número de muestras) determinar la distancia de desplazamiento del pie, es decir la longitud de la zancada. Para determinar la velocidad promedio, lo primero que se debe hacer es detectar el inicio y el final de la fase de balanceo del

Figura 22: Diagrama de flujo de la estimación de longitud del paso basado en la frecuencia.



pie. Uno de los métodos de detección del paso implementados en este trabajo fue la detección de la fase de apoyo, por lo que el resto del ciclo se considera fase de balanceo y estas fueron las muestras de aceleración almacenadas en cada paso. Luego de tener la aceleración, se integró para obtener una ventana de muestras de velocidad, y a estas se les calculó el valor promedio.

Sin embargo, como ya se mencionó, al integrar se introduce un error de *drift* el cual debe ser corregido. Para esto se aplica ZUPT, el cual se explicó en el capítulo 2. En la figura 23 se puede observar la señal de velocidad de 16 pasos sin corregir y corregida por medio de ZUPT.

Figura 23: a) Señal de velocidad perturbada por el *drift*. b) Señal de velocidad corregida por ZUPT.

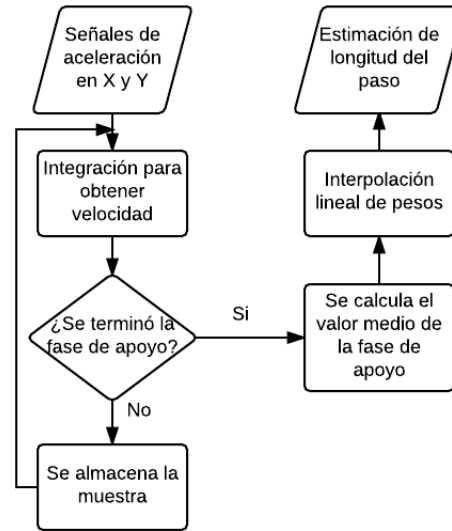
Después de tener la señal de velocidad corregida, la segunda parte consiste en calcular la longitud de zancada. Durante la fase de balanceo se cuenta el número de muestras, y conociendo la frecuencia de muestreo se calcula la distancia recorrida por el pie por medio de la ecuación 3.7.

$$SL_k = V_k \frac{\text{samples}}{f_s} = V_k(\text{samples} * T_s) \quad (3.7)$$

Donde  $SL_k$  es la longitud de la zancada  $k$ ,  $\text{samples}$  es el número de muestras durante la etapa de balanceo y  $f_s$  es la frecuencia de muestreo que para este trabajo fue de 100Hz. Para este método el sensor se ubica en el pie del usuario. Un diagrama de bloques de este método se observa en la figura 24.

**Método basado en un parámetro de proporcionalidad.** El último método es el presentado en [35]. Este método propone que la longitud del paso se calcula con la raíz cúbica del promedio de la magnitud

Figura 24: Diagrama de flujo de la estimación de longitud del paso aplicando ZUPT.

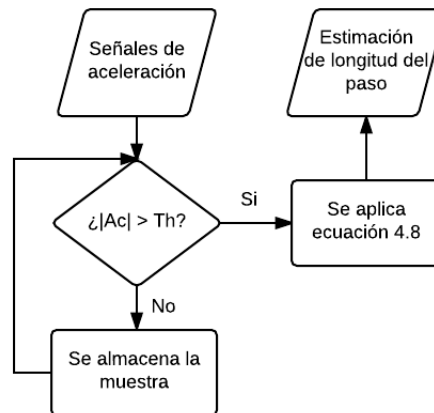


de la aceleración en una ventana específica y un parámetro de proporcionalidad el cual debe ser ajustado para cada usuario. La ecuación para la estimación de longitud del paso se observa en la ecuación 3.8.

$$SL = M * \sqrt[3]{\frac{\sum_{k=1}^N |A_k|}{N}} \quad (3.8)$$

Donde  $SL$  es la longitud del paso,  $A_k$  es la aceleración de la muestra  $k$ ,  $N$  es el número de muestras de la ventana y  $M$  es el parámetro de proporcionalidad el cual debe ser ajustado para cada usuario. En este trabajo no se tuvo un tamaño fijo de ventana, pues este lo determinaba la duración del paso. El sensor para este método se ubica en la espalda baja del usuario. Un diagrama de flujo donde se observa la estimación de longitud del paso aplicando este método se observa en la figura 25.

Figura 25: Diagrama de flujo de la estimación de longitud del paso basado en un parámetro de proporcionalidad.



## Estimación de orientación y métodos de corrección de errores

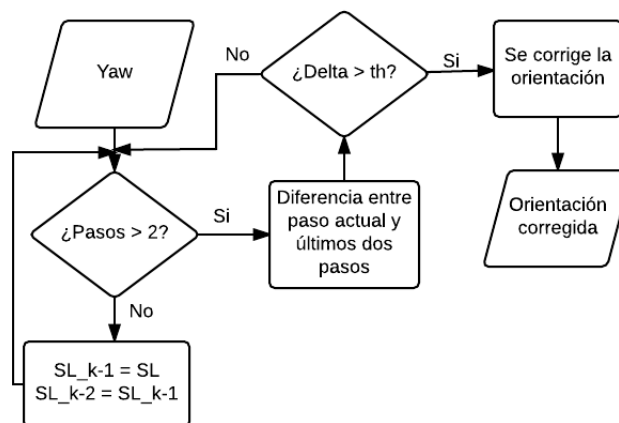
Para estimar la orientación se utilizaron dos métodos. El primero fue mediante integración de las señales de velocidad angular. El segundo fue utilizar el *yaw* que arroja el sensor. En este trabajo se implementaron tres métodos de corrección de errores: a) Reducción Heurística en la Orientación (HHR por sus siglas en inglés) b) ZARU c) Modelo del *yaw*. Además, la orientación se calculó de dos maneras: la primera fue tomando el *yaw* que entrega el sensor, y la segunda fue integrando la velocidad angular del eje que apunta en la misma dirección que el usuario estuviera caminando. A continuación se hace una explicación de cada uno de los métodos y su implementación.

**Reducción Eurística en la Orientación.** Este método es propuesto en [29]. El método consiste en corregir la orientación cuando la persona se encuentra caminando en línea recta, comparando la diferencia entre la orientación actual y el promedio de la orientación de los últimos dos pasos con un cierto umbral. Si esta diferencia es menor al umbral se considera que el sujeto camina en línea recta y se procede a corregir la orientación, pero si la diferencia es mayor, se considera que la persona está haciendo algún tipo de giro. El método principalmente se usa junto con un filtro de Kalman, sin embargo, en este trabajo se calcula el error y se corrige por medio de una interpolación. El error se calcula aplicando la ecuación:

$$\Delta\Psi_k = \Psi_k - \frac{\Psi_{k-(k_s-k_{s-1})} + \Psi_{k-(k_s-k_{s-2})}}{2} \quad (3.9)$$

Donde  $\Delta\Psi_k$  es la diferencia entre la orientación del paso actual y el promedio de los dos anteriores,  $\Psi_k$  es la orientación del paso actual,  $\Psi_{k-(k_s-k_{s-1})}$  es la orientación del paso anterior y  $\Psi_{k-(k_s-k_{s-2})}$  es la orientación del penúltimo paso. Este método se probó con el sensor en la espalda baja del usuario y utilizando el *yaw* como orientación. Un diagrama de flujo de este método se observa en la figura 26.

Figura 26: Diagrama de flujo del método de corrección de error de la orientación HHR.

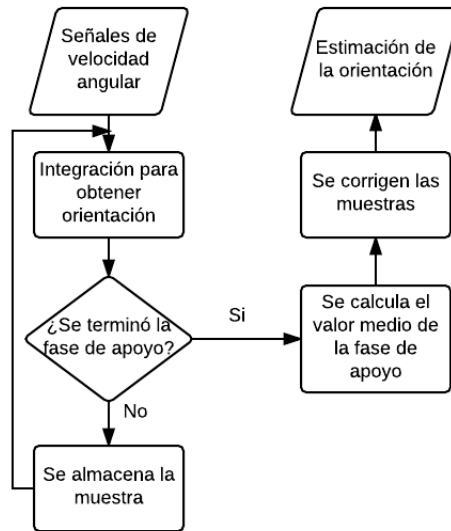


**Actualización de Velocidad Angular Cero.** Este método ha sido propuesto por varios autores [16] [30] [55]. Es similar al ZUPT, toma en cuenta que durante la fase de apoyo el pie está quieto, por lo que la velocidad angular al igual que las aceleraciones deben ser idealmente cero, por lo que el valor medido es

considerado el error y debe ser corregido. La fase de apoyo para este método se detectó con la velocidad angular y el umbral utilizado fue de  $2rad/s$ .

Sin embargo, al hacer la corrección y luego integrar para calcular la orientación, un *drift* es introducido a la señal. Esto se puede evitar quitando el valor promedio de la ventana de datos [63]. Un diagrama de flujo que muestra este método se observa en la figura 27.

Figura 27: Diagrama de flujo del método de corrección de error de la orientación ZARU.



Además a lo anterior, se hace una comparación con el *yaw* proporcionado por el sensor corrigiendo el bias. Esto se hace teniendo en cuenta que todas las medidas proporcionadas por sensores inerciales se ven afectadas por un bias, el cual debe ser corregido. Hay que aclarar que no se trata de un método de corrección de error, sino de un modelo para tratar con medidas inerciales que se ven perturbadas por un bias. La ecuación para esta corrección se observa en 3.10.

$$\Psi = \hat{\Psi} - (B + \delta) \quad (3.10)$$

Donde  $\Psi$  es la orientación verdadera,  $\hat{\Psi}$  es la orientación obtenida del sensor,  $B$  es el bias verdadero y  $\delta$  es ruido Gaussiano. El bias verdadero se modela con la ecuación 3.11.

$$B = B_{sensor} + \epsilon_B \quad (3.11)$$

Donde  $B$  es el bias verdadero,  $B_{sensor}$  es el bias del sensor y  $\epsilon_B$  es el error del bias. Varias pruebas fueron realizadas para establecer el bias del sensor. El error del bias se estima en cada iteración. Finalmente esta orientación se compara con los dos métodos de corrección implementados previamente.

## Estimación de errores utilizando Filtro de Kalman

El filtro de Kalman suele aplicarse a fenómenos lineales, sin embargo, el patrón de la marcha de una persona no lo es. Por esta razón, en este trabajo se implementó un filtro de Kalman extendido que permite modelar las no linealidades aplicando Jacobianos y series de Taylor.

### Filtro de Kalman Extendido

El modelo utilizado en este trabajo es el planteado en [53]. Este filtrado es usado para estimar el error en la longitud del paso y el del bias de la orientación, utilizando la información del receptor GPS y la IMU. Esto permite corregir la longitud del paso y la orientación para determinar los valores verdaderos. Lo primero es definir el vector de estados:

$$x = [\Delta E \ \Delta N \ \epsilon_s \ \epsilon_B] \quad (3.12)$$

Donde  $\epsilon_s$  y  $\epsilon_B$  son el error de la longitud del paso y el bias de la orientación respectivamente. Los primeros dos términos se determinan de la siguiente manera:

$$\Delta E = PE_k - PE_{k-1} \quad (3.13)$$

$$\Delta N = PN_k - PN_{k-1} \quad (3.14)$$

Donde  $PE$  y  $PN$  representa la posición Este y Norte respectivamente. Para terminar de definir el modelo de estado se debe definir la matriz de transición:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-\beta_s \Delta t} & 0 \\ 0 & 0 & 0 & e^{-\beta_B \Delta t} \end{bmatrix} \quad (3.15)$$

El error en la longitud del paso y el bias de la orientación se modelaron como procesos Gauss-Markov, donde  $\beta_s^{-1}$  y  $\beta_B^{-1}$  son las constantes de tiempo de cada proceso. El  $\Delta t$  es proporcionado por el GPS, es decir 1 segundo. Lo siguiente es establecer el modelo de medida. El vector de medida está dado por:

$$y = [\hat{\Delta E} \ \hat{\Delta N} \ \hat{s} \ \hat{\Psi}] \quad (3.16)$$

Donde  $\hat{\Delta E}$  y  $\hat{\Delta N}$  se calculan usando las medidas del GPS,  $\hat{s}$  es la longitud del paso calculada con cualquiera de los métodos ya mencionados y  $\hat{\Psi}$  es la orientación obtenida de la IMU. La función no lineal está definida como:

$$y = [\Delta E \ \Delta N \ S_{GPS} - \epsilon_s \ atan(\Delta E / \Delta N) - B] \quad (3.17)$$

Donde  $S_{GPS}$  es la distancia calculada entre dos puntos consecutivos de la información del GPS, B es definido en la ecuación 3.11. La matriz de observación se define como el Jacobiano de h:

$$H = \frac{\delta h}{\delta x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\Delta E}{\sqrt{\Delta E^2 + \Delta N^2}} & \frac{\Delta N}{\sqrt{\Delta E^2 + \Delta N^2}} & -1 & 0 \\ \frac{\Delta N}{\Delta E^2 + \Delta N^2} & \frac{-\Delta E}{\Delta E^2 + \Delta N^2} & 0 & -1 \end{bmatrix} \quad (3.18)$$

Donde las variables contenidas en la matriz de observación se refieren al vector de estado predicho por el filtro. El Bias del sensor normalmente es calculado como la diferencia en la orientación del GPS y la IMU en los primeros cinco pasos, sin embargo pruebas severas demostraron que este cálculo hacía que la estimación de la orientación no converja, por esta razón, el bias del sensor se calculó por tramas. Durante el recorrido de la persona, el bias del sensor se calcula como el promedio de todas las diferencias de orientación entre los dos sensores. Cuando se detecta un giro, este bias se pone en cero y se comienza a recalcular hasta que se detecte un nuevo giro. Esto hace que el cálculo del bias sea por tramos.

### Ajuste del modelamiento de errores de la matriz de transición (modelo Gauss-Markov)

Refiriéndose a la matriz de transición (Ecuación 3.15), el error en la longitud del paso y en el bias de la orientación se modelan como modelos Gauss-Markov de primer orden con constantes de tiempo  $\beta_s^{-1}$  y  $\beta_B^{-1}$  respectivamente. Sin embargo, estas constantes son fijas y algunos autores han propuesto que un cálculo dinámico de estas variables podría mejorar el desempeño del filtro. Por esta razón, en este trabajo se propuso determinar un banco de constantes de tiempo para cada variable del vector de estado que se modela como proceso Gauss-Markov (error en longitud del paso y en el bias de la orientación).

**Constante de tiempo para el error en el bias de la orientación.** Para determinar las diferentes constantes de tiempo, se realizaron pruebas preliminares que concluyeron en que el factor que más afectaba el desempeño eran las perturbaciones magnéticas del ambiente. Una explicación para esto puede ser que la orientación es el *yaw* obtenida de la IMU y corregida con las ecuaciones 3.10 3.11, por lo que perturbaciones magnéticas afectan de manera drástica su exactitud.

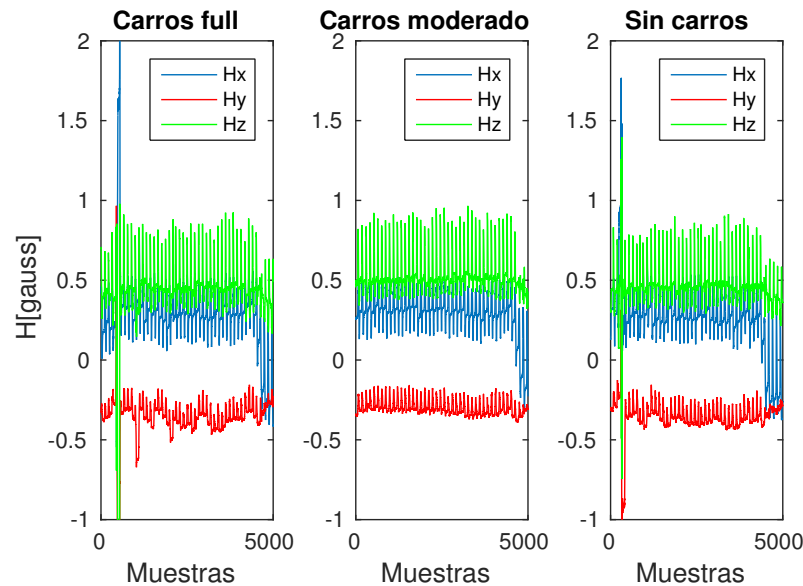
Teniendo en cuenta lo anterior, se hicieron distintas pruebas siempre bajo la misma ruta de referencia (Figura 28), esto con el fin de evitar otro tipo de perturbaciones magnéticas externas. Las pruebas fueron realizadas en el campus de la Universidad Industrial de Santander en el parqueadero bajo tres condiciones: a) Sin carros b) Parcialmente ocupado por carros c) Lleno de carros. Esto quiere decir que las perturbaciones magnéticas consideradas fueron únicamente las generadas por los carros.

Figura 28: Ruta de referencia.



Lo primero fue analizar las medidas de los magnetómetros para los tres escenarios. Una gráfica de estas señales se pueden observar en la figura 29. Para los tres escenarios a simple visteo no hay diferencias notorias. Por esta razón se procedió a analizar dos medidas estadísticas como lo son la media y la varianza, con el fin de identificar un parámetro que discrimine los tres escenarios.

Figura 29: Campos magnéticos para los tres escenarios planteados.



Se tomó únicamente las muestras correspondientes al tramo donde hubo variaciones en los carros. Se calculó la media y la varianza de la señal. Un resumen de estos resultados se muestran en la tabla 9.

Tabla 9: Varianzas en las señales de campos magnéticos del tercer tramo para cada escenario.

Escenario	Varianza	Media
Sin carros	0,0099	0,3337
Carros moderado	0,014	0,2285
Full carros	0,019	0,3406

El cambio en la varianza es mas notoria en los campos magnéticos del eje X, por esta razón se decidió utilizar este sensor como determinante de cuando el filtro debía variar la constante de tiempo del modelo Gauss-Markov usado para el error en la longitud del paso y el bias de la orientación. Las constantes de tiempo se hicieron a prueba y error *offline*. El banco definitivo de las constantes de tiempo del modelo se presentan en la tabla 10.

Tabla 10: Banco de constantes de tiempo para el modelo Gauss-Markov del error en el bias de la orientación.

Escenario	Constante de tiempo
Sin carros	0,8
Carros moderado	0,6
Full carros	0,3

**Constantes de tiempo para el error de la longitud del paso.** Para el cálculo de la constante de tiempo para el error en la longitud del paso, un proceso similar se llevó a cabo. Aquí, se delimitó a que el factor determinante para una variación en la constante de tiempo era la longitud del paso. Esto es similar a referirse a la estatura del usuario, pues se ha comprobado que para una misma frecuencia de caminata, una persona con mayor estatura presenta una mayor longitud del paso que una de menor estatura.

Para esta parte, se realizaron pruebas con cinco usuarios de diferente estatura y se calcularon sus longitudes del paso. A todos se les solicitó que caminaran a una frecuencia normal de caminata. Un resumen de estos resultados se presentan en la tabla 11.

Tabla 11: Resumen de longitudes del paso a una frecuencia normal de caminata para usuarios de diferente estatura.

Usuario	Estatura	Media longitud del paso
Usuario 1	<i>Baja</i>	0.9122
Usuario 2	<i>Baja</i>	0.988
Usuario 3	<i>Media</i>	1.05
Usuario 4	<i>Media</i>	1.011
Usuario 5	<i>Alta</i>	1.1453

Las constantes de tiempo para cada escenario también fueron estimadas a prueba y error de forma *offline*. Una tabla que muestra los resultados obtenidos y el banco de constantes para el modelo del error en la longitud del paso se muestra a continuación:

Tabla 12: Banco de constantes de tiempo para el modelo Gauss-Markov del error en la longitud del paso.

Estatura	Constante de tiempo	Longitud del paso
Baja	1	Menor a 1
Media	0.7	Menor a 1,1 y mayor a 1
Alta	0.5	Mayor a 1,1

El factor determinante para la elección de la constante de tiempo fue la longitud del paso, pues la varianza no era muy diferente a distintas estaturas.

## Validación de los métodos de corrección de errores y estimación de parámetros del PDR

La primera parte fue probar los tres métodos de estimación de longitud del paso y los tres métodos de corrección de errores en la orientación. Estos métodos al depender únicamente de las medidas de la unidad de medida inercial, se pudieron probar en interiores. Con base en estos resultados, se conformó un sistema PDR con el mejor método de estimación de longitud del paso y el mejor de corrección de errores en la orientación. Finalmente este algoritmo fue probado en exteriores y comparado con el mismo algoritmo pero corregido con el filtro de Kalman. A continuación se muestran los resultados obtenidos.

Los métodos de estimación de longitud del paso fueron probados por separado al igual que los de

corrección de errores en la orientación. Además se presenta una comparación de los dos métodos de detección del paso implementados. Estas pruebas se hicieron en interiores.

### Desempeño de la detección del paso

La primera prueba consistió en una caminata recta de 30m. Le segunda fue caminar por una ruta de 96m en forma de cuadrado, es decir con cuatro giros de 90° cada uno. En la tabla 13 y 14 se observan los resultados obtenidos por el método de detección de la fase de apoyo y el método basado en la señal de aceleración horizontal respectivamente.

Tabla 13: Resultados de detección del paso a diferentes frecuencias de caminata detectando la fase de apoyo.

<b>Frecuencia lenta</b>				
Prueba	Pasos reales	Pasos contados	Error de pasos	Error de pasos %
1	15	15	0	0
2	103	99	4	3,883
<b>Frecuencia normal</b>				
1	15	15	0	0
2	81	81	0	0
<b>Frecuencia rápida</b>				
1	15	15	0	0
2	70	70	0	0

Se le solicitó al usuario repetir ambas pruebas dos veces a tres frecuencias de caminata diferentes: lenta, rápida y normal. Esto con el fin de determinar como es el comportamiento de ambos métodos a diferentes frecuencias de caminata.

Tabla 14: Resultados de detección del paso a diferentes frecuencias con la señal de aceleración horizontal.

<b>Frecuencia lenta</b>				
Prueba	Pasos reales	Pasos contados	Error de pasos	Error de pasos %
1	30	29	1	3,333
2	188	181	7	3,723
<b>Frecuencia normal</b>				
1	30	30	0	0
2	160	159	1	0,625
<b>Frecuencia rápida</b>				
1	30	30	0	0
2	140	140	0	0

De los resultados se observa que ambos métodos a frecuencias de caminata normal y rápida tienen un desempeño bueno, solo uno de ellos presentó error y este es menor al 1%. A frecuencia lenta, ambos métodos tienden a aumentar el error. Esto puede ser debido a que el golpe con el suelo no es tan fuerte por lo que el pico negativo no es tan pronunciado, además que los valores de aceleración durante la etapa de balanceo son más bajos, e inclusive a veces no superan el umbral de detección.

## Desempeño de los métodos de estimación de longitud del paso

Dos pruebas fueron realizadas para probar cada uno de los métodos de estimación de longitud del paso. La primera consistía en caminar por un pasillo recto de 10.68m. La segunda en hacer la ruta cuadrada de 96.86m. Además, a cada participante se le midió su longitud del paso real, por lo que al final de manera manual se le agregaba la distancia debida a los pasos que el algoritmo dejaba de detectar, esto con el fin de eliminar el error debido al algoritmo de detección del paso y centrar los resultados únicamente en la estimación de longitud.

Tabla 15: Resultados de la estimación de longitud del paso para cada método implementado.

Distancia real[m]	Distancia medida[m]	Error[m]	Pasos no detectados	Error[m]
<b>Método ZUPT</b>				
10,68	9,9611	0,7188	0	0,7188
96,86	94,5376	2,3224	0	2,3224
<b>Método dependiente de la frecuencia del paso</b>				
10,68	8,4359	2,2441	2	1,1896
96,86	93,5425	3,3175	3	1,6409
<b>Método dependiente del parámetro</b>				
10,68	10,9643	-0,2843	0	-0,2843
96,86	95,3354	1,5245	2	0,31

El método con mejor resultado es el dependiente del parámetro, sin embargo por el método de detección del paso podría llegar a tener mayor error que los demás. Otra limitante con este es el tener que calibrarse para cada usuario, y esta calibración se hace a prueba y error lo que lo hace poco útil para la mayoría de aplicaciones de localización de personas. El otro método que evidenció buenos resultados fue el ZUPT, que además de tener un error no muy alto no debe ser ajustado a cada usuario. Finalmente el método dependiente de la frecuencia del paso muestra también buenos resultados pero también requiere el ajuste de la curva para cada usuario. Sin embargo esto podría solucionarse con una interpolación.

## Desempeño de los métodos de corrección de error de la orientación

Para evaluar los métodos de corrección de orientación en la orientación se diseñaron dos pruebas. La primera consistía en caminar en línea recta y hacer un giro de 90°. En la segunda prueba, se le solicitaba al usuario caminar por una ruta cuadrada, es decir por una ruta con cuatro giros de 90°. Una gráfica que muestra los resultados de la primera prueba se observa en la figura 30. Además, los resultados cuantitativos de esta prueba se observan en la tabla 16.

Para la segunda prueba se compararon los cambios de orientación en cada giro con respecto al cambio verdadero. La tabla 17 resume estos resultados.

El método de corrección que mejor desempeño obtuvo en las dos pruebas fue el modelo del *yaw*, y una gráfica de los cambios de orientación en la prueba de cuatro giros se puede observar en la figura 31. El método HHR presenta un error bajo, sin embargo este método es útil para corregir el error en caminatas rectas, pues cuando se detecta giro no aplica ninguna corrección. Además el método de ZARU

Figura 30: Resultados de los tres métodos de corrección de error en la orientación de la primera prueba.

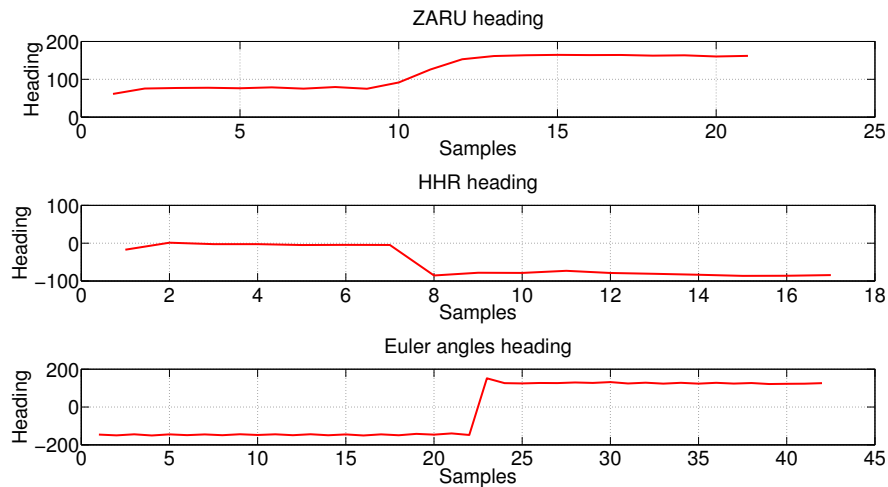


Tabla 16: Resultados de los métodos de corrección de error en la orientación de la primera prueba.

Prueba 1			
Método	Ángulo Real[°]	Ángulo estimado[°]	Error[°]
HHR	90	87,2502	2,7497
ZARU	90	76,9023	13,0976
Yaw	90	89,3274	0,6725

se ve afectado por el *drift*, por lo que retirar el valor promedio después de integrar no es suficiente y se evidencia en el alto error.

## Algoritmo de localización propuesto

Basándose en los resultados presentados en la sección anterior, se escogieron los métodos de longitud del paso y de corrección de error en orientación que menos error presentaron. Para la longitud del paso se escogió el método ZUPT que a pesar que no presenta el menor error, si es un error tolerable y tiene la ventaja que no debe ser ajustado para cada usuario. Para la corrección de la orientación, se escogió el modelo del *yaw* al ser el que menor error presentó. Un diagrama de bloques del algoritmo final de localización en interiores se muestra en la figura 32.

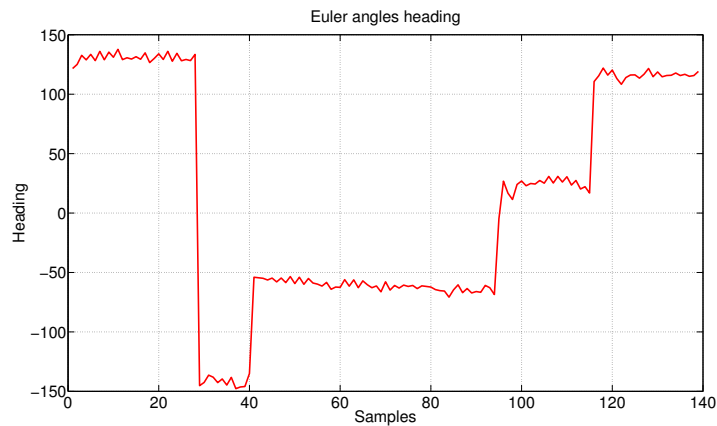
Este algoritmo depende únicamente de las medidas de la unidad de medida inercial, por lo que puede ser usado en interiores, sin embargo, es de interés evaluar su desempeño respecto a algoritmos de fusión de datos con el GPS. Por esta razón, se implementó un algoritmo de localización aplicando el filtro de Kalman extendido presentado en las secciones anteriores. Un diagrama de bloques que muestra el nuevo algoritmo de localización se observa en la figura 33.

Ambos algoritmos fueron probados en las instalaciones de la universidad. Los resultados de estas pruebas se observan en la figura 34.

De los resultados se observa que el filtro de Kalman extendido es sin duda el mejor método de corrección

Tabla 17: Resultados de la prueba de cuatro giros.

Prueba 2				
Method	Giro 1	Giro 2	Giro 3	Giro 4
HHR	87,06	86,55	84,26	88,78
ZARU	73,37	67,75	76,95	69,7
Yaw	86,82	85,36	89,36	88,82

Figura 31: Orientación de los cuatro giros con el modelo *yaw*.

del error en la orientación y además corrige la longitud del paso. Sin embargo, al ser un sistema que depende del GPS, su funcionamiento se garantiza para ambientes externos. Para interiores, se toma el promedio del error del bias en la orientación y el de la longitud del paso y se corrige hasta que el usuario vuelva a una zona donde el GPS tenga señal.

Consideraciones del algoritmo Para la estimación de longitud del paso aplicando corrección ZUPT, primero se debe hacer un cambio del sistema de referencia del cuerpo al sistema de referencia global. Ya teniendo las aceleración en este sistema, se procede con el cálculo de las velocidades en los tres ejes coordenados. Finalmente, para calcular la velocidad promedio durante la etapa de balanceo del pie, se utiliza la magnitud de la velocidad de los dos ejes del plano 2D de movimiento del usuario. Esto se decidió ya que observando el movimiento, se evidencia que el pie no se mueve siempre en línea recta, por lo que también hay un desplazamiento hacia los lados y esta distancia sería la verdadera longitud de zancada.

Además, durante la fase de apoyo se almacenan todas las orientaciones obtenidas del sensor y corregidas, y el promedio de esta ventana es la orientación del paso. Esto se hace debido a que al observar las señales se notaban algunas variaciones de esta pero que su valor promedio reflejaba el valor real de la orientación del usuario.

## Desempeño en interiores y exteriores

El algoritmo de localización final tiene un filtro de Kalman extendido como método de estimación de errores. Este se basa en las medidas del GPS y de una unidad de medida inercial. Sin embargo, hay lugares por los que es normal que una persona transite y por alguna condición el GPS pierda la señal (interiores,

Figura 32: Diagrama de bloques del algoritmo de localización para interiores.

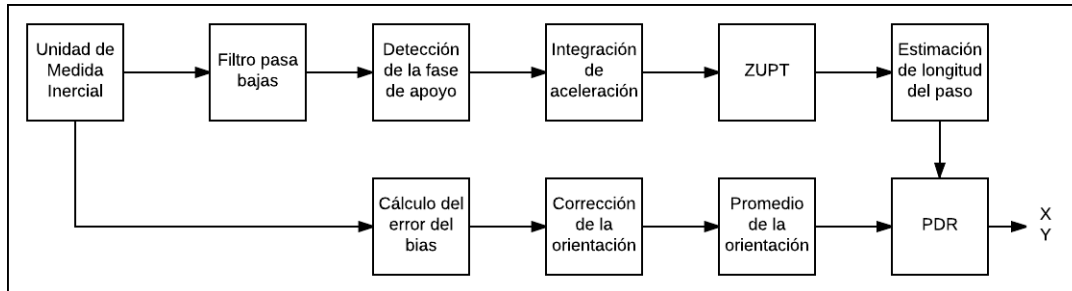
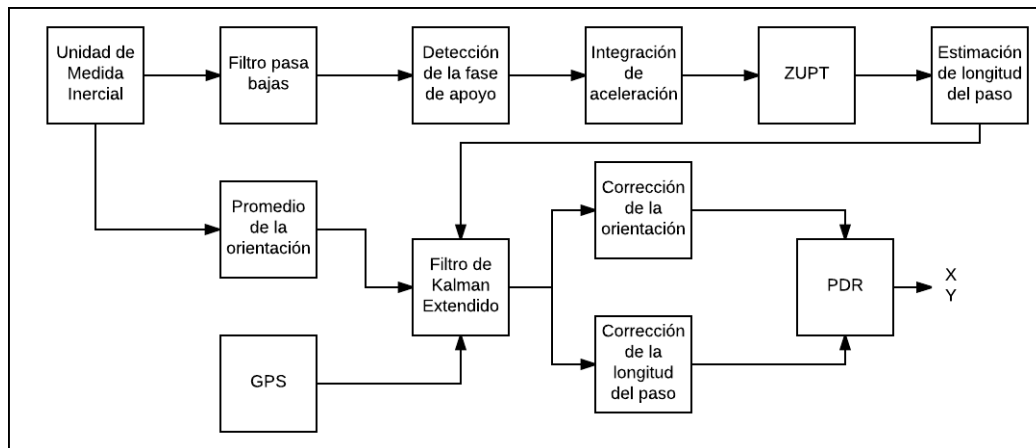


Figura 33: Diagrama de bloques del algoritmo de localización con filtro de Kalman.



interferencias, entre otras). Por esta razón, un método de corrección debe proponerse pues ya se evidenció que un sistema sin corrección de error diverge de manera rápida. A continuación se hace una explicación del método implementado en el algoritmo final y una comparación de resultados bajo condiciones normales y cuando el GPS pierde la señal.

### Localización en interiores

Los dos parámetros necesarios para aplicar corrección es el error en la longitud del paso y en el bias de la orientación. Esto lo predice el filtro de Kalman extendido en cada iteración. Sin embargo, cuando la información del GPS no está disponible, el filtro es incapaz de estimar una buena estimación de estos errores. Por esta razón, en este trabajo se propone almacenar el error de la longitud del paso y del bias de la orientación de las últimas 100 iteraciones. Cuando el usuario ingresa a un lugar donde no está disponible la información del GPS, se utiliza el promedio de los errores almacenados como el error de cada parámetro y se corrige. En la figura 35 se muestra una gráfica de los resultados obtenidos bajo condiciones normales y cuando el GPS no tiene señal.

Como la ruta de referencia donde se ejecutó la prueba es en exteriores y el GPS no perdía señal, durante un tramo del recorrido se desconectó la antena obligando a perder la información del receptor.

**Consideraciones.** Al comenzar la localización del usuario, este debe iniciar en un lugar donde el GPS tenga señal, y debe mantenerse 50 iteraciones mínimo antes que el GPS pierda la señal. Si no se tienen por lo menos 50 iteraciones, la estimación de errores puede ser menos precisa, sin embargo el algoritmo

Figura 34: Resultados de los dos algoritmos de localización bajo la misma ruta de referencia.



Figura 35: Comparación de resultados del algoritmo en interiores y exteriores.



seguirá estimando el error aunque seguramente con un error mayor.

Teniendo en cuenta la forma como el bias del sensor es calculado por tramos, si durante el tiempo donde no se dispone información del GPS el usuario hace algún giro notorio (encima del umbral), el bias del sensor en este caso no podrá ser puesto en cero, pues no se tiene información del GPS para re calcularlo, por lo que seguirá con el mismo que se tenía. Esto es una fuente de error pues la estimación a partir de este punto será menos exacta. Sin embargo, si el usuario durante este tiempo no realiza ningún giro, la estimación de posición tendrá menor error.

## Capítulo 4

# VALIDACIÓN DEL PROTOTIPO

El prototipo diseñado e implementado consta de una parte de *hardware*, referente a las conexiones físicas y comunicación con los sensores, y una parte de *software* que consiste en la implementación del algoritmo de localización en tiempo real. En esta sección se presenta un resumen del desempeño de ambos campos, dando a conocer el error del prototipo y algunas características como consumo de potencia, eficiencia, entre otros.

### Resultados en localización

Como se mencionó en el capítulo anterior, el prototipo final consta de una receptor GPS, una Unidad de Medida Inercial y una unidad de procesamiento. El sistema está formado por un método de detección del paso, uno de estimación de su longitud y el cálculo de orientación. Además se implementó un filtro de Kalman extendido que permite la estimación de errores en la longitud del paso y en el bias de la orientación. Finalmente se cuenta con un banco de constantes de tiempo para el modelo Gauss - Markov usado para modelar el error en la longitud del paso y el bias de la orientación del vector de estados del filtro para mejorar su desempeño bajo ciertos ambientes.

Para el análisis del desempeño respecto a localización, se consideraron dos aspectos: comparación de la distancia total recorrida con la distancia real (ruta de referencia) y la orientación del usuario respecto a la verdadera por tramos. En la tabla 18 se observan los resultados obtenidos en distancia recorrida para tres pruebas diferentes bajo las mismas condiciones.

Tabla 18: Resultados obtenidos en localización en tiempo real con constantes de tiempo dinámicas.

	Distancia real	Distancia calculada	Error[m]	Error %
Prueba 1	395,50	392,87	2,63	0,66
Prueba 2	395,50	391,43	4,07	1,03
Prueba 3	395,50	394,73	0,77	0,19

Estas mismas pruebas se realizaron con constantes de tiempo fijas del modelo Gauss-Markov de primer orden con el que fueron modelados el error en la longitud del paso y el error del bias en la orientación. Estos resultados se presentan en la tabla 19.

Tabla 19: Resultados obtenidos en localización en tiempo real con constantes de tiempo fijas.

	Distancia real	Distancia calculada	Error[m]	Error %
Prueba 1	395,50	390,94	4,56	1,15
Prueba 2	395,50	389,57	5,43	1,49
Prueba 3	395,50	391,49	3,51	1,11

Observando la ruta de referencia (Figura 28) , se evidencian cuatro tramos de línea recta, los cuales sirven para medir la orientación real del usuario y la orientación de la ruta de referencia. En la tabla 20 se observan estos resultados.

Tabla 20: Resultados obtenidos en localización en tiempo real de la orientación por tramos con constantes de tiempo dinámicas.

	<b>Orientación real</b>	<b>Orientación calculada</b>	<b>Error</b>
Tramo 1	-73,37	-29,30	44,07
Tramo 2	89,06	86,12	2,94
Tramo 3	4,45	8,12	3,67
Tramo 4	-56,98	-57,93	0,95

Al igual que con la distancia recorrida, se comparó los resultados de orientación con constantes de tiempo dinámicas con respecto a las mismas pruebas pero con constantes de tiempo dinámicas. Estos resultados se observan en la tabla 21.

Tabla 21: Resultados obtenidos en localización en tiempo real de la orientación por tramos con constantes de tiempo fijas.

	<b>Orientación real</b>	<b>Orientación calculada</b>	<b>Error</b>
Tramo 1	-73,37	-29,79	43,58
Tramo 2	89,06	80,73	8,33
Tramo 3	4,45	12,16	7,71
Tramo 4	-56,98	-66,15	9,17

En ambos casos, la orientación y la longitud del paso evidencian errores considerablemente bajos, que inclusive mejoran la localización obtenida sólo por el sistema GPS sin ningún tipo de corrección de errores. Sin embargo, en el primer tramo, el error en la orientación es bastante alto, esto es mientras el filtro se estabiliza, pues en los demás tramos los errores bajan considerablemente.

## Desempeño en tiempo real

El algoritmo de localización implementado se ejecuta en la unidad de procesamiento en tiempo real. Sin embargo, debido a diferentes factores como velocidad de procesamiento y métodos utilizados de programación, existe cierto retraso entre la ubicación real del usuario y la estimada por el sistema. A continuación se hace un análisis de estos retrasos mencionados y algunas consideraciones que se tuvieron en cuenta para la implementación *online* del algoritmo.

### Retrasos

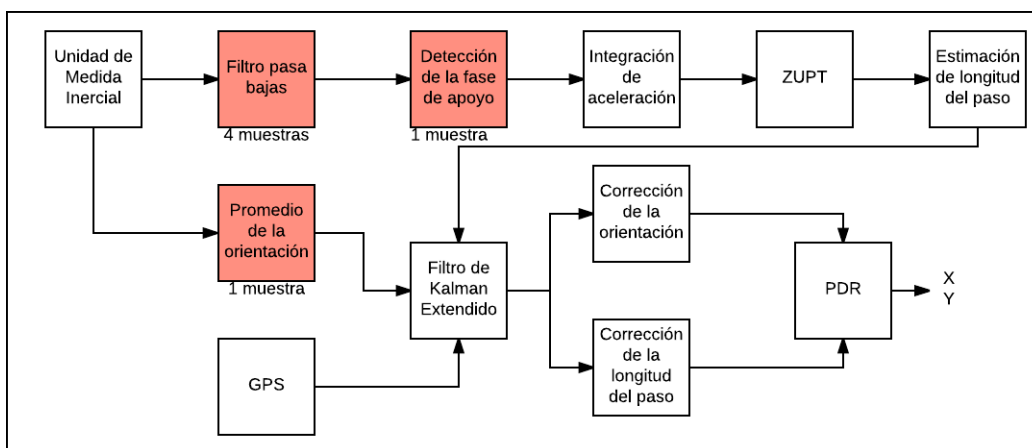
Los retrasos presentes se deben a dos factores: velocidad de procesamiento del micro controlador y a las producidas por el propio algoritmo.

**Retraso por velocidad de procesamiento.** Este retraso puede ser grande como puede ser pequeño, y depende de la complejidad del algoritmo en ejecución y de la capacidad de procesamiento del micro controlador. Para este caso en específico, como ya se mencionó, se trabajó con una unidad de procesamiento *Beaglebone* la cual tiene una frecuencia de procesamiento de 1GHz, lo que es una velocidad suficiente para este tipo de aplicación, además de ser superior a muchos micro controladores en el mercado actual.

Por otra parte, una medición del tiempo que dura en ejecutarse el algoritmo (función en C) arrojó que el algoritmo no demora más de  $100\mu$  en ejecutarse, tiempo más que suficiente para esperar la otra medida disponible de los sensores ( $10ms$ ). Estos tiempos son despreciables comparados con los tiempo de retardo causado por el algoritmo.

**Retrasos debido al algoritmo.** Estos retardos son introducidos en el sistema debido a la forma en que el algoritmo se ejecuta. En la figura 36 se muestra un diagrama de bloques del algoritmo de localización con sus respectivos retardos.

Figura 36: Diagrama de bloques del algoritmo con los respectivos retardos.



Los retardos se mencionan en muestras, y conociendo la frecuencia de muestreo ( $100Hz$ ) se puede conocer el retardo en tiempo. El primer retardo es debido al filtro de la señal. Este solo se presenta al iniciar al algoritmo y teniendo en cuenta que el filtro es de orden cuatro, se requieren cuatro muestras para obtener la primera muestra filtrada. El segundo bloque que introduce retardo es el de detección del

paso. Esto se debe a que el algoritmo detecta una fase de apoyo justo en la muestra siguiente que supera el umbral, es decir que se requiere de una muestra que pertenezca a la fase de balanceo para que el algoritmo identifique que la fase de apoyo se ha terminado y se proceda con el resto del procesamiento. Esto mismo ocurre con el cálculo de orientación, ya que esta es calculada como el promedio de las mediciones durante la fase de apoyo.

Con base en lo anterior, se presenta un retardo de aproximadamente cinco muestras, que un periodo de muestreo de  $10ms$  corresponde a un retardo total aproximado de  $50ms$ . Esto es un retardo considerablemente bajo para este tipo de aplicaciones, teniendo en cuenta que sistemas como el GPS actualizan la ubicación de un usuario cada segundo.

**Otro retardos.** Debe mencionarse que hay otros retardos que son introducidos directamente por los mismos sensores. Este error se le conoce como *scale factor* y se refiere al tiempo de desfase entre la entrada y la salida del sensor. Sin embargo este error puede despreciarse debido a que es mucho menor que los mencionados anteriormente.

### Consideraciones de implementación

A continuación se mencionan algunas consideraciones que se tuvieron en cuenta para la implementación en tiempo real.

- ❖ Al ser una aplicación en tiempo real, se debe tener información del receptor GPS y la IMU al mismo tiempo. Por esta razón, la implementación de *threads* fue necesario. Esto es una forma de paralelizar las tareas que debe hacer el controlador o el procesador. Para este caso se utilizó un *thread* para cada sensor.
- ❖ La sincronización entre sensores puede causar un error en la distancia total recorrida al final. Esto debido a que la ejecución del filtro de Kalman depende de cuando se tenga disponible una nueva medida del GPS, pero de manera paralela se está calculando la distancia recorrida por el usuario con las medidas de la IMU. Si la medida del GPS llega y el usuario no ha terminado de salir de la fase de apoyo, la distancia calculada por la IMU será de cero, lo que introduce un error de la longitud de una zancada en la posición final, que será corregido inmediatamente llegue la nueva medida del GPS.

## Desempeño del hardware

Para analizar el desempeño del *hardware*, se tuvieron en cuenta aspectos como eficiencia, potencia consumida, latencia, entre otros. A continuación se mencionan algunos análisis realizados para evaluar el prototipo.

### Potencia consumida

Tanto el receptor GPS como la IMU fueron configuradas en modo ahorro de energía, esto quiere decir que entran en modo reposo cuando detectan que el usuario no está en movimiento. Esto permite que la duración de la batería sea mayor. Sin embargo, para el caso en el que el usuario esté en movimiento todo

el tiempo, el consumo del prototipo será constante. Un resumen de estos consumos se observan en la tabla 22.

Tabla 22: Tabla con el consumo de potencia de cada componente del prototipo.

Dispositivo	Potencia consumida
Unidad de procesamiento	210 mA
Receptor GPS	100 mA
IMU	15 mA
Batería	250 mA

Observando las características de la batería, el prototipo bajo condiciones de operación continuas duraría un periodo de tiempo demasiado corto para este tipo de aplicaciones. Esto es sin duda una deficiencia pues es de interés realizar localización por periodos de horas e incluso de días.

## Capítulo 5

# CONCLUSIONES Y TRABAJO FUTURO

### Conclusiones

En este trabajo se diseñó y se implementó un prototipo para localización de personas, conformado por una unidad de procesamiento (*Beaglebone black*), una IMU (*Xsens MTi 1 series*) y un receptor GPS (*uBlox EVK M8*), implementando el algoritmo PDR, el cual consta de tres parámetros fundamentales: la estimación de la longitud del paso, la estimación de orientación y conocer la ubicación del instante anterior. Para determinar que métodos utilizar para calcular cada parámetro, se seleccionaron dos para detección del paso: aceleración vertical y detección de la fase de apoyo; tres para la estimación de longitud del paso: ZUPT, método basado en la frecuencia de caminata y método basado en un parámetro de proporcionalidad; y dos de corrección de orientación: ZARU y HHR. Además se utilizó un modelo para corregir el bias de la orientación. De estos, se seleccionó ZUPT y corrección del bias para el algoritmo final. Finalmente, este algoritmo PDR fue comparado con otro que tenía estimación de errores con un filtro de Kalman extendido. De los resultados obtenidos se concluye que la estimación de errores con fusión de datos aumenta hasta un 5 % en comparación con métodos para interiores.

Además, teniendo en cuenta que el error en la longitud del paso y el error en el bias de la orientación fueron modelados como procesos Gauss-Markov de primer orden, se estableció un banco con tres constantes de tiempo diferentes para cada variable. Para el error en la longitud del paso se establecieron 1, 0.7 y 0.5, y se tuvo como factor determinante la longitud del paso del usuario. Para la orientación se establecieron 0.8, 0.6 y 0.3, y se estableció como único factor determinando la varianza en la señal de campos magnéticos, es decir las perturbaciones generadas por los carros. Se realizó una comparación entre el algoritmo con constantes de tiempo fijas y dinámicas, y se logró reducir el error hasta un 0.5 % de lo que se tenía.

Con respecto a la implementación del *hardware* y la ejecución en tiempo real, se seleccionó el protocolo de comunicación UART para comunicar de manera serial la unidad de procesamiento y los sensores. Para polarización del prototipo, se diseñó un circuito que permitiera alimentar la unidad de procesamiento, y a través de esta energizar los sensores. Sin embargo, al utilizar una batería de 9V como se diseñó, el tiempo de duración es de máximo 40 minutos, lo que es muy poco para este tipo de aplicaciones, por lo que otro sistema de polarización debe ser implementado si se desea una ejecución independiente y de larga duración. Con respecto al desempeño en tiempo real, el análisis se basó en identificar los retardos generados por el sistema con respecto a la actualización actual del usuario. La principal fuente de retardo es la que

causa la forma de implementación del algoritmo, la cual alcanza a introducir un retardo de hasta  $50ms$ . A esto hay que sumarle los retardos considerablemente menores propios de los sensores y los causados por la capacidad de procesamiento del procesador. Sin embargo, este retraso es muy bajo para este tipo de aplicaciones, considerando que la frecuencia de de caminata de una persona puede ser de hasta 0.6 metros por segundo.

Finalmente, la evaluación del prototipo en entornos cerrados como abiertos fue realizada. Cuando el algoritmo no tiene disponible información del receptor GPS, este procede a utilizar el promedio de las últimas 60 estimaciones realizadas que se guardan iteración tras iteración. Sin embargo, el usuario podría permanecer hasta dos minutos sin información del GPS y siempre y cuando su dirección sea una línea recta, de hacer algún giro (umbral de 20 grados), el bias aumenta e incluso puede llegar a hacer divergir la estimación.

## Trabajo a futuro

Como trabajo a futuro se plantean dos caminos a continuar: Desde la parte de *software* se propone la combinación de otros sensores para aumentar la precisión en localización de interiores como: RFID, UWB, WiFi, entre otros, además de otros métodos de fusión de datos como filtros de partículas o *map matching*. Desde el punto de vista de *hardware* se plantea una mejora al sistema de polarización, que permita una mayor duración de funcionamiento del prototipo, teniendo en cuenta que para este tipo de aplicaciones los resultados obtenidos en este proyecto para este campo no son suficientes. Como alternativa se propone probar el dispositivo a diferentes frecuencias de muestreo, con el fin de reducir el consumo de potencia y probar hasta qué punto el algoritmo sigue trabajando con una precisión aceptable.

También se propone extender el análisis a otro tipo de usuarios, con limitaciones físicas. Esto ya que en este proyecto las pruebas realizadas siempre fueron con adultos jóvenes sanos. Para esto se deben tomar patrones de la marcha y hacer una clasificación basándose en las diferentes patologías que los usuarios presenten. Esto abre la puerta a involucrar más Unidades de Medida Inercial que permitan un análisis de patrones de diferentes partes del cuerpo, con el fin de lograr a proponer algoritmos distintos a los tradicionales.

Finalmente, se propone trabajar con técnicas de inteligencia artificial, ya que en la revisión bibliográfica se ha evidenciado que este puede tener mejores resultados en el campo de localización de personas que los métodos tradicionales.

# BIBLIOGRAFÍA

Abdulrahim, K., Hide, C., Moore, T., and Hill, C. (2011). Aiding Low Cost Inertial Navigation with Building Heading for Pedestrian Navigation. *Journal of Navigation*, 64(02):219–233.

Abdulrahim, K., Moore, T., Hide, C., and Hill, C. (2014). Understanding the Performance of Zero Velocity Updates in MEMS-based Pedestrian Navigation. *International Journal of Advancements in Technology*, 5(2):53–60.

Angermann, M., Robertson, P., Kemptner, T., and Khider, M. (2010). A high precision reference data set for pedestrian navigation using foot-mounted inertial sensors. *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, (September):15–17.

Bancroft, J. B. (2010). *Multiple Inertial Measurement Unit Integration for Pedestrian Navigation*. PhD thesis, SCHULICH.

Beaugard, S. and Haas, H. (2006). Pedestrian dead reckoning: A basis for personal positioning. *Positioning, Navigation and Communication*, pages 27–35.

Brown, R. G. and Hwang, P. Y. (2012). *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, Inc, United States, 4 edition.

Castañeda, N. and Lamy-Perbal, S. (2010). An improved shoe-mounted inertial navigation system. *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, (September):15–17.

Cavallo, F., Sabatini, A. M., and Genovese, V. (2005). A step toward GPS/INS personal navigation systems: Real-time assessment of gait by foot inertial sensing. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 109–113.

Chen, L. and Hu, H. (2012). IMU/GPS based pedestrian localization. *2012 4th Computer Science and Electronic Engineering Conference, CEEC 2012 - Conference Proceedings*, pages 23–28.

Cho, S. Y. and Park, C. G. (2005). MEMS Based Pedestrian Navigation System. *Journal of Navigation*, 59(01):135.

Derawi, M. (2010). Accelerometer-Based Gait Analysis , A survey Mohammad Omar Derawi Accelerometer Based Gait Analysis. *Norwegian Information Security Conference*, (Mv):33–45.

Dorobantu, R. and Zebhauser, B. (1999). Field Evaluation of a Low-Cost Strapdown IMU by means GPS.

- Faulkner, W. T., Alwood, R., and Faulkner, T. (2010). GPS-Denied Pedestrian Tracking in Indoor Environments Using an IMU and Magnetic Compass. *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation January 25 - 27, 2010 Catamaran Resort Hotel San Diego, CA*, pages 198–204.
- Feng, W., Zhao, H., Zhao, Q., and Li, J. (2013). Integration of GPS and low cost INS for pedestrian navigation aided by building layout. *Chinese Journal of Aeronautics*, 26(5):1283–1289.
- Foxlin, E. (2002). Motion tracking requirements and technologies. In *Handbook of virtual environment technology*, chapter 7, pages 163–210. Kay stanne edition.
- Foxlin, E. (2005). Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38–46.
- Frank, K., Robertson, P., and Angermann, M. (2006). Reliable Real-Time Recognition of Motion Related Human Activities Using MEMS Inertial Sensors. *ION GNSS 2010*.
- Godha, S. and Lachapelle, G. (2008). Foot mounted inertial system for pedestrian navigation. *Measurement Science and Technology*, 19(7):075202.
- Guo, H., Uradzinski, M., Yin, H., and Yu, M. (2015). Indoor positioning based on foot-mounted IMU. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 63(3):629–634.
- Harle, R. (2013). IEEE Xplore - A Survey of Indoor Inertial Positioning Systems for Pedestrians. 15(3):1281–1293.
- Herrera, E., Kaufmann, H., and Quirós, R. (2007a). Assessment of step determination in a GPS/compass/IMU system for personal positioning. *20th International Technical Meeting of the Satellite Division of The Institute of Navigation 2007 ION GNSS 2007*, 5(January):1508–1514.
- Herrera, E., Quirós, R., and Kaufmann, H. (2007b). Analysis of a Kalman Approach for a Pedestrian Positioning System in Indoor Environments. *Euro-Par 2007 Parallel Processing*, (C):931–940.
- Herrera, E. P. and Kaufmann, H. (2010). Adaptive Methods of Kalman Filtering for Personal Positioning Systems. *23rd International Technical Meeting of the Satellite Division of The Institute of Navigation*, (May 2016):584–589.
- Hide, C. (2003). *Integration of GPS and low cost INS measurements*. PhD thesis, Nottingham.
- Hide, C., Moore, T., and Smith, M. (2004). Adaptive Kalman filtering algorithms for integrating GPS and low cost INS. *PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No.04CH37556)*, pages 227–233.
- Hundza, S. R., Hook, W. R., Harris, C. R., Mahajan, S. V., Leslie, P. A., Spani, C. A., Spalteholz, L. G., Birch, B. J., Commandeur, D. T., and Livingston, N. J. (2014). Accurate and reliable gait cycle detection in parkinson’s disease. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(1):127–137.

- Jiménez, A. R., Seco, F., Prieto, C., and Guevara, J. (2009a). A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. *WISP 2009 - 6th IEEE International Symposium on Intelligent Signal Processing - Proceedings*, pages 37–42.
- Jiménez, A. R., Seco, F., Prieto, C., and Guevara, J. (2009b). 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.
- Jimenez, A. R., Seco, F., Prieto, J. C., and Guevara, J. (2010). Indoor Pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. *Proceedings of the 2010 7th Workshop on Positioning, Navigation and Communication, WPNC'10*, pages 135–143.
- Jiménez, A. R., Seco, F., Zampella, F., Prieto, J. C., and Guevara, J. (2011). PDR with a foot-mounted IMU and ramp detection. *Sensors*, 11(10):9393–9410.
- Jiménez Ruiz, A. R., Seco Granja, F., Prieto Honorato, J. C., and Guevara Rosas, J. I. (2012). Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements. *IEEE Transactions on Instrumentation and Measurement*, 61(1):178–189.
- Jirawimut, R., Ptasinski, P., Garaj, V., Cecelja, F., and Balachandran, W. (2003). A method for dead reckoning parameter correction in pedestrian navigation system. *IEEE Transactions on Instrumentation and Measurement*, 52(1):209–215.
- Kaplan, E. D. and Hegarty, C. J. (2006). *Understanding GPS. Principles and Applications*. Norwood, 2 edition.
- Kietlinski-zaleski, J., Yamazato, T., and Katayama, M. (2010). TDOA UWB Positioning with Three Receivers Using Known Indoor Features. *International Conference on Ultra-Wideband*, pages 4–7.
- Kim, J., Jang, H., and Hwang, D. (2004). A step, stride and heading determination for the pedestrian navigation system. *Global Positioning Systems*, 3(1):273–279.
- Kong, X. (2000). *Inertial navigation system algorithms for low cost IMU*. PhD thesis, University of Sydney.
- Kouroggi, M., Ishikawa, T., Kameda, Y., Ishikawa, J., Aoki, K., and Kurata, T. (2009). Pedestrian Dead Reckoning and its applications. *Ismar '09*, (C):3–4.
- Kouroggi, M., Sakata, N., Okuma, T., and Kurata, T. (2006). Indoor/outdoor pedestrian navigation with an embedded GPS/RFID/self-contained sensor system. *Advances in Artificial Reality and Tele-Existence*, 4282:1310–1321.
- Kubrak, D., Macabiau, C., and Monnerat, M. (2006). Real time pedestrian navigation system. *ION GNSS 2006, 19th International Technical Meeting of the Satellite Division of The Institute of Navigation*, pages 2990–2998.
- Lachapelle, G., Mezentsev, O., Collin, J., and MacGougan, G. (2003). Pedestrian and Vehicular Navigation Under Signal Masking Using Integrated HSGPS and Self Contained Sensor Technologies.

- Lategahn, J., Muller, M., and Rohrig, C. (2014). Extended Kalman filter for a low cost TDoA/IMU pedestrian localization system. *2014 11th Workshop on Positioning, Navigation and Communication, WPNC 2014*.
- Levi, R. and Judd, T. (1996). Dead reckoning navigational system using accelerometer to measure foot impacts. US Patent 5,583,776.
- Li, Y. (2014). Pedestrian Navigation System Using Shoe-mounted INS. (July).
- Li, Y., Wang, J. J., Xiao, S., and Luo, X. (2012). Dead reckoning navigation with Constant Velocity Update (CUPT). *2012 12th International Conference on Control, Automation, Robotics and Vision, ICARCV 2012*, 2012(December):160–165.
- Liu, H., Darabi, H., Banerjee, P., and Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37(6):1067–1080.
- Lowe, S. A. and ÓLaighin, G. (2014). Monitoring human health behaviour in one’s living environment\_ A technological review. 36(2):147–168.
- Mautz, R. (2012). Indoor Positioning Technologies. (February 2012):127.
- Miller, J. and Bevly, D. (2007). Position and orientation determination for a guided K-9. . . . *Meeting of the Satellite Division of . . .*, pages 1768–1776.
- Muro-de-la Herran, A., García-Zapirain, B., and Méndez-Zorrilla, A. (2014). Gait Analysis Methods: An Overview of Wearable and Non-Wearable Systems, Highlighting Clinical Applications. *Sensors*, 14(2):3362–3394.
- Nilsson, J. O., Skog, I., H??ndel, P., and Hari, K. V. S. (2012). Foot-mounted INS for everybody - An open-source embedded implementation. *Record - IEEE PLANS, Position Location and Navigation Symposium*, pages 140–145.
- Ojeda, L. and Borenstein, J. (2007). Personal Dead-reckoning System for GPS-denied Environments. (September):1–6.
- Pittet, S., Renaudin, V., and Merminod, B. (2008). The journal of navigation. *The Journal of Navigation*, 61:369–384.
- Pulido Herrera, E., Kaufmann, H., Secue, J., Quir??s, R., and Fabregat, G. (2013). Improving data fusion in personal positioning systems for outdoor environments. *Information Fusion*, 14(1):45–56.
- Rahim, K. A. (2012). *Heading Drift Mitigation for Low-Cost Inertial Pedestrian Navigation*. PhD thesis, Nottingham.
- Rajagopal, S. (2008). Personal dead reckoning system with shoe mounted inertial sensors. *Signal Processing*, (C):1–47.

- Romanovas, M., Goridko, V., Al-Jawad, A., Schwaab, M., Traechtler, M., Klingbeil, L., and Manoli, Y. (2012). A study on indoor pedestrian localization algorithms with foot-mounted sensors. *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings*, pages 13–15.
- S. Godha, G. Lachapelle, M. E. C. (2006). Integrated GPS/INS System for Pedestrian Navigation in a Signal Degraded Environment. *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, (September):2151–2164.
- Seco, F., Plagemann, C., and Burgard, W. (2010). Improving RFID-Based Indoor Positioning Accuracy Using Gaussian Processes. *International Conference on indoor positioning and indoor navigation*, (September):15–17.
- Senanayake, C. M. and Arossha Senanayake, S. M. N. (2010). Evaluation of gait parameters for gait phase detection during walking. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 127–132.
- Skog, I., Händel, P., Nilsson, J.-O., and Rantakokko, J. (2010a). Zero-velocity detection — an algorithm evaluation. *IEEE transactions on bio-medical engineering*, 57(11):2657–2666.
- Skog, I., Nilsson, J. O., and Händel, P. (2010b). Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, (September):15–17.
- Stirling, R., Fyfe, K., and Lachapelle, G. (2005). Evaluation of a New Method of Heading Estimation for Pedestrian Dead Reckoning Using Shoe Mounted Sensors. *The Journal of Navigation*, 58(01):31–45.
- Stirling, R. G. (2004). Development of a Pedestrian Navigation System Using Shoe Mounted Sensors by. page 132.
- Walther-Franks, B. and Malaka, R. (2008). *Evaluation of an Augmented Photograph-Based Pedestrian Navigation System*, pages 94–105. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Wang, J. J. and Kong, X. (2013). Zero velocity update with stepwise smoothing for inertial pedestrian navigation.
- Wang, W., Engelaar, R., Chen, X., and Chase, J. (2009). The State-of-Art of Underwater Vehicles - Theories and Applications, Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions. chapter 7, pages 129–152. InTech.
- Wang, Z. L., Zhao, H. Y., Qiu, S., and Gao, Q. (2015). Stance phase detection for ZUPT-aided foot-mounted pedestrian navigation system. *IEEE/ASME Transactions on Mechatronics*, 20(6):3170–3181.
- Yan, L., Yue-e, L., and Jian, H. (2010). Gait recognition based on MEMS accelerometer. *IEEE 10th International Conference on Signal Processing*, pages 1679–1681.
- Young, S. S. and Sangkyung, P. (2009). Pedestrian inertial navigation with gait phase detection assisted zero velocity updating. In *ICARA 2009 - Proceedings of the 4th International Conference on Autonomous Robots and Agents*, volume 1, pages 336–341.

Yuan, X., Liu, C., Zhang, S., Yu, S., and Liu, S. (2014). Indoor pedestrian navigation using miniaturized low-cost MEMS inertial measurement units. *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, pages 487–492.

# REFERENCIAS

- [1] Abdulrahim, K., Hide, C., Moore, T., and Hill, C. (2011). Aiding Low Cost Inertial Navigation with Building Heading for Pedestrian Navigation. *Journal of Navigation*, 64(02):219–233.
- [2] Abdulrahim, K., Moore, T., Hide, C., and Hill, C. (2014). Understanding the Performance of Zero Velocity Updates in MEMS-based Pedestrian Navigation. *International Journal of Advancements in Technology*, 5(2):53–60.
- [3] Angermann, M., Robertson, P., Kemptner, T., and Khider, M. (2010). A high precision reference data set for pedestrian navigation using foot-mounted inertial sensors. *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, (September):15–17.
- [4] Bancroft, J. B. (2010). *Multiple Inertial Measurement Unit Integration for Pedestrian Navigation*. PhD thesis, SCHULICH.
- [5] Beauregard, S. and Haas, H. (2006). Pedestrian dead reckoning: A basis for personal positioning. *Positioning, Navigation and Communication*, pages 27–35.
- [6] Brown, R. G. and Hwang, P. Y. (2012). *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, Inc, United States, 4 edition.
- [7] Castañeda, N. and Lamy-Perbal, S. (2010). An improved shoe-mounted inertial navigation system. *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, (September):15–17.
- [8] Cavallo, F., Sabatini, A. M., and Genovese, V. (2005). A step toward GPS/INS personal navigation systems: Real-time assessment of gait by foot inertial sensing. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 109–113.
- [9] Chen, L. and Hu, H. (2012). IMU/GPS based pedestrian localization. *2012 4th Computer Science and Electronic Engineering Conference, CEEC 2012 - Conference Proceedings*, pages 23–28.
- [10] Cho, S. Y. and Park, C. G. (2005). MEMS Based Pedestrian Navigation System. *Journal of Navigation*, 59(01):135.
- [11] Derawi, M. (2010). Accelerometer-Based Gait Analysis , A survey Mohammad Omar Derawi Accelerometer Based Gait Analysis. *Norwegian Information Security Conference*, (Mv):33–45.
- [12] Dorobantu, R. and Zebhauser, B. (1999). Field Evaluation of a Low-Cost Strapdown IMU by means GPS.

- [13] Faulkner, W. T., Alwood, R., and Faulkner, T. (2010). GPS-Denied Pedestrian Tracking in Indoor Environments Using an IMU and Magnetic Compass. *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation January 25 - 27, 2010 Catamaran Resort Hotel San Diego, CA*, pages 198–204.
- [14] Feng, W., Zhao, H., Zhao, Q., and Li, J. (2013). Integration of GPS and low cost INS for pedestrian navigation aided by building layout. *Chinese Journal of Aeronautics*, 26(5):1283–1289.
- [15] Foxlin, E. (2002). Motion tracking requirements and technologies. In *Handbook of virtual environment technology*, chapter 7, pages 163–210. Kay stanne edition.
- [16] Foxlin, E. (2005). Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38–46.
- [17] Frank, K., Robertson, P., and Angermann, M. (2006). Reliable Real-Time Recognition of Motion Related Human Activities Using MEMS Inertial Sensors. *ION GNSS 2010*.
- [18] Godha, S. and Lachapelle, G. (2008). Foot mounted inertial system for pedestrian navigation. *Measurement Science and Technology*, 19(7):075202.
- [19] Guo, H., Uradzinski, M., Yin, H., and Yu, M. (2015). Indoor positioning based on foot-mounted IMU. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 63(3):629–634.
- [20] Harle, R. (2013). IEEE Xplore - A Survey of Indoor Inertial Positioning Systems for Pedestrians. 15(3):1281–1293.
- [21] Herrera, E., Kaufmann, H., and Quirós, R. (2007a). Assessment of step determination in a GPS/compass/IMU system for personal positioning. *20th International Technical Meeting of the Satellite Division of The Institute of Navigation 2007 ION GNSS 2007*, 5(January):1508–1514.
- [22] Herrera, E., Quirós, R., and Kaufmann, H. (2007b). Analysis of a Kalman Approach for a Pedestrian Positioning System in Indoor Environments. *Euro-Par 2007 Parallel Processing*, (C):931–940.
- [23] Herrera, E. P. and Kaufmann, H. (2010). Adaptive Methods of Kalman Filtering for Personal Positioning Systems. *23rd International Technical Meeting of the Satellite Division of The Institute of Navigation*, (May 2016):584–589.
- [24] Hide, C. (2003). *Integration of GPS and low cost INS measurements*. PhD thesis, Nottingham.
- [25] Hide, C., Moore, T., and Smith, M. (2004). Adaptive Kalman filtering algorithms for integrating GPS and low cost INS. *PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No.04CH37556)*, pages 227–233.
- [26] Hundza, S. R., Hook, W. R., Harris, C. R., Mahajan, S. V., Leslie, P. A., Spani, C. A., Spalteholz, L. G., Birch, B. J., Commandeur, D. T., and Livingston, N. J. (2014). Accurate and reliable gait cycle detection in parkinson’s disease. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(1):127–137.

- [27] Jiménez, A. R., Seco, F., Prieto, C., and Guevara, J. (2009a). A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. *WISP 2009 - 6th IEEE International Symposium on Intelligent Signal Processing - Proceedings*, pages 37–42.
- [28] Jiménez, A. R., Seco, F., Prieto, C., and Guevara, J. (2009b). 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.
- [29] Jimenez, A. R., Seco, F., Prieto, J. C., and Guevara, J. (2010). Indoor Pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. *Proceedings of the 2010 7th Workshop on Positioning, Navigation and Communication, WPNC'10*, pages 135–143.
- [30] Jiménez, A. R., Seco, F., Zampella, F., Prieto, J. C., and Guevara, J. (2011). PDR with a foot-mounted IMU and ramp detection. *Sensors*, 11(10):9393–9410.
- [31] Jiménez Ruiz, A. R., Seco Granja, F., Prieto Honorato, J. C., and Guevara Rosas, J. I. (2012). Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements. *IEEE Transactions on Instrumentation and Measurement*, 61(1):178–189.
- [32] Jirawimut, R., Ptasinski, P., Garaj, V., Cecelja, F., and Balachandran, W. (2003). A method for dead reckoning parameter correction in pedestrian navigation system. *IEEE Transactions on Instrumentation and Measurement*, 52(1):209–215.
- [33] Kaplan, E. D. and Hegarty, C. J. (2006). *Understanding GPS. Principles and Applications*. Norwood, 2 edition.
- [34] Kietlinski-zaleski, J., Yamazato, T., and Katayama, M. (2010). TDOA UWB Positioning with Three Receivers Using Known Indoor Features. *International Conference on Ultra-Wideband*, pages 4–7.
- [35] Kim, J., Jang, H., and Hwang, D. (2004). A step, stride and heading determination for the pedestrian navigation system. *Global Positioning Systems*, 3(1):273–279.
- [36] Kong, X. (2000). *Inertial navigation system algorithms for low cost IMU*. PhD thesis, University of Sydney.
- [37] Kouroggi, M., Ishikawa, T., Kameda, Y., Ishikawa, J., Aoki, K., and Kurata, T. (2009). Pedestrian Dead Reckoning and its applications. *Ismar '09, (C)*:3–4.
- [38] Kouroggi, M., Sakata, N., Okuma, T., and Kurata, T. (2006). Indoor/outdoor pedestrian navigation with an embedded GPS/RFID/self-contained sensor system. *Advances in Artificial Reality and Tele-Existence*, 4282:1310–1321.
- [39] Kubrak, D., Macabiau, C., and Monnerat, M. (2006). Real time pedestrian navigation system. *ION GNSS 2006, 19th International Technical Meeting of the Satellite Division of The Institute of Navigation*, pages 2990–2998.
- [40] Lachapelle, G., Mezentsev, O., Collin, J., and MacGougan, G. (2003). Pedestrian and Vehicular Navigation Under Signal Masking Using Integrated HSGPS and Self Contained Sensor Technologies.

- [41] Lategahn, J., Muller, M., and Rohrig, C. (2014). Extended Kalman filter for a low cost TDoA/IMU pedestrian localization system. *2014 11th Workshop on Positioning, Navigation and Communication, WPNC 2014*.
- [42] Levi, R. and Judd, T. (1996). Dead reckoning navigational system using accelerometer to measure foot impacts. US Patent 5,583,776.
- [43] Li, Y. (2014). Pedestrian Navigation System Using Shoe-mounted INS. (July).
- [44] Li, Y., Wang, J. J., Xiao, S., and Luo, X. (2012). Dead reckoning navigation with Constant Velocity Update (CUPT). *2012 12th International Conference on Control, Automation, Robotics and Vision, ICARCV 2012*, 2012(December):160–165.
- [45] Liu, H., Darabi, H., Banerjee, P., and Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37(6):1067–1080.
- [46] Lowe, S. A. and ÓLaighin, G. (2014). Monitoring human health behaviour in one’s living environment\_ A technological review. 36(2):147–168.
- [47] Mautz, R. (2012). Indoor Positioning Technologies. (February 2012):127.
- [48] Miller, J. and Bevly, D. (2007). Position and orientation determination for a guided K-9. ... *Meeting of the Satellite Division of ...*, pages 1768–1776.
- [49] Muro-de-la Herran, A., García-Zapirain, B., and Méndez-Zorrilla, A. (2014). Gait Analysis Methods: An Overview of Wearable and Non-Wearable Systems, Highlighting Clinical Applications. *Sensors*, 14(2):3362–3394.
- [50] Nilsson, J. O., Skog, I., H??ndel, P., and Hari, K. V. S. (2012). Foot-mounted INS for everybody - An open-source embedded implementation. *Record - IEEE PLANS, Position Location and Navigation Symposium*, pages 140–145.
- [51] Ojeda, L. and Borenstein, J. (2007). Personal Dead-reckoning System for GPS-denied Environments. (September):1–6.
- [52] Pittet, S., Renaudin, V., and Merminod, B. (2008). The journal of navigation. *The Journal of Navigation*, 61:369–384.
- [53] Pulido Herrera, E., Kaufmann, H., Secue, J., Quir??s, R., and Fabregat, G. (2013). Improving data fusion in personal positioning systems for outdoor environments. *Information Fusion*, 14(1):45–56.
- [54] Rahim, K. A. (2012). *Heading Drift Mitigation for Low-Cost Inertial Pedestrian Navigation*. PhD thesis, Nottingham.
- [55] Rajagopal, S. (2008). Personal dead reckoning system with shoe mounted inertial sensors. *Signal Processing*, (C):1–47.

- [56] Romanovas, M., Goridko, V., Al-Jawad, A., Schwaab, M., Traechtler, M., Klingbeil, L., and Manoli, Y. (2012). A study on indoor pedestrian localization algorithms with foot-mounted sensors. *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings*, pages 13–15.
- [57] S. Godha, G. Lachapelle, M. E. C. (2006). Integrated GPS/INS System for Pedestrian Navigation in a Signal Degraded Environment. *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, (September):2151–2164.
- [58] Seco, F., Plagemann, C., and Burgard, W. (2010). Improving RFID-Based Indoor Positioning Accuracy Using Gaussian Processes. *International Conference on indoor positioning and indoor navigation*, (September):15–17.
- [59] Senanayake, C. M. and Aroscha Senanayake, S. M. N. (2010). Evaluation of gait parameters for gait phase detection during walking. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 127–132.
- [60] Skog, I., Händel, P., Nilsson, J.-O., and Rantakokko, J. (2010a). Zero-velocity detection — an algorithm evaluation. *IEEE transactions on bio-medical engineering*, 57(11):2657–2666.
- [61] Skog, I., Nilsson, J. O., and Händel, P. (2010b). Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, (September):15–17.
- [62] Stirling, R., Fyfe, K., and Lachapelle, G. (2005). Evaluation of a New Method of Heading Estimation for Pedestrian Dead Reckoning Using Shoe Mounted Sensors. *The Journal of Navigation*, 58(01):31–45.
- [63] Stirling, R. G. (2004). Development of a Pedestrian Navigation System Using Shoe Mounted Sensors by. page 132.
- [64] Walther-Franks, B. and Malaka, R. (2008). *Evaluation of an Augmented Photograph-Based Pedestrian Navigation System*, pages 94–105. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [65] Wang, J. J. and Kong, X. (2013). Zero velocity update with stepwise smoothing for inertial pedestrian navigation.
- [66] Wang, W., Engelaar, R., Chen, X., and Chase, J. (2009). The State-of-Art of Underwater Vehicles - Theories and Applications, Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions. chapter 7, pages 129–152. In Tech.
- [67] Wang, Z. L., Zhao, H. Y., Qiu, S., and Gao, Q. (2015). Stance phase detection for ZUPT-aided foot-mounted pedestrian navigation system. *IEEE/ASME Transactions on Mechatronics*, 20(6):3170–3181.
- [68] Yan, L., Yue-e, L., and Jian, H. (2010). Gait recognition based on MEMS accelerometer. *IEEE 10th International Conference on Signal Processing*, pages 1679–1681.
- [69] Young, S. S. and Sangkyung, P. (2009). Pedestrian inertial navigation with gait phase detection assisted zero velocity updating. In *ICARA 2009 - Proceedings of the 4th International Conference on Autonomous Robots and Agents*, volume 1, pages 336–341.

- [70] Yuan, X., Liu, C., Zhang, S., Yu, S., and Liu, S. (2014). Indoor pedestrian navigation using miniaturized low-cost MEMS inertial measurement units. *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, pages 487–492.

# Appendices

## APÉNDICE A

# Protocolos y direcciones

### Protocolo XS

Tabla 23: Identificadores del protocolo XS.

Nombre de grupo	Tipo de nombre	XDA tipo de nombre	Valor hexadecimal
<b><i>Timestamp</i></b>		<b><i>XDI_TimestampGroup</i></b>	
	<i>Packet Counter</i>	<i>XDI_PacketCounter</i>	1020
	<i>Sample Time Fine</i>	<i>XDI_SampleTimeFine</i>	1060
<b><i>Orientation Data</i></b>		<b><i>XDI_OrientationGroup</i></b>	
	<i>Quaternion</i>	<i>XDI_Quaternion</i>	201y
	<i>Rotation Matrix</i>	<i>XDI_RotationMatrix</i>	202y
	<i>Euler angles</i>	<i>XDI_EulerAngles</i>	203y
<b><i>Acceleration</i></b>		<b><i>XDI_AccelerationGroup</i></b>	
	<i>Delta V (dv)</i>	<i>XDI_DeltaV</i>	401y
	<i>Acceleration</i>	<i>XDI_Acceleration</i>	402y
	<i>Free Acceleration</i>	<i>XDI_FreeAcceleration</i>	404y
	<i>AccelerationHR</i>	<i>XDI_AccelerationHR</i>	404y
<b><i>Angular Velocity</i></b>		<b><i>XDI_AngularVelocityGroup</i></b>	
	<i>Rate of Turn</i>	<i>XDI_RateOfTurn</i>	802y
	<i>Delta Q (dq)</i>	<i>XDI_DeltaQ</i>	803y
	<i>RateOfTurnHR</i>	<i>XDI_RateOfTurnHR</i>	804y
<b><i>Magnetic</i></b>		<b><i>XDI_MagneticGroup</i></b>	
	<i>Magnetic Field</i>	<i>XDI_Magneticfield</i>	C02y
<b><i>Status</i></b>		<b><i>XDI_StatusGroup</i></b>	
	<i>Status Word</i>	<i>XDI_StatusWord</i>	E020

## Protocolo UBX

Tabla 24: Identificadores del protocolo UBX.

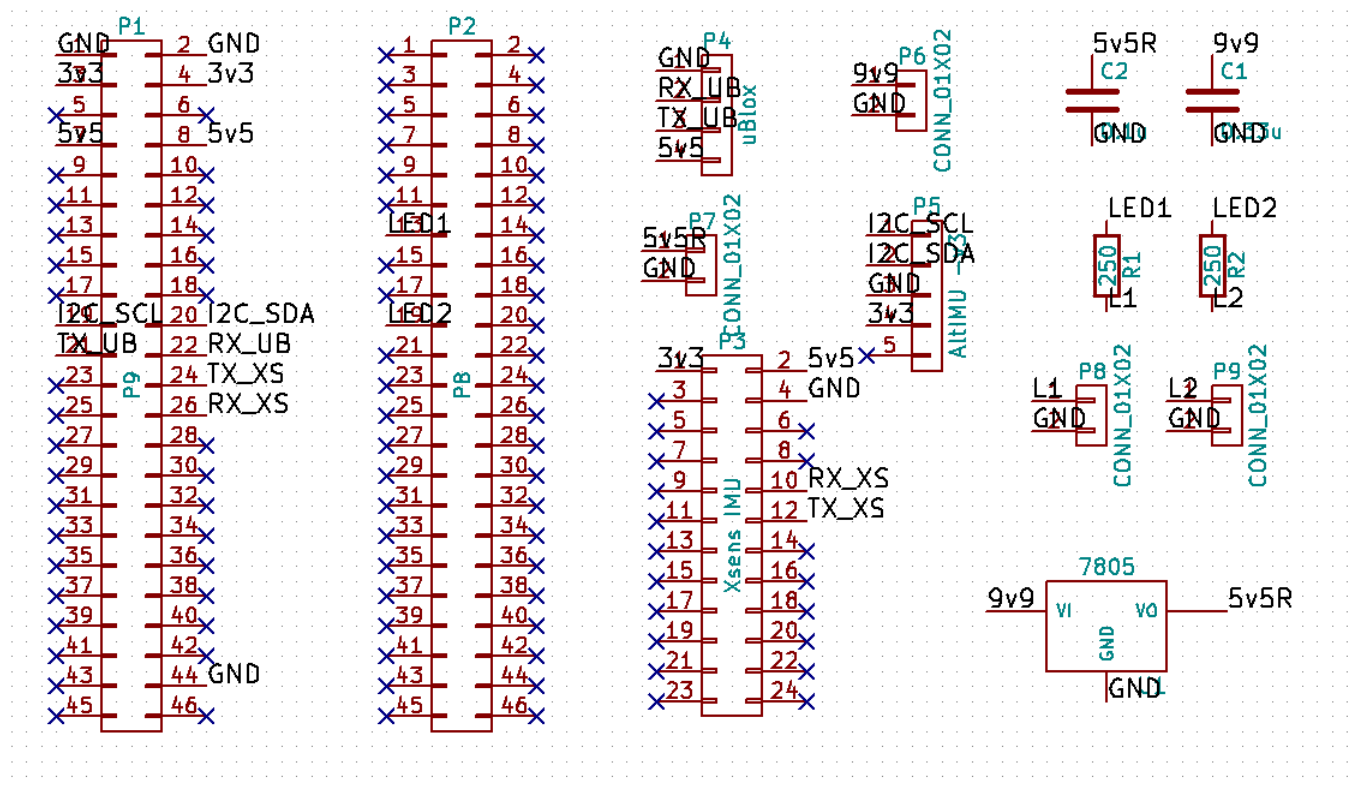
<i>Mnemonic</i>	Cls/ID	Descripción
<b>Mensajes Estandar NMEA</b>		<b>Mensajes Estandar</b>
DTM	0xF0 0x0A	Dato de referencia
GBS	0xF0 0x09	Detección de fallos satelitales GNSS
GGA	0xF0 0x00	Datos del arreglo del sistema de posicionamiento global
GLL	0xF0 0x01	Latitud y longitud, con tiempo de posición y estatus
GLQ	0xF0 0x43	Solicitar un mensaje estandar (si el ID actual es GL)
GNQ	0xF0 0x42	Solicitar un mensaje estandar (si el ID actual es GN)
GNS	0xF0 0x0D	Datos del arreglo GNSS
GPQ	0xF0 0x40	Solicitar un mensaje estandar (si el ID actual es GP)
GRS	0xF0 0x06	Rango de residuos GNSS
GSA	0xF0 0x02	Satélites activos y GNSS DOP
GST	0xF0 0x07	Errores estadísticos de pseudo rango GNSS
GSV	0xF0 0x03	Satélites en vista GNSS
RMC	0xF0 0x04	Datos mínimos recomendados
TXT	0xF0 0x41	Transmisión de texto
VTG	0xF0 0x05	Curso y velocidad en tierra
ZDA	0xF0 0x08	Fecha y hora
<b>Mensajes PUBX NMEA</b>		<b>Mensajes de propiedad</b>
CONFIG	0xF1 0x41	Establecer protocolos y velocidad de transferencia
POSITION	0xF1 0x00	Solicitar un mensaje PUBX,00
POSITION	0xF1 0x00	Datos de posición latitud y longitud
RATE	0xF1 0x40	Establecer frecuencia de salida de los mensajes NMEA
SVSTATUS	0xF1 0x03	Solicitar un mensaje PUBX,03
SVSTATUS	0xF1 0x03	Estado de satélites
TIME	0xF1 0x04	Solicitar un mensaje PUBX,04
TIME	0xF1 0x04	Información de fecha y reloj

# APÉNDICE B

## Esquemático del circuito impreso

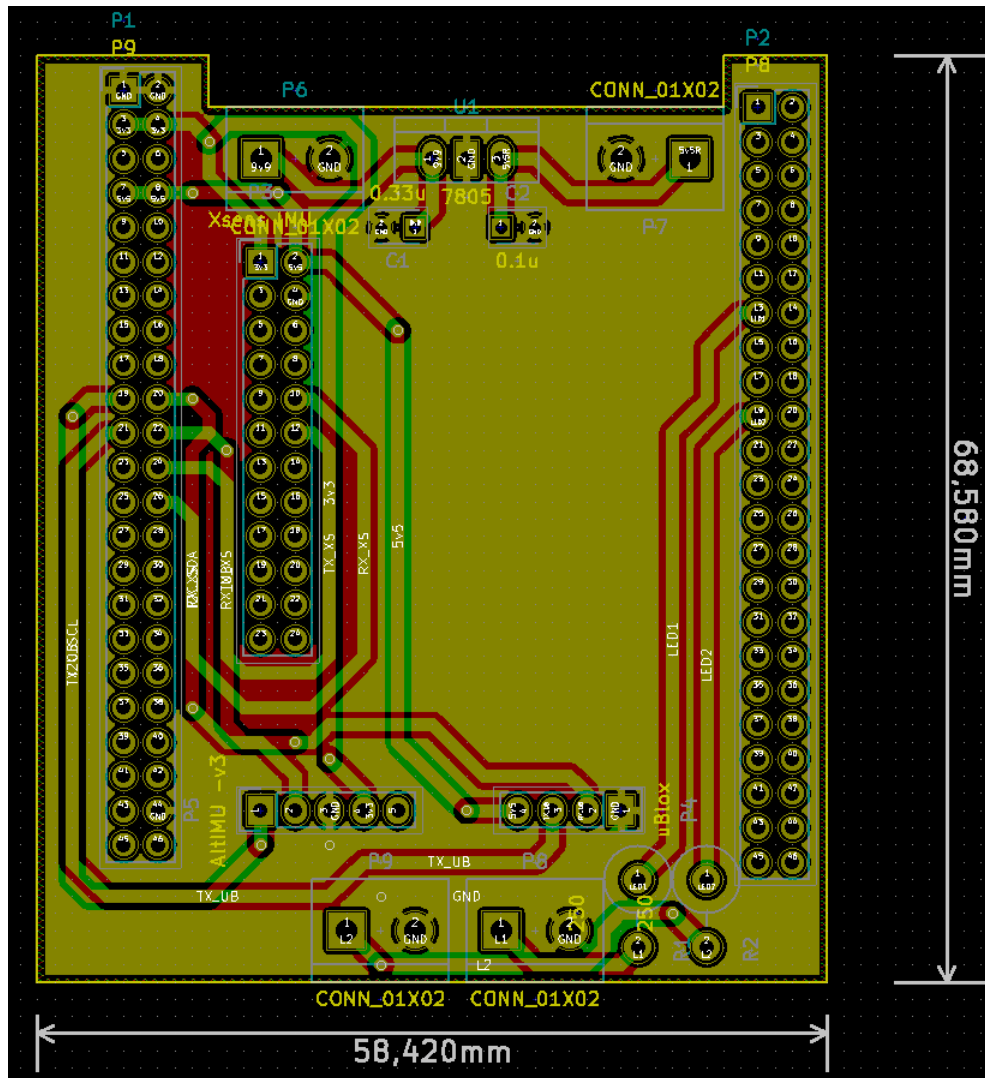
### Esquemático

Figura 37: Esquemático del circuito impreso implementado.



# Circuito impreso

Figura 38: Circuito impreso del esquemático diseñado.



## APÉNDICE C

# Resultados, productos e impactos

Los resultados, productos e impactos alcanzados son presentados a continuación.

### Resultados

- ❖ Diseño de un prototipo para estimación de posición de personas.
- ❖ Análisis comparativo de métodos de corrección de errores presentados por algunos autores en la literatura.
- ❖ Banco de constantes de tiempos para el error en la longitud del paso y el error en el bias de la orientación del modelo Gauss-Markov utilizado para modelar el error en la longitud del paso y el bias de la orientación del vector de estados del filtro de Kalman Extendido.
- ❖ Algoritmo de localización en tiempo real con filtro de Kalman como herramienta de estimación de errores.

### Productos

- ❖ Prototipo conformado por un receptor GPS, una unidad de procesamiento y una Unidad de Medida Inercial para localización de personas en tiempo real.
- ❖ Artículo sobre comparación de métodos de corrección de errores: "*Comparison of Step Length and Heading Estimation Methods for Indoor Environments*".

### Impactos

- ❖ El artículo: "*Comparison of Step Length and Heading Estimation Methods for Indoor Environments*" fue presentado en el congreso INTERCON2017, llevado a cabo del 15 al 18 de Agosto en la ciudad de Cusco, Perú.