

TESIS TESIS TESIS TESIS TESIS



**UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES**

CENTRO DE CIENCIAS BÁSICAS

TESIS

**RECONOCIMIENTO Y MAPEO DE UN ENTORNO A TRAVÉS DE
ECO-LOCALIZACIÓN ORIENTADO A LA NAVEGACIÓN DE ROBOTS
AUTÓNOMOS**

PRESENTA

Rubén Hernández Torres

**PARA OBTENER EL GRADO DE DOCTOR EN CIENCIAS APLICADAS Y
TECNOLOGÍA**

TUTORES

**Dr. José Antonio Guerrero Díaz de León
Dr. Jorge Eduardo Macías Díaz**

ASESOR

Dr. Angel Eduardo Muñoz Zavala

TESIS TESIS TESIS TESIS TESIS

Aguascalientes, Ags., 10 de Marzo de 2025

CARTA DE VOTO APROBATORIO
COMITÉ TUTORAL

MTRO. EN C. JORGE MARTÍN ALFÉREZ CHÁVEZ
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

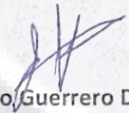
PRESENTE

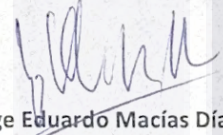
Por medio del presente como **Miembros del Comité Tutorial** designado del estudiante **RUBÉN HERNÁNDEZ TORRES** con ID **116090** quien realizó la tesis titulada: **RECONOCIMIENTO Y MAPEO DE UN ENTORNO A TRAVÉS DE ECO-LOCALIZACIÓN ORIENTADO A LA NAVEGACIÓN DE ROBOTS AUTÓNOMOS**, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia damos nuestro consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que nos permitimos emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla así como continuar con el procedimiento administrativo para la obtención del grado.

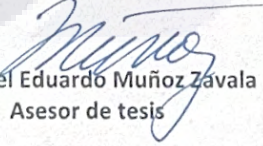
Ponemos lo anterior a su digna consideración y sin otro particular por el momento, le enviamos un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"

Aguascalientes, Ags., a 14 de Marzo de 2025.


Dr. José Antonio Guerrero Díaz de León
Tutor de tesis


Dr. Jorge Eduardo Macías Díaz
Co-Tutor de tesis


Dr. Ángel Eduardo Muñoz Zavala
Asesor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría Técnica del Programa de Posgrado

Elaborado por: Depto. Apoyo al Posgrado.
Revisado por: Depto. Control Escolar/Depto. Gestión de Calidad.
Aprobado por: Depto. Control Escolar/ Depto. Apoyo al Posgrado.

Código: DO-SEE-FO-16
Actualización: 00
Emisión: 17/05/19

DICTAMEN DE LIBERACION ACADEMICA PARA INICIAR LOS TRAMITES DEL EXAMEN DE GRADO

Fecha de dictaminación dd/mm/aaaa: 14/03/2025

NOMBRE: Rubén Hernández Torres ID 116090

PROGRAMA: Doctorado en Ciencias Aplicadas y Tecnología LGAC (del posgrado): Inteligencia Artificial y Modelación Estadística

TIPO DE TRABAJO: (X) Tesis () Trabajo Práctico

TITULO: Reconocimiento y mapeo de un entorno a través de eco-localización orientado a la navegación de robots autónomos

IMPACTO SOCIAL (señalar el impacto logrado): Los logros obtenidos permitieron entender de mejor manera como el sonido se comporta y su aplicación dentro de la robótica. Esto tiene un amplio alcance de aplicación, desde el desarrollo tecnológico en robótica hasta otros campos como la psicología y la educación.

INDICAR	SI	NO	N.A.	(NO APLICA)	SEGÚN	CORRESPONDA:
Elementos para la revisión académica del trabajo de tesis o trabajo práctico:						
SI						El trabajo es congruente con las LGAC del programa de posgrado
SI						La problemática fue abordada desde un enfoque multidisciplinario
SI						Existe coherencia, continuidad y orden lógico del tema central con cada apartado
SI						Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda
SI						Los resultados presentados en el trabajo son de gran relevancia científica, tecnológica o profesional según el área
SI						El trabajo demuestra más de una aportación original al conocimiento de su área
SI						Las aportaciones responden a los problemas prioritarios del país
SI						Generó transferencia del conocimiento o tecnológica
SI						Cumple con la ética para la investigación (reporte de la herramienta antiplagio)
El egresado cumple con lo siguiente:						
SI						Cumple con lo señalado por el Reglamento General de Docencia
SI						Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias, estancia, predoctoral, etc)
SI						Cuenta con los votos aprobatorios del comité tutorial, en caso de los posgrados profesionales si tiene solo tutor podrá liberar solo el tutor
N.A						Cuenta con la carta de satisfacción del Usuario
SI						Coincide con el título y objetivo registrado
SI						Tiene congruencia con cuerpos académicos
SI						Tiene el CVU del Conacyt actualizado
SI						Tiene el artículo aceptado o publicado y cumple con los requisitos institucionales (en caso que proceda)
En caso de Tesis por artículos científicos publicados						
N.A						Aceptación o Publicación de los artículos según el nivel del programa
N.A						El estudiante es el primer autor
N.A						El autor de correspondencia es el Tutor del Núcleo Académico Básico
N.A						En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabajo de investigación.
N.A						Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados
N.A						La aceptación o publicación de los artículos en revistas indexadas de alto impacto

Con base a estos criterios, se autoriza se continúen con los trámites de titulación y programación del examen de grado:

Sí X
No

Elaboró:

* NOMBRE Y FIRMA DEL CONSEJERO SEGÚN LA LGAC DE ADSCRIPCIÓN:

FIRMAS
P.D.

Dr. Jaime Muñoz Arteaga

NOMBRE Y FIRMA DEL SECRETARIO TÉCNICO:

Dr. Francisco Javier Álvarez Rodríguez

* En caso de conflicto de intereses, firmará un revisor miembro del NAB de la LGAC correspondiente distinto al tutor o miembro del comité tutorial, asignado por el Decano

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO:

Dr. Alejandro Padilla Díaz

Autorizó:

NOMBRE Y FIRMA DEL DECANO:

M. en C. Jorge Martín Alférez Chávez

Nota: procede el trámite para el Depto. de Apoyo al Posgrado

En cumplimiento con el Art. 105C del Reglamento General de Docencia que a la letra señala entre las funciones del Consejo Académico: Cuidar la eficiencia terminal del programa de posgrado y el Art. 105F las funciones del Secretario Técnico, llevar el seguimiento de los alumnos.



A template-based algorithm by geometric means for the automatic and efficient recognition of music chords

Rubén Hernández¹ · Antonio Guerrero² · Jorge E. Macías-Díaz^{3,4} 

Received: 29 May 2022 / Revised: 8 August 2022 / Accepted: 14 August 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

In this work, we introduce a template-based computational method to recognize chords through an audio recording of a musical instrument. The algorithm is based on a temporal frequency analysis using Gabor's filter banks. These filters are centered over adjusted frequencies of musical notes in different octaves and the adjustment is accomplished in terms of the detunings on the recording. Using the results in the filtering stage, a geometric mean of each chord is calculated. It is important to mention that these statistics are calculated from the combination of notes that form each chord and are automatically grouped as templates. The presence of chords is determined from these metrics. Several experiments are carried out for major, minor, augmented, diminished and suspended chords played on acoustic guitar, classic guitar, electric guitar, piano and ukulele. A comparative study against machine-learning classifiers is presented. The results show a superior performance of the present approach. In addition, the proposed method presents the advantage that it does not require a training stage, in contrast with the methods based on machine-learning algorithms. This reduces significantly the storage and time required for processing.

Keywords Music chords recognition · Automatic classification · Geometric means · Signal processing · Gabor filters

Rubén Hernández, Antonio Guerrero, Jorge E. Macías-Díaz: These authors contributed equally to this work.

Jorge E. Macías-Díaz
jmacias@comciencias.mx

Rubén Hernández
rubende21@gmail.com

Antonio Guerrero
aguerro@comciencias.mx

¹ Centro de Ciencias Exactas, Universidad Autónoma de Aguascalientes, Avenida Universidad 940, Ciudad Universitaria, 20100 Aguascalientes, Ags., México

² Departamento de Estadística, Universidad Autónoma de Aguascalientes, Avenida Universidad 940, Ciudad Universitaria, 20100 Aguascalientes, Ags., México

³ Department of Mathematics and Statistics of Mathematics, Toluca University, Nueva unit 25, 10010 Toluca, Mexico

⁴ Departamento de Matemáticas y Física, Universidad Autónoma de Aguascalientes, Avenida Universidad 940, Ciudad Universitaria, Aguascalientes 20100, Ags., México

1 Introduction

One of the problems in the intersection between digital audio processing and music is the development of systems of transcription and musical analysis. The relevance of these topics lies in that these systems can be used in music teaching [1], musical data bases, automatic accompaniment [2], transcription, improvisation systems and musical research [3]. Due to the complexity of music itself and pragmatic needs in this field, there is a research area devoted to developing systems and algorithms to analyze sounds produced by musical instruments [4]. In particular, the automatic detection of chords played on instruments has a practical importance in view of all the potential applications. This problem has been tackled by [5, 6], who proposed two approaches for chord recognition systems, namely, using chord templates or employing classification algorithms. Most of the recent efforts are mainly centered in methods based on deep learning [7–11]. It is worth pointing out that these approaches use data bases of popular songs in the training stage.

The objective of the present work is to develop an algorithm to recognize the chords of a musical piece through an audio recording. The recognition must be precise and robust. The methodology introduced here will be based on the use



Artificial neural network-based positioning error modeling and compensation for low-cost encoders of four-wheeled vehicles

Rubén Hernández¹ · Antonio Guerrero-Díaz-de-León² · Jorge E. Macías-Díaz^{3,4}

Received: 2 February 2024 / Revised: 28 February 2024 / Accepted: 25 March 2024

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

Several academic and budgetary applications in robotics use low-cost encoders which usually present errors inherent to the fabrication of their components or the surrounding environment. However, the data gathered from these sensors could be used successfully if the estimations based on the data are be compensated. This note presents an efficient method to compensate the error of estimation for the distance traveled by robotic four-wheeled vehicles with a speed control based on pulse width modulation. The current approach uses a methodology based on artificial neural networks which compensate the estimation error and enables a way to model the error for a further method of position estimation. Precisely, the model proposed in this work is based on a neural network with one hidden layer and five nodes. Our approach uses the information from the pulse width modulation value to control the speed of the vehicle as well as the values for the tick counting of the encoders in the four wheels. The back-propagation algorithm was used to calculate the weights of the nodes in the neural network. This method showed an improvement in the results with an error comparable to the intrinsic error due to the precision of the sensor, and gives an error model based on a zero-mean normal distribution.

Keywords Artificial neural networks · Error compensation · Low-cost optimal incremental encoders · Pulse width modulation · Statistical error modeling

1 Introduction

In a wide range of robotics applications, educational projects and research prototypes, the most usual form of getting information to estimate the distance followed by a mobile robot is

based on the use of incremental optical encoders in the wheels of vehicles [1]. However, these selected encoders are frequently not sufficiently precise or accurate [2, 3]. The inaccuracies could be due to different factors such as the fabrication, the quality and precision of their parts and the surrounding environment. Indeed, the estimations might not be exact because of these facts, though the present manuscript aims at presenting a methodology where the usage of a simple budget module HC-020K (which works as an incremental encoder with a single output digital signal) can be used to estimate the distance traveled by a vehicle in a precise way [4].

The importance of encoders in control systems and robotics lies in that the sensors give information about the position change of a mobile part of the system [5, 6]. For example, encoders on the wheels of a vehicle could describe the displacement as a function of the data from the sensor. In those cases (as well as in the particular problem tackled in the present report), the sensor is a rotary encoder [7, 8]. From a more technical point of view, an encoder is a device that is used as sensor and turns a linear or rotational position into a signal. In turn, an incremental rotary encoder is an electro-mechanical device which converts an angular displacement

✉ Jorge E. Macías-Díaz
jmacias@correo.cim.unam.mx

Rubén Hernández
rubenher23@gmail.com

Antonio Guerrero-Díaz-de-León
aguerro@correo.cim.unam.mx

¹ Centro de Ciencias-Básicas, Universidad Autónoma de Aguascalientes, Avenida Universidad 940, Ciudad Universitaria, Aguascalientes 20100, Ags., México

² Departamento de Estadística, Universidad Autónoma de Aguascalientes, Avenida Universidad 940, Ciudad Universitaria, Aguascalientes 20100, Ags., México

³ Departamento de Matemáticas, Tulane University, New Orleans 70118, EE.UU.

⁴ Departamento de Matemáticas y Física, Universidad Autónoma de Aguascalientes, Avenida Universidad 940, Ciudad Universitaria, Aguascalientes 20100, Ags., México



A method for obstacle-range estimation based on echo-location using high-end sound with applications

J. Antonio Guerrero-Díaz-de-León^{1,5} · Rubén Hernández-Torres³ · Siegfried Macías^{2,3} · Jorge E. Macías-Díaz^{4,5}

Received: 12 November 2024 / Revised: 18 December 2024 / Accepted: 27 December 2024 / Published online: 9 January 2025
© The Author(s) 2025, under license to Springer-Verlag GmbH Germany, part of Springer Nature 2025

Abstract

Efficient mapping algorithms have been implemented in different fields of robotics applications. Some of those applications are in the mapping of environments with difficult accessibility, the planning of service or emergency assistant robots, or as supporting tools in helping people with disabilities. In those examples, the essential task of a mapping algorithm is to detect obstacles which may interfere in motion paths. One of the most widely used characteristics in pulse-echo range estimation is the time of flight. However, the ultrasonic frequencies (range from 30 to 120 KHz) are poorly suitable for the long distances faced in in-door applications. In view of the importance of developing efficient mapping algorithms, the present manuscript reports an efficient method based on high-frequency waves in order to range static objects under a controlled environment. In the present approach, the signal acquisition is made out-of-band of the processing algorithm. This process is carried out by using an active speaker to draw the control ping sound. The resulting audio is captured through a unidirectional dynamic microphone Sennheiser® SV100. The audio-signal processing is based on selective filtering via Gabor models and wavelet decimation de-noising. The proposed method is compared against two common methodologies for range estimation in robotics based on ultrasonic and infrared sensor. The current methodology demonstrated a competitive and, in some cases, better performance than the standard methods.

Keywords Efficient mapping algorithms · Obstacle-range estimation · High-frequency sound · Echo-location · Gabor model de-filtering · Wavelet decimation de-noising

J. Antonio Guerrero-Díaz-de-León
jaguerrero@ciqa.unam.mx
Rubén Hernández-Torres
rubentor23@gmail.com
Siegfried Macías
siegfriedmacias@ciqa.unam.mx
Jorge E. Macías-Díaz
jorge@ciqa.unam.mx, jmacias@ciqa.unam.mx

¹ Department of Statistics, Autonomous University
of Aguascalientes, Aguascalientes, Mexico

² University Center of Los Lagos, University of Oaxaca,
Tlaxiaco, Mexico

³ Faculty of Sciences, Autonomous University
of Aguascalientes, Aguascalientes, Mexico

⁴ Department of Mathematics and Data Mining Mathematics
School of Digital Technologies, Technical University of Madrid,
Madrid

⁵ Department of Mathematics and Physics, Autonomous
University of Aguascalientes, Avenida Universidad 940,
Aguascalientes, 20121 Aguascalientes, Ags., Mexico

1 Introduction

The acquisition of spatial models without human intervention finds a wide range of applications, which go down from mapping inaccessible or dangerous places to the building of systems that may help in daily tasks [1–3]. Among the last class of applications, some of them may involve wandering into an environment with different obstacles. In general, the mapping done by an autonomous system is composed of two main processes [4]: self-localization and obstacle range estimation. The first process grants the robot a reference framework to locate itself and locate obstacles in the environment. Meanwhile, the second step allows the autonomous system to measure the distance between it and the obstacles, based on the same frame.

Among the different approaches for autonomous mapping, audio-oriented techniques occupy an important role [5]. Various examples of audio-oriented implementations for static obstacle ranging could be mentioned within surveillance and navigation problems [6, 7]. This mapping

Agradecimientos

El presente trabajo no hubiese sido posible sin el apoyo de muchas personas.

Agradezco al Doctor José Antonio Guerrero por orientarme y encaminarme durante la realización de este trabajo así como todo el tiempo dedicado. Agradezco de igual manera al Doctor Jorge Macías Díaz y el Doctor Ángel Muñoz Zavala por su valioso apoyo, estuvieron siempre dispuestos a resolver mis dudas y preguntas. Así mismo doy las gracias a todos mis profesores que con sus materias me proporcionaron las herramientas necesarias para concluir mi investigación. Igualmente expreso mi gratitud con el personal de la Universidad Autónoma de Aguascalientes quienes facilitaron mi estancia en la institución así como todos los trámites que fueron necesarios.

Agradezco al pueblo de México quién a través de SECIHTI, y por medio de su programa de becas, recibí el apoyo económico para poder concluir mis estudios de doctorado.

De manera muy especial agradezco a mis papás Rubén y Gaby por haberme apoyado en continuar mis estudios así como a mis hermanos, Alexis y Erick que siempre me impulsaron a ser mejor. Así mismo agradezco a mis amigos por toda su motivación, quienes sin su apoyo y confianza ésto no hubiera sido posible.

Índice general

1. Localización y mapeo simultáneo	15
1.1. Robótica probabilística	16
1.2. Estimación recursiva de estados	16
1.3. Filtro de Bayes	17
1.4. Predicción: estimación de la posición de un robot móvil	18
1.5. Corrección: actualización mediante sensores	22
1.6. FastSLAM	23
2. Algoritmo para el reconocimiento automático de acordes musicales usando plantillas basadas en la media geométrica	31
2.1. Marco teórico	32
2.2. Metodología	37
2.3. Experimentos	42
2.4. Conclusiones	50
3. Modelado y compensación del error en la estimación de la posición con encoders de bajo costo en robots de cuatro ruedas con redes neuronales artificiales	53
3.1. Marco teórico	53
3.2. Metodología	56
3.3. Experimentos	59
3.4. Conclusiones	62
4. Método para la estimación de rango de obstáculos usando eco-localización con sonido de alta frecuencia	65
4.1. Marco teórico	65
4.2. Metodología	71
4.3. Experimentos	74
4.4. Conclusiones	79

ÍNDICE GENERAL

ÍNDICE GENERAL

5. Conclusiones y trabajo a futuro	81
5.1. Conclusiones	81
5.2. Trabajo a futuro	82



Índice de figuras

1.1. Esquema del modelo cinemático para un robot móvil con dirección por derrape (<i>SSMR Skid-Steering Mobile Robot</i>)[18]. COM (Center of Mass).	20
1.2. Cálculo de la velocidad de una llanta[18].	20
1.3. Esquema del funcionamiento de un encoder óptico.	22
1.4. Forma de la trayectoria propuesta en el modelo odométrico de movimiento[37].	25
1.5. Distribución del error propuesto para el método FastSLAM[37] para un sensor de distancia.	27
1.6. Modelo del cono de medición de un sensor de rango[37]. El cono muestra tres zonas importantes: en gris celdas del mapa que no se tiene suficiente certeza que están ocupadas, celdas en claro que representa celdas libres de obstáculos y celdas oscuras que indican con mayor certeza que están ocupadas.	29
2.1. Modelo del algoritmo propuesto	38
2.2. Filtros de Gabor correspondientes a una nota en particular antes y después del ajuste con respecto a la frecuencia nominal. Las líneas verticales punteadas corresponden a los pasos posibles dentro del proceso de ajuste, mientras que las dos líneas verticales de guiones corresponden a las frecuencias centrales del filtro original y ajustado.	39
2.3. Resultado de la detección de acordes en la secuencia A (C Cm E Edim Dsus4 C#aug) para el Experimento 1.	43
2.4. Resultado de la detección de acordes en la secuencia B (Am Baug Esus4 Dsus4 A#m) para el Experimento 1.	44
2.5. Resultado de la detección de acordes en la secuencia C (F# Gm Gdim D# Bm Bsus2) para el Experimento 1.	45
2.6. Resultado de la detección de acordes en la secuencia D (Csus4 Eaug G Dbdim F Em) para el Experimento 1.	46

2.7. Resultado de la detección de acordes para las secuencias A (Asus2 A $\sharp$ dim Caug C $\sharp$ sus4 D $\sharp$ G $\sharp$ m), B (Bdim C C $\sharp$ m Csus2 Daug G $\sharp$ sus4), C (Am A $\sharp$ C $\sharp$ dim Dm Faug Gm) y D (Bsus2 Csus4 D $\sharp$ aug E Edim F $\sharp$ m) del Experimento 2.	48
2.8. Resultado de la detección de acordes en la secuencia A $\sharp$ sus2, B, Caug, Ddim, Esus4 y F $\sharp$ m para el Experimento 3 en guitarra clásica, guitarra eléctrica, piano y ukulele.	50
3.1. Representación de una red neuronal artificial de dos entradas y una capa oculta. En esta representación, los círculos vacíos representan las variables de cada capa, las líneas representan la conexión entre las neuronas que indican que el valor de una neurona en específico depende de neuronas previas. Los círculos con un valor unitario (1) dentro representan los términos de bias.	55
3.2. Fotografía del prototipo empleado.	57
3.3. Marco de referencia para los experimentos realizados[11]	58
3.4. Composición del grupo de datos de entrenamiento del modelo de red neuronal artificial.	58
3.5. Grafo correspondiente al modelo de red neuronal propuesto.	60
3.6. Histogramas de los valores de error de estimación (a) antes de la red neuronal propuesta, y (b) después de la red neuronal propuesta. . . .	61
3.7. Comparación de la distribución de error para el grupo de datos de prueba de la red neuronal propuesta contra una función gaussiana y una función triangular centradas en la media de los errores.	62
3.8. Resultados de la estimación de la distancia recorrida con la expresión lineal del encoder y con la red neuronal propuesta.	63
4.1. Esquema de la metodología de descomposición wavelet multi-nivel. . .	70
4.2. Esquema de la metodología de reconstrucción wavelet multi-nivel . .	71
4.3. Gráficas que muestran cómo cada método de umbralizado afecta a una función identidad $y = x$	72
4.4. Diagrama de la metodología propuesta.	73
4.5. Respuesta en frecuencia y patrón polar del micrófono Shure [®] SV-100. Figura tomada del manual del dispositivo.	74
4.6. Se muestra en azul el espectro de la grabación correspondiente a la estimación de rango utilizando un tren de pulsos de una señal sinusoidal de 15KHz. En naranja se muestra la respuesta unitaria de un filtro de gabor centrado a 15 KHz con valor de σ 300[13].	75

4.7. Resultado del procesamiento de una señal sinusoidal con un obstáculo a 30cm: (a) señal original obtenida a través del micrófono, (b) onda después del filtrado de Gabor en frecuencia, (c) señal después de la eliminación de ruido con wavelets y (d) onda umbralizada[13].	77
4.8. Resultado del procesamiento de una señal sinusoidal con un obstáculo a la mínima distancia permitida por el sistema: (a) señal original obtenida a través del micrófono, (b) onda después del filtrado de Gabor en frecuencia, (c) señal después de la eliminación de ruido con wavelets y (d) onda umbralizada[13].	79
4.9. Comparación entre el método propuesto, un sensor ultrasónico y un sensor infrarrojo para la estimación de distancia. Los experimentos consistieron en realizar 5 mediciones de distancia entre 20 cm y 200 cm con incrementos de 20 cm utilizando cada método[13].	80

TESIS TESIS TESIS TESIS TESIS

ÍNDICE DE FIGURAS

ÍNDICE DE FIGURAS



TESIS TESIS TESIS TESIS TESIS

TESIS TESIS TESIS TESIS TESIS

Índice de tablas

2.1.	Tabla de confusión para las combinaciones de notas (números en comparativa) en las progresiones A, B, C y D del Experimento 2.	49
2.2.	Medidas de calidad para el Experimento 3.	49
2.3.	Tabla de confusión para las combinaciones de notas (números en comparativa) en la progresión ejecutada en guitarra clásica, guitarra eléctrica, piano y ukelele del Experimento 3.	51
2.4.	Medidas de calidad para el Experimento 3.	51
3.1.	WT1, WT2, WT3 y WT4 corresponden al número de tick para cada una de las llantas, W_m indica la media de los ticks de todas las llantas, ED es la distancia estimada mediante la expresión 3.6 y MD es la distancia medida[11].	57
3.2.	Configuraciones de capas ocultas utilizadas en la selección del modelo de red neuronal. Los números entre paréntesis corresponden al número de neuronas en las capas ocultas del modelo.	59
4.1.	Tabla de la estimación de la distancia a un obstáculo usando cada una de las formas de ondas propuestas. Se emplea un tren de pulsos compuesto por 5 ciclos completos de la onda a una frecuencia de 15 KHz.	76
4.2.	Resultados comparativos de las estimaciones obtenidas con el método propuesto, un sensor ultrasónico y un sensor infrarrojo. Cada experimento consistió en realizar 5 mediciones de distancia entre 20 cm y 200 cm con incrementos de 20 cm utilizando cada método. La tabla presenta el promedio de las estimaciones, la desviación estándar y el error absoluto medio para cada medición.	78

Resumen

En este trabajo se presenta el desarrollo de un método para la localización y mapeo simultáneo por un robot móvil de cuatro ruedas empleando sonido para el reconocimiento de su entorno. Se muestra la implementación de un prototipo funcional de robot móvil así como de los algoritmos de estimación cinemática, medición y mapeo con sensores de obstáculos empleando sonido. Así mismo, el trabajo se integra con tres resultados publicados relacionados con la totalidad del proyecto. El artículo 'A template-based algorithm by geometric means for the automatic and efficient recognition of music chords'[12], integra fundamentos del procesamiento digital de audio basado en filtros de Gabor y un ejemplo de uso en el análisis y clasificación de acordes musicales. Así mismo, presenta las técnicas empleadas posteriormente para la estimación de rango de obstáculos basada en eco-localización. La publicación 'Artificial neural network based positioning error modeling and compensation for low-cost encoders of four-wheeled vehicles'[11] describe la metodología propuesta para la compensación del error de estimación de la posición de un vehículo de cuatro ruedas. Los resultados de este trabajo se incluyeron en el algoritmo de SLAM propuesto dentro de la etapa de localización del robot móvil. Se basó la estimación y compensación del error producido por los materiales y el entorno en un modelo de redes neuronales artificiales. El trabajo publicado como 'A method for obstacle-range estimation based on echo-location using high end sound with applications'[13] presenta la metodología y algoritmo propuesto para un sistema de estimación de rango basado en sonido de alta frecuencia dentro del espectro audible. En él se propone un sistema encaminado a ser empleado en la etapa de mediciones dentro del algoritmo de SLAM propuesto. Por último, se muestran las conclusiones generales del trabajo realizado y posibles líneas de investigación derivadas de los resultados obtenidos.

Abstract

This document presents the development of a method for simultaneous localization and mapping by a four-wheeled mobile robot using sound for environment recognition. A functional prototype of a mobile robot is implemented, along with the kinematic estimation, measurement, and mapping algorithms using sound-based obstacle sensors. Additionally, this work integrates three published results related to the entire project. The article 'A template-based algorithm by geometric means for the automatic and efficient recognition of music chords'[12] incorporates fundamentals of digital audio processing based on Gabor filters and provides an example of its use in the analysis and classification of musical chords. Furthermore, it presents the techniques later used for obstacle range estimation based on echolocation. The publication 'Artificial neural network based positioning error modeling and compensation for low-cost encoders of four-wheeled vehicles' [11] describes the proposed methodology for compensating the positioning estimation error of a four-wheeled vehicle. The results of this work were incorporated into the proposed SLAM algorithm within the localization stage of the mobile robot. The estimation and compensation of errors caused by materials and the environment were based on an artificial neural network model. The work published as 'A method for obstacle-range estimation based on echo-location using high-end sound with applications'[13] presents the proposed methodology and algorithm for a range estimation system based on high-frequency sound within the audible spectrum. This system is intended for use in the measurement stage of the proposed SLAM algorithm. Finally, the general conclusions of the research are presented, along with possible future research directions derived from the obtained results.

TESIS TESIS TESIS TESIS TESIS

ÍNDICE DE TABLAS

ÍNDICE DE TABLAS



TESIS TESIS TESIS TESIS TESIS

TESIS TESIS TESIS TESIS TESIS

Introducción

Un robot es una máquina capaz de realizar acciones complejas de manera automática. Así mismo, un robot autónomo es un robot que puede ejecutar sus acciones con cierto grado de independencia. Algunas características que cumple un robot autónomo son:

- Obtiene información de su entorno.
- Funciona por largos periodos de tiempo sin intervención del humano.
- Interactúa con su entorno sin asistencia humana.

La navegación en robótica consiste en el campo de estudio del cómo es la manera en que se relaciona un agente o robot con los elementos que componen su entorno, esto con una multitud de objetivos previamente establecidos. Las tareas más importantes dentro del campo de la navegación son la localización y reconocimiento de parte del agente robótico y el mapeo del entorno en el cual éste transita.

Los procesos de localización, reconocimiento y mapeo se reúnen en el concepto de SLAM (por sus siglas en inglés *Simultaneous Localization and Mapping*), el cual es un problema de la geometría computacional y la robótica que se encarga de construir o actualizar el mapa de un ambiente desconocido mientras registra la localización del agente o vehículo. Existen diversos medios para el análisis y recreación de entornos a través de la exploración con robots los cuales están basados principalmente en visión computacional y el procesamiento de audio.

Planteamiento del problema

El desarrollo de algoritmos eficientes de mapeo, localización y navegación permite su implementación en robots móviles (por ejemplo vehículos de cuatro ruedas), los cuales pueden ser empleados en el diseño de planos de entornos inaccesibles para el hombre, construcción de robots de servicio o asistentes para emergencias.

TESIS TESIS TESIS TESIS TESIS

Dentro del proceso de localización y navegación a través de un entorno desconocido, los medios principales para reconocer el ambiente y tomar decisiones son la visión y el sonido.

Si bien que la visión podría considerarse el medio más inmediato para el contacto con el medio, el sonido en el entorno y sus interacciones proporcionan una enorme cantidad de información que puede ser explotada.

Así pues, algunos casos donde el procesamiento de audio significa un proceso importante para un sistema de reconocimiento, navegación y mapeo:

- El procesamiento de audio puede implicar un menor número de recursos computacionales.
- Uso en medios donde se carece de visibilidad.
- El procesamiento en binomio de visión-sonido propiciaría el desarrollo de sistemas más precisos.
- Es mas fácilmente implementado en dispositivos de apoyo para personas con alguna discapacidad.

Objetivo general

El objetivo general es diseñar un algoritmo usando la interacción de las ondas de sonido con el medio a partir del cual un robot autónomo móvil reconozca su entorno.

Objetivos particulares

- Desarrollar un prototipo de robot móvil que hará de agente dentro de un proceso de localización y mapeo simultáneo.
- Implementar los algoritmos de estimación cinemática, medición y mapeo con sensores de obstáculos empleando sonido.
- Implementar una metodología de ajuste de error en la estimación de posición del vehículo involucrado en el proceso de SLAM propuesto.
- Desarrollar un algoritmo que a través de la interacción de las ondas de sonido con el entorno dé una descripción del mismo.
- Obtener un mapeo inicial de un entorno dadas las implementaciones de algoritmos y prototipos propuestos.

Desarrollo de la tesis

El presente trabajo se integra de la siguiente manera:

En el Capítulo 1 se desarrollan las bases teóricas que sustentan los trabajos publicados así como el contexto de los mismos dentro de la totalidad del proyecto de investigación propuesto.

El Capítulo 2 presenta los resultados publicados en el artículo *A template-based algorithm by geometric means for the automatic and efficient recognition of music chords*[12]. Se integran fundamentos del procesamiento digital de audio basado en filtros de Gabor y un ejemplo de uso en el análisis y clasificación de acordes musicales. Así mismo, presenta las técnicas empleadas posteriormente para la estimación de rango de obstáculos basada en eco-localización.

En el Capítulo 3 se describe la metodología propuesta para la compensación del error de estimación de la posición de un vehículo de cuatro ruedas. El algoritmo y resultados corresponden a la publicación *Artificial neural network-based positioning error modeling and compensation for low-cost encoders of four-wheeled vehicles*[11]. Los resultados de este trabajo se incluyeron en algoritmo de SLAM propuesto dentro de la etapa de localización del robot móvil. Se basó la estimación y compensación del error producido por los materiales y el entorno en un modelo de redes neuronales artificiales.

El Capítulo 4 presenta la metodología y algoritmo propuesto para un sistema de estimación de rango basado en sonido de alta frecuencia dentro del espectro audible. Los resultados fueron publicados en *A method for obstacle-range estimation based on echo-location using high-end sound with applications*[13]. El trabajo propone un sistema encaminado a ser empleado en la etapa de mediciones dentro del algoritmo de SLAM propuesto.

Por último, el Capítulo 5 muestra las conclusiones generales del trabajo realizado y posibles líneas de investigación derivadas de los resultados obtenidos.

TESIS TESIS TESIS TESIS TESIS



TESIS TESIS TESIS TESIS TESIS

Capítulo 1

Localización y mapeo simultáneo

La localización y mapeo simultáneo (por sus siglas en inglés SLAM) se encamina a la solución del problema de obtener un mapeo espacial de un entorno, mientras que de manera simultánea se obtiene la localización del robot con respecto a este modelo. La construcción de métodos robustos para el mapeo de entornos sin estructura, dinámicos o de gran escala aún se presenta como un caso de estudio abierto [38].

El problema SLAM se define como:

Dado un robot móvil dentro de un entorno desconocido, este arranca en una posición inicial x_0 . Su movimiento cuenta con cierta incertidumbre lo cual provoca, de manera gradual, que sea más difícil determinar su posición actual en función de un marco de referencia global. Conforme el robot deambula, éste puede tomar mediciones del entorno a través de un sensor con ruido inherente. Así, la finalidad consiste en construir un mapa del entorno mientras el robot determina su posición de manera simultánea, de manera relativa al mapa y basado en datos con ruido.

Las tres tendencias principales para resolución del problema son[37]:

- Filtro de Kalman extendido.
- Filtro de partículas.
- Técnicas de optimización basadas en grafos.

La precisión en los sistemas robóticos, y en particular en los algoritmos SLAM es fundamental. La primicia consiste en diseñar sistemas inteligentes que asistan en la toma de decisiones de los dispositivos de control, sustentados en modelos de la incertidumbre intrínseca de los fenómenos del mundo físico, así como los factores que contribuyen a los mismos. Algunos de estos factores son:

- La incertidumbre del entorno de un robot es impredecible, particularmente en entornos dinámicos o que no están estructurados en una forma en particular.
- Los sensores se limitan para lo que están diseñados para medir. Así mismo, se agregan limitaciones de índole física, de resolución o de ruido al cual están sometidos.
- Los actuadores de los robots presentan incertidumbre en su funcionamiento, debido a ruido, desgaste o fallas mecánicas.
- Los modelos en sí mismos son una aproximación del mundo real. Así pues, de manera inherente, cuentan con un error que generan incertidumbre en el funcionamiento de sistemas basados en procesos físicos.

1.1. Robótica probabilística

La robótica probabilística es un acercamiento hacia el diseño e implementación de sistemas de control tomando en cuenta la incertidumbre que presentan estos sistemas y emplearla tal cual usando cálculos de teoría de la probabilidad.

Así pues, los algoritmos probabilísticos representan la información en forma de distribuciones de probabilidad sobre la totalidad de un espacio de estados dinámicos, proporcionando un grado de certidumbre a cada uno basado en un sustento matemático.

Basados en este paradigma, se abordan dos problemas dentro de la robótica: percepción del robot y control y planeación.

1.2. Estimación recursiva de estados

Un sistema físico puede ser modelado matemáticamente como un conjunto de entradas, salidas y variables de estado relacionadas entre sí. Las variables de estado son parámetros dentro del sistema dinámico que evolucionan a lo largo del tiempo y cuyos valores dependen del valor que tienen en un tiempo determinado así como del valor de los parámetros de entrada del sistema. Así mismo, los valores de salida dependen de los valores de las variables de estado.

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t) + D(t)u(t)$$

donde $\dot{x}(t)$ es el vector de estado cambiante en el tiempo (no siempre observable directamente), $u(t)$ es la entrada de control del sistema, y es la salida medible, y A ,

B , C , D son matrices que describen la dinámica del sistema.

La base de la robótica probabilística está en estimar estados a partir de la información obtenida por sensores. La estimación de estados parte del problema de estimar datos que no son directamente observables a partir de información proveniente de sensores o que no son enteramente creíbles por diferentes factores como el ruido. Sin embargo, se asume que esta información puede ser inferida con cierta precisión.

El grado de confianza (o por su expresión en inglés *belief*) dentro del contexto de la robótica probabilística, refleja la certeza interna de un robot acerca de un estado de su entorno en un momento determinado. El belief se representa a través de una probabilidad condicional donde su distribución es una probabilidad a posteriori sobre las variables de estado, y condicionadas a los datos disponibles.

Se define el belief sobre una variable de estado x_t como la distribución de probabilidad a posteriori en el tiempo t condicionada a todas las mediciones pasadas, la medición actual y todos los controles pasados y el actual como:

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

donde $z_{1:t}$ corresponde a la serie de mediciones observadas desde la primer instante hasta el tiempo t y $u_{1:t}$ los controles correspondientes para cada instante de tiempo desde el inicio hasta el tiempo t .

Así mismo, se establece como predicción, al cálculo de la distribución de probabilidad de un estado x_t dadas las mediciones pasadas y el control actual y controles pasados, definiéndose como:

$$\overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t})$$

donde $z_{1:t-1}$ corresponde a la serie de mediciones observadas desde el inicio hasta un instante previo al tiempo t .

El cálculo de $bel(x_t)$ es llamado corrección o actualización de medición y $\overline{bel}(x_t)$ predicción.

1.3. Filtro de Bayes

Dentro de la teoría de procesos estocásticos, el filtrado consiste en el problema de determinar el estado de un sistema a partir de un conjunto de observaciones ya sea incompleto o que potencialmente ha sido sometido a ruido[36]. Este problema puede ser abordado mediante el cálculo de certidumbre o *belief* sobre los estados presentes en un sistema dinámico.

Para fines prácticos y de complejidad computacional se asume que los sistemas a resolver cumplen con el principio de Markov:

La distribución de probabilidad del valor futuro de una variable aleatoria depende únicamente de su valor actual.

Por ende, la representación y análisis de los sistemas dinámicos dentro del espacio de estados puede analizarse desde la perspectiva de un proceso de Markov[36].

El algoritmo más general para el cálculo de beliefs en un sistema dinámico robótico está dado por el algoritmo de filtrado de Bayes. Este algoritmo calcula las distribuciones de certidumbre a partir de datos de control y mediciones.

```

Bayes_Filter( $bel(x_{t-1}), u_t, z_t$ ):
  for all  $x_t$  do
     $\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1})dx_{t-1}$ 
     $bel(x_t) = \eta p(z_t|x_t)\overline{bel}(x_t)$ 
  endfor
  return  $bel(x_t)$ 

```

Donde $bel(x_{t-1})$ es el valor del belief *a priori*, u_t es la función de control aplicada al sistema robótico, z_t las mediciones recabadas por los sensores, η un factor de normalización de probabilidad y $bel(x_t)$ el belief *a posteriori*. Así pues, la estimación de beliefs se realiza de manera recursiva.

Así pues, dentro del proceso de filtrado estocástico podemos identificar dos etapas: La etapa de **predicción** y la etapa de **corrección**. Estos pasos están definidos cada uno por un modelo de movimiento y un modelo de medición, donde a partir de cada uno se obtiene una probabilidad de transición sobre las cuales se realiza la estimación del estado real del sistema dinámico. Estos dos procesos son identificados dentro del filtro de Markov en la estimación de $\overline{bel}(x_t)$ y $bel(x_t)$.

1.4. Predicción: estimación de la posición de un robot móvil

Una de las tareas más importantes dentro de las aplicaciones de robots móviles, consiste en estimar la distancia recorrida por este mismo, y por consiguiente su ubicación después de un movimiento.

La navegación con base a estimadores (en inglés referido como *dead reckoning*) consiste en procedimiento matemáticos y computacionales cuya finalidad es la de estimar la ubicación actual de un vehículo en función de su estado pasado y del rumbo y velocidad de sus movimientos a lo largo del tiempo, donde dicha información es obtenida únicamente a través de sensores a bordo del vehículo.

De igual manera, se emplean modelos matemáticos para la estimación de la posición de un robot móvil, visualizándolo como un subsistema compuesto por elementos cinemáticos y dinámicos[18][26][3]. De acuerdo a [18], para un robot móvil con dirección por derrape (por sus siglas en inglés *SSMR Skid-Steering Mobile Robot*), se puede construir su modelo cinemático asumiendo que el vehículo se encuentra en una superficie plana y dada una base ortonormal.

Los movimientos básicos de un Skid-Steering Mobile Robot tomando en cuenta la velocidad angular de las llantas de cada lado del vehículo son:

- Movimiento hacia adelante (Forward): Ambas ruedas giran en la misma dirección y a la misma velocidad angular ($\omega_1 == \omega_2$). El robot se mueve en línea recta hacia adelante.
- Movimiento hacia atrás (Backward): Ambas ruedas giran en la misma dirección pero en sentido contrario al movimiento hacia adelante, y a la misma velocidad angular ($-\omega_1 == -\omega_2$). El robot se mueve en línea recta hacia atrás.
- Giro en el lugar (Rotación en el punto, Zero-radius turn): Las ruedas giran en direcciones opuestas ($\omega_1 == -\omega_2$). El robot gira sobre su propio eje (centro de rotación en el centro del robot).
- Giro con radio de curvatura (Curved motion): Las ruedas giran en la misma dirección pero a velocidades angulares diferentes ($\omega_1 \neq \omega_2$). El robot describe una trayectoria curva con un radio de curvatura determinado por la diferencia de velocidades.
- Movimiento lateral (Skidding): Dependiendo de la fricción y el diseño del robot, es posible generar un movimiento lateral deslizante intrínseco a la inercia del robot.

Ahora, suponiendo que el robot se mueve en el plano con una velocidad lineal dentro su marco de referencia local como $\mathbf{v} = [v_x, v_y, 0]^T$ y gira con una velocidad angular $\omega = [0, 0, \omega]^T$. Si $\mathbf{q} = [X, Y, \theta]^T$ es el vector de coordenadas globales del robot, entonces $[\dot{X}, \dot{Y}, \dot{\theta}]^T$ denota el vector de velocidades en función de las coordenadas globales del mismo. La Figura 1.1 muestra el diagrama del robot, su marco local y global de referencia, así como los vectores de velocidad y posición del mismo. Así mismo de la Figura 1.1 se pueden relacionar las variables $\dot{X}$ y $\dot{Y}$ con las coordenadas del vector local de velocidad como:

$$\dot{X} = [\cos \theta \quad -\sin \theta]$$

$$\dot{Y} = [\sin \theta \quad \cos \theta]$$

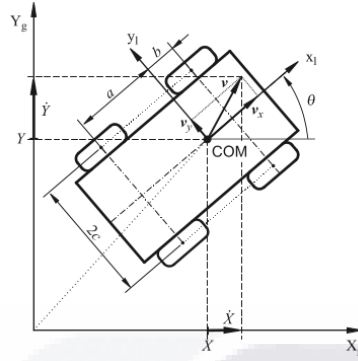


Figura 1.1: Esquema del modelo cinemático para un robot móvil con dirección por derrape (*SSMR Skid-Steering Mobile Robot*)[18]. COM (Center of Mass).

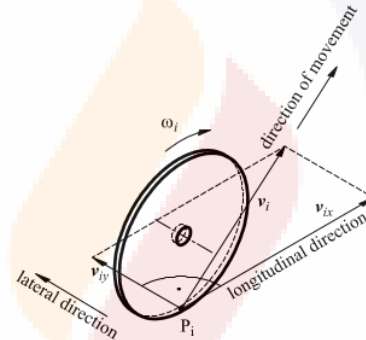


Figura 1.2: Cálculo de la velocidad de una llanta[18].

Suponiendo que cada i -ésima llanta del vehículo gira con una velocidad angular $\omega_i(t)$, donde $i = 1, 2, 3, 4$, ésta puede verse como una entrada de control del vehículo. Asumiendo un derrape lateral nulo del robot, se puede asumir la relación:

$$v_{ix} = r_i \omega_i$$

Donde v_{ix} es el componente longitudinal del vector total de velocidad $\mathbf{v}_i$ de la i -ésima llanta y r_i denota el radio de la llanta. La Figura 1.2 muestra una representación gráfica de una llanta y los vectores que actúan en ella.

Tomando en cuenta el centro de rotación instantáneo (en inglés *ICR Instantaneous Center of Rotation*) dadas las cuatro llantas en el movimiento del vehículo y asumiendo el radio efectivo de cada llanta $r_i = r$, podemos escribir:

$$\omega_\omega = \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \frac{1}{r} \begin{bmatrix} v_L \\ v_R \end{bmatrix}$$

Donde ω_L y ω_R son las velocidades angulares de las llantas izquierdas y derechas respectivamente. Así, podemos resumir lo anterior en las expresiones:

$$\psi = \begin{bmatrix} v_x \\ \omega_z \end{bmatrix} = r \begin{bmatrix} \frac{\omega_L + \omega_R}{2} \\ \frac{\omega_R - \omega_L}{2c} \end{bmatrix}$$

Donde ψ es una nueva entrada de control cinemático del vehículo. La velocidad generalizada $\dot{q}$ puede escribirse como:

$$\dot{q} = S(q)\psi$$

donde

$$S(q) = \begin{bmatrix} \cos(\theta) & x_{ICR} \sin(\theta) \\ \sin(\theta) & -x_{ICR} \cos(\theta) \\ 0 & 1 \end{bmatrix}$$

Con lo anterior como referencia, la cinemática en la robótica probabilística consiste en un conjunto de técnicas para describir el efecto de las acciones ejercidas en un robot y en cómo se refleja en su configuración o pose. Los modelos cinemáticos probabilísticos, o modelos de movimiento, de forma resumida tratan de describir la transición de estados de un robot con base en el control de acción cinemática ejercido sobre el mismo y al estado anterior a la acción. Esto puede ser modelado mediante una función de densidad condicional:

$$p(x_t | u_t, x_{t-1})$$

Donde x_t es el estado a estimar, u_t la función de control cinemática y x_{t-1} la posición anterior del robot.

Uno de los métodos más utilizados de localización en robótica es con base de modelos odométricos.

La **odometría** son técnicas en las cuales las estimaciones de cambio en posición de un vehículo son calculadas basadas en conteos de sensores o *encoders* instalados en las ruedas. El funcionamiento del sensor se fundamenta en un arreglo de una fuente de luz (por lo regular infrarroja) y un foto-sensor separados entre sí por un disco con muescas que permite o no el paso de la luz a través de él. El giro del disco produce un cambio constante en la salida del foto-sensor, la cual es entrada de un comparador para así generar una señal para ser procesada posteriormente. La Figura 1.3 muestra la disposición de los elementos de un encoder óptico así como la base de su funcionamiento. A través de la señal producida por el encoder, puede ser

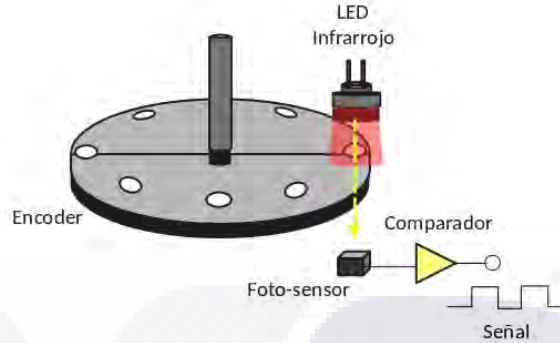


Figura 1.3: Esquema del funcionamiento de un encoder óptico.

estimada la distancia lineal que recorre una llanta con dicho encoder instalado. La distancia es en teoría una correspondencia lineal dada por:

$$\Delta l = D\pi \frac{ticks}{\mu}$$

Donde Δl corresponde a la estimación de distancia lineal recorrida por una llanta, D el diámetro de la llanta, $ticks$ el conteo de pulsos detectado por el encoder y μ el número total de huecos con el cual cuenta el disco del encoder. A pesar que la estimación es trivial en su forma, en una aplicación práctica, diferentes factores como los componentes del sensor, ruido, las condiciones físicas a las cuales se somete o el procesador de la señal producida por el encoder puede resultar en estimaciones erróneas que a largo plazo significaría un error acumulado considerable. Este tema será abordado tanto en el algoritmo propuesto de SLAM como en el Capítulo 3.

1.5. Corrección: actualización mediante sensores

El segundo paso involucrado en la estimación de estados basada en el filtros de Bayes aplicados en robótica corresponde al proceso de corrección (o actualización). Durante esta etapa se incorporan las mediciones obtenidas por los sensores del robot, refinando el valor de **belief** sobre un estado en particular. Así pues, dado un estado o posición del robot x_t , éste recibe una medición de sus sensores z_t la cual se asume está sometida a ruido. El proceso de corrección actualiza la certeza o *belief* acerca de su posición siguiendo la regla de Bayes:

$$P(x_t|z_t) = \frac{P(z_t|x_t) \cdot P(x_t)}{P(z_t)}$$

donde $P(x_t|z_t)$ corresponde al belief a posteriori después de incorporar las mediciones del sensor, $P(z_t|x_t)$ es la probabilidad de obtener dicha medición en la posición dada, $P(x_t)$ corresponde al valor de belief obtenido del proceso de predicción y $P(z_t)$ corresponde a la probabilidad marginal de las mediciones z_t . No se incluyen los controles u_t por conveniencia para enfocarse en la etapa de corrección. Este término representa la probabilidad total de observar la medición z_t y en este caso funciona como factor de normalización en la expresión. En secciones más adelante será introducido este concepto dentro del algoritmo SLAM propuesto y su relación con el artículo descrito en el Capítulo 4.

1.6. FastSLAM

El algoritmo FastSLAM emplea filtros de partículas para la estimación del camino recorrido por un robot móvil. Así mismo, una de las restricciones del método es asumir que cada uno de los errores que afectan a los elementos del mapa a construir son condicionalmente independientes.

Un filtro de partículas es un método de estimación secuencial de estados en sistemas dinámicos no lineales y no gaussianos. El filtro emplea algoritmos basados en muestras (partículas) para aproximar distribuciones de probabilidad que evolucionan a lo largo del tiempo. Un filtro de partículas aproxima la distribución *a posteriori* mediante este conjunto de partículas y sus pesos asociados, siguiendo las siguientes etapas dentro del proceso: Inicialización, predicción, actualización, normalización y re-muestreo.

Una de las ventajas clave del algoritmo FastSLAM consiste en que lleva un registro de múltiples relaciones de datos y estimaciones por partículas. Esto hace al método mas robusto y paralelizable. Así mismo, como es mencionado en [37] ofrece una alternativa adaptable a diferentes casos de uso y evita los cálculos matriciales complejos que se presentan en propuestas como el uso de Unscented Kalman Filters (UKF) para resolver el problema de SLAM[37].

Por otro lado, dada la naturaleza continua de un mapa, se requiere algún mecanismo para discretizar el entorno a ser estimado. Por ello se toma como base un mapa vacío de un número de celdas o rejillas finito (del inglés *grid*) que corresponden a un estado del sistema bien definido en el espacio. En cada movimiento del robot, se actualiza cada una de las celdas en función de los controles de movimiento y lecturas de los sensores del vehículo.

Para poder adaptar el algoritmo FastSLAM a una estimación con base a celdas, es suficiente con cumplir con tres pasos mostrados en el siguiente pseudo-código:

```

fastSLAM_OccupancyGrid( $\chi_{t-1}, u_t, z_t$ ):
  for  $k=1$  to  $M$  do
     $x_t^{[k]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[k]})$ 
     $\omega_t^{[k]} = \text{measurement\_model\_map}(z_t, x_t^{[k]}, m_{t-1}^{[k]})$ 
     $m_t^{[k]} = \text{update\_occupancy\_grid}(z_t, x_t^{[k]}, m_{t-1}^{[k]})$ 
     $\bar{\chi}_t = \bar{\chi}_t + \langle x_t^{[k]}, m_t^{[k]}, \omega_t^{[k]} \rangle$ 
  endfor
  for  $k=1$  to  $M$  do
    draw  $i$  with probability  $\propto \omega_t^{[k]}$ 
    add  $\langle x_t^{[k]}, m_t^{[k]} \rangle$  to  $\chi_t$ 
  endfor
  return  $\chi_t$ 

```

donde χ_t corresponde a un grupo de partículas en un tiempo dado, χ_{t-1} el mismo grupo en un tiempo previo, M el tamaño del conjunto de partículas, k el índice de una partícula en el grupo y m_t el mapa en el tiempo t .

La función *sample_motion_model* realiza un muestreo de un modelo de movimiento, *measurement_model_map* realiza una estimación del mapa para cada partícula dada las lecturas obtenidas de los sensores y *update_occupancy_grid* calcula y actualiza los pesos de importancia por cada partícula actualizando el estado de cada celda en función de la posición estimada y las mediciones obtenidas en los dos procesos anteriores. La última sección del pseudo-código ejemplifica el re-muestreo en función de los pesos actualizados.

1.6.1. Modelo odométrico de movimiento

El modelo odométrico de movimiento en inglés (Odometry motion model) emplea las mediciones odométricas como base para estimar los cambios de posición de un robot a lo largo del tiempo, donde una de sus características es que estima el movimiento en retrospectiva.

Con el modelo propuesto, la trayectoria recorrida por un robot en un intervalo de tiempo dado después de una acción de movimiento se modela mediante tres estados[37]: un giro inicial, una traslación y una segunda rotación. Cada uno de estos movimientos se asumen están sujetos a ruido. La trayectoria propuesta se muestra en la Figura 1.4.

El pseudo-código del modelo propuesto está dado por:

```

Motion_Model_Odometry( $x_t, u_t, x_{t-1}$ ):

```

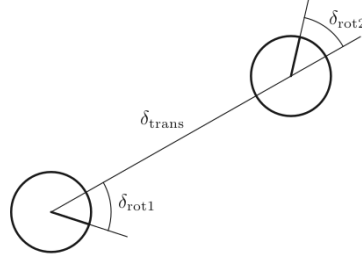


Figura 1.4: Forma de la trayectoria propuesta en el modelo odométrico de movimiento[37].

$$\begin{aligned}
 \delta_{rot1} &= \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta} \\
 \delta_{trans} &= \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2} \\
 \delta_{rot2} &= \bar{\theta}' - \bar{\theta} - \delta_{rot1} \\
 \hat{\delta}_{rot1} &= \delta_{rot1} - \text{sample}(\alpha_1 \hat{\delta}_{rot1}^2 + \alpha_2 \hat{\delta}_{trans}^2) \\
 \hat{\delta}_{trans} &= \delta_{trans} - \text{sample}(\alpha_3 \hat{\delta}_{trans}^2 + \alpha_4 \hat{\delta}_{rot1}^2 + \alpha_4 \hat{\delta}_{rot2}^2) \\
 \hat{\delta}_{rot2} &= \delta_{rot2} - \text{sample}(\hat{\delta}_{rot2}, \alpha_1 \hat{\delta}_{rot2}^2 + \alpha_2 \hat{\delta}_{trans}^2) \\
 x' &= x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1}) \\
 y' &= y + \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1}) \\
 \theta' &= \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \\
 \text{return } x'_t &= (x', y', \theta')
 \end{aligned}$$

Donde el control u_t está dado por $(\bar{x}_{t-1}, \bar{x}_t)$, con $\bar{x}_{t-1} = (\bar{x}, \bar{y}, \bar{\theta})$ y $\bar{x}_t = (\bar{x}', \bar{y}', \bar{\theta}')$

1.6.2. Percepción del robot

Los modelos de medición del entorno son el segundo paso empleado en el proceso de estimación y mapeo propuesto. Estos métodos proporcionan una estimación del entorno en función de las mediciones de rango obtenidas por los sensores a bordo del robot y de la posición previamente estimada, a su vez que toma en cuenta el ruido al cual están sometidos los sensores. Estos sensores suelen estar basados en visión, luz o sonido. Durante el presente trabajo se emplearon sensores ultrasónicos y se propuso un sistema basado en sonido de alta frecuencia descrito en el Capítulo 4.

El modelo de SLAM propuesto incluye cuatro tipos de errores a los cuales están sometidos los sensores de rango:

- Correcta estimación de rango con ruido local de medición, modelado por una distribución gaussiana (ρ_{hit}).
- Objetos inesperados en el mapa, que corresponden a objetos que posiblemente se encuentran en movimiento y pueden generar errores de rango fortuitos. Está modelado por una distribución exponencial (ρ_{short}).
- Fallos en medición. Por lo regular corresponden al rango máximo que es obtenido por el sensor cuando ocurre algún error en la estimación. Es modelado mediante una distribución punto-masa centrada en el rango máximo que puede ser medido por el sensor de rango (ρ_{max}).
- Mediciones aleatorias. Son mediciones obtenidas por razones inexplicables y es modelada mediante una distribución uniforme distribuida a lo largo de todo el rango del sensor (0 al rango máximo obtenible por el sensor) (ρ_{rand}).

Estas cuatro distribuciones son mezcladas mediante una media ponderada, definida por los parámetros z_{hit} , z_{short} , z_{max} , z_{rand} con $z_{hit} + z_{short} + z_{max} + z_{rand} = 1$. Así pues, la k -ésima medición z en un conjunto de mediciones de tamaño K , en un tiempo t está definida por la probabilidad condicional:

$$p(z_t^k | x_t, m) = \begin{bmatrix} z_{hit} \\ z_{short} \\ z_{max} \\ z_{rand} \end{bmatrix}^T \cdot \begin{bmatrix} \rho_{hit}(z_t^k | x_t, m) \\ \rho_{short}(z_t^k | x_t, m) \\ \rho_{max}(z_t^k | x_t, m) \\ \rho_{rand}(z_t^k | x_t, m) \end{bmatrix}$$

La Figura 1.5 muestra como se ve la distribución de error propuesta.

1.6.3. Campos de probabilidad para sensores de rango

Una vez definida la forma del error supuesto en las mediciones del robot, se toma el método que estima el entorno del vehículo en función de su posición y las lecturas de sus sensores.

El algoritmo de campos de probabilidad para sensores de rango (del inglés *Likelihood Field Range Finder Model*) consiste en proyectar los puntos detectados por un sensor de rango en las coordenadas globales de un mapa. Para esto, es necesario conocer la relación entre las coordenadas relativas dentro del marco del robot a coordenadas globales, permitiendo conocer donde el rayo del sensor se origina y hacia donde finalmente apunta. Empleando esta metodología se puede calcular la probabilidad

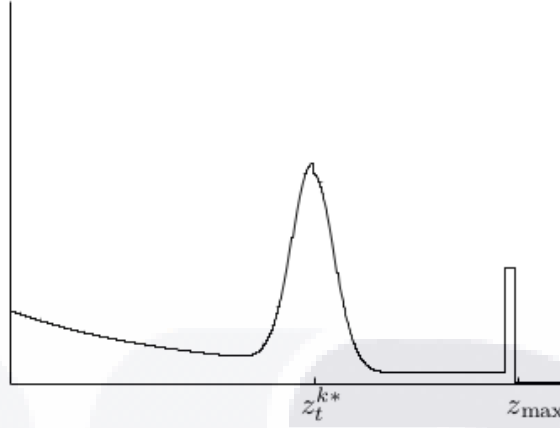


Figura 1.5: Distribución del error propuesto para el método FastSLAM[37] para un sensor de distancia.

de obtener una medición dada en función de una posición previamente estimada. El pseudo-código está dado por:

```

Likelihood_Field_Range_Finder_Model( $z_t, x_t, m$ ):
     $q = 1$ 
    for all  $k$  do
        if  $z_t^k \neq z_{max}$ 
             $x_{z_t^k} = x + x_{k,sens} \cos \theta - y_{k,sens} \sin \theta + z_t^k \cos(\theta + \theta_{k,sens})$ 
             $y_{z_t^k} = y + y_{k,sens} \cos \theta + x_{k,sens} \sin \theta + z_t^k \sin(\theta + \theta_{k,sens})$ 
             $dist = \min_{x', y'} \left\{ \sqrt{(x_{z_t^k} - x')^2 + (y_{z_t^k} - y')^2} \mid \langle x', y' \rangle \text{ occupied in } m \right\}$ 
             $q = q \cdot (z_{hit} \cdot \text{prob}(dist, \sigma_{hit}) + \frac{z_{random}}{z_{max}})$ 
    return  $q$ 

```

donde la función **prob**($dist, \sigma_{hit}$) calcula la probabilidad de $dist$ bajo una distribución gaussiana centrada en cero y una desviación estándar de σ_{hit} y emplea una distancia euclidiana para encontrar el vecino mas cercano.

1.6.4. Occupancy grid mapping

La finalidad de el algoritmo de estimación de ocupación de mapa en reja es calcular una distribución a posteriori a lo largo de un mapa discretizado (rejas o celdas) dado un conjunto de datos:

$$p(m|z_{1:t}, x_{1:t})$$

donde m es el mapa, que puede ser definido como un conjunto de celdas $\mathbf{m}_i$, $z_{1:t}$ el conjunto de mediciones hasta un tiempo t y $x_{1:t}$ la trayectoria recorrida por un robot descrito en forma de poses $\langle x_t, y_t, \theta_t \rangle$. Así mismo, a cada celda $\mathbf{m}_i$ se le puede adjudicar un valor que indica si la celda está ocupada (1) o no (0). Con esto, puede descomponerse el problema completo de la estimación de distribución del mapa, en un conjunto de problemas binarios de estado estático, que corresponden a cada una de las celdas. Así pues, la distribución total del mapa puede ser aproximada a través del producto de sus marginales con base a cada una de las celdas y su relación con sus vecinos:

$$p(m|z_{1:t}, x_{1:t}) = \prod_i p(\mathbf{m}_i|z_{1:t}, x_{1:t})$$

El algoritmo para la actualización de la distribución de probabilidades de un mapa discretizado en reja y basado en un filtro de Bayes está dado como:

```

Occupancy_Grid_Mapping( $l_{t-1,i}, x_t, z_t$ ):
  for all cells  $\mathbf{m}_i$  do
    if  $\mathbf{m}_i$  in perceptual field of  $z_t$  then
       $l_{t,i} = l_{t-1,i} + \text{inverse sensor model}(\mathbf{m}_i, x_t, z_t) - l_0$ 
    else
       $l_{t,i} = l_{t-1,i}$ 
    endif
  endfor
  return  $l_{t,i}$ 

```

donde el $l_{t,i}$ es la función *logit* para representar la ocupación de una celda definida como:

$$l_{t,i} = \log \frac{p(\mathbf{m}_i|z_{1:t}, x_{1:t})}{1 - p(\mathbf{m}_i|z_{1:t}, x_{1:t})}$$

y la probabilidad de cada celda es obtenida a como:

$$p(\mathbf{m}_i|z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + \exp \{l_{t,i}\}}$$

El algoritmo definido en el pseudo-código *Occupancy_Grid_Mapping* itera sobre cada una de las celdas i del mapa m , y actualiza aquellas que caen dentro del cono de la medición z_t del sensor de rango mostrado en la Figura 1.6.

El pseudo-código *Inverse_Range_Sensor_Model* calcula el modelo inverso determinando el índice del rayo de medición k y el rango r para el centro de masa de la celda $\mathbf{m}_i$. El pseudo-código es:

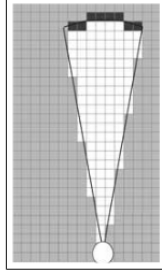


Figura 1.6: Modelo del cono de medición de un sensor de rango[37]. El cono muestra tres zonas importantes: en gris celdas del mapa que no se tiene suficiente certeza que están ocupadas, celdas en claro que representa celdas libres de obstáculos y celdas oscuras que indican con mayor certeza que están ocupadas.

```

Inverse_Range_Sensor_Model( $m_i, x_t, z_t$ ):
 $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$ 
 $\phi = \arctan 2(y_i - y, x_i - x) - \theta$ 
 $k = \arg \min_j |\phi - \theta_{j,sens}|$ 
if  $r > \min(z_{max}, z_t^k + \alpha/2)$  or  $|\phi - \theta_{k,sens}| > \beta/2$  then
    return  $l_0$ 
if  $z_t^k < z_{max}$  and  $|r - z_t^k| < \alpha/2$ 
    return  $l_{occ}$ 
if  $r \leq z_t^k$ 
    return  $l_{free}$ 
endif

```

donde l_0 es el *logit* para representar la ocupación incierta de una celda, l_{occ} es el *logit* para representar la ocupación de una celda, l_{free} es el *logit* para representar una celda libre, el ancho de la región del cono del sensor está controlado por α y el ángulo de apertura por β .

Dado el contexto del proyecto de investigación tratado, los capítulos siguientes 2, 3 y 4 muestran los artículos publicados que fueron resultado del desarrollo del tema de investigación.



Capítulo 2

Algoritmo para el reconocimiento automático de acordes musicales usando plantillas basadas en la media geométrica

El objetivo principal de este trabajo de tesis es explorar y aplicar técnicas de eco-localización en robots. Una parte crucial de este proyecto de investigación es la capacidad de detectar frecuencias específicas dentro de una señal de audio, así como identificar sus características fundamentales. Este proceso no solo es esencial para la eco-localización, sino que también tiene aplicaciones en el campo de la acústica y el procesamiento de señales.

Para ilustrar la efectividad y precisión de las herramientas de análisis de señales, se propone como caso de estudio la aplicación de estas técnicas en la identificación de acordes dentro de un archivo de audio. La detección de acordes implica el reconocimiento de combinaciones de frecuencias sonoras que corresponden a notas musicales simultáneas. Esta tarea requiere un análisis preciso de las frecuencias presentes en la señal de audio, lo que demuestra la capacidad de las herramientas de análisis utilizadas para manejar señales complejas en condiciones reales. Además, la identificación de acordes no solo resalta la aplicabilidad del algoritmo en el ámbito musical, sino que también establece una base para la utilización de la eco-localización en sistemas robóticos, donde la identificación precisa de señales acústicas específicas es crucial para tareas de navegación y detección en entornos no estructurados.

2.1. Marco teórico

El procesamiento digital de audio en robótica es fundamental para la interacción de un agente robótico y su entorno. Los robots pueden usar el sonido para detectar obstáculos, identificar fuentes de sonido, localizar objetos en un entorno ruidoso o limitados en visión. Diferentes herramientas matemáticas como la transformada de Fourier o filtros digitales son empleados dentro de aplicaciones de localización de sonido o detección de anomalías.

2.1.1. Análisis en frecuencia

Tanto en señales periódicas como no periódicas, es posible realizar un análisis frecuencial. Éste consiste en descomponer la señal en componentes sinusoidales y determinar su distribución en el dominio de la frecuencia. Entre las herramientas más importantes para este análisis destaca la transformada de Fourier, que permite obtener el espectro de la señal, representando su energía en función de la frecuencia.

La transformada de Fourier

La transformada de Fourier parte del supuesto de que una señal no periódica puede considerarse como una señal periódica con un periodo infinito. En este caso, los componentes armónicos de la serie de Fourier se distribuyen de manera continua.

Definición 2.1.1 Sea $x(t)$ una función continua, no periódica e integrable. Su transformada de Fourier está definida como:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt$$

La transformada inversa se define como:

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft} df$$

La transformada discreta de Fourier

Considerando una secuencia finita $x[n]$ de N muestras en el dominio del tiempo, donde $n = 0, 1, 2, \dots, N - 1$, se tienen las siguientes definiciones:

Definición 2.1.2 Sea $x[n]$ una secuencia discreta. Su transformada discreta de Fourier está definida como:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, 2, \dots, N - 1$$

y su inversa se define como:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{-j \frac{2\pi}{N} kn}, \quad n = 0, 1, 2, \dots, N-1$$

La transformada rápida de Fourier

La transformada discreta de Fourier es una importante herramienta en el procesamiento y análisis de señales digitales. Sin embargo, el cálculo directo de la transformada siguiendo la definición 2.1.2 implica una complejidad computacional $\mathcal{O}(N^2)$ lo cual es costoso para señales largas. La transformada rápida de Fourier basada en el algoritmo de Cooley–Tukey, (en inglés Fast Fourier Transform FFT), optimiza el cálculo siguiendo un paradigma de 'divide y vencerás' recursivo, reduciendo la complejidad a $\mathcal{O}(N \log N)$, sustentado en la propiedad de simetría y periodicidad de la DFT.

Filtros

Un filtro se define como un sistema o dispositivo diseñado para modificar una señal de entrada mediante la selección de ciertas características. Los filtros se clasifican en función de su respuesta como:

- **Filtros lineales:** Sistemas cuya respuesta es proporcional a la señal de entrada.
- **Filtros no lineales:** Sistemas cuya respuesta no es proporcional a la señal de entrada.

Teorema de la convolución

La convolución es una operación matemática que transforma dos funciones $f(x)$ y $g(x)$ en una tercera función que consiste en la superposición de una versión trasladada e invertida de la segunda sobre la primera. La convolución en tiempo discreto se define como:

Definición 2.1.3 Sean dos series de valores discretos $f[x]$ y $g[x]$, se define la convolución discreta como:

$$f[x] * g[x] = \sum_{n=0}^{N-1} f[n] g[x - n]$$

donde $g[x - n]$ es un desplazamiento de la señal g en n muestras.

Una propiedad importante de la convolución consiste en que la transformada de Fourier de la convolución de dos funciones es igual al producto de las transformadas de cada una de ellas. Así pues puede interpretarse como que el proceso de filtrado, con una mayoría de filtros lineales, es equivalente a multiplicar el espectro en frecuencia de la señal de entrada por la respuesta en frecuencia del filtro. Si f y g son funciones integrables, la convolución también es integrable, por lo que son válidas las siguientes igualdades:

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g)$$

$$\mathcal{F}(f \cdot g) = \mathcal{F}(f) * \mathcal{F}(g)$$

Aquí $\mathcal{F}$ es un operador que asocia a cada señal con su transformada de Fourier.

Filtros de Gabor

Los filtros de Gabor son filtros lineales compuestos por una función sinusoidal o cosenoidal dentro de un marco de una ventana temporal. Dada su composición, posee un comportamiento en frecuencia muy selectivo. Los filtros de Gabor son a su vez un filtro complejo en el tiempo, cuya parte real e imaginaria se definen como:

$$g_e(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \cos(2\pi\omega_0 x), \quad (2.1)$$

$$g_o(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \sin(2\pi\omega_0 x). \quad (2.2)$$

Aquí, $g_e(x)$ es un filtro par que corresponde a la parte real del filtro completo, $g_o(x)$ un filtro impar correspondiente a la parte imaginaria, ω_0 la frecuencia central del filtro, y σ corresponde a la dispersión de la ventana gaussiana que delimita al filtro. La paridad de un filtro es definida según la simetría de la respuesta al impulso del filtro con respecto a su centro.

La respuesta en frecuencia de los filtros de Gabor son representadas por una campana gaussiana centrada en la frecuencia de su función sinusoidal. Esta respuesta en frecuencia es modelada como:

$$G(f) = \exp\left(\frac{-(f - f_0)^2}{2\sigma^2}\right). \quad (2.3)$$

Así mismo, otra forma de emplear el filtro es modificando su comportamiento en frecuencia a lo largo del espectro de manera logarítmica. Esta respuesta es modelada como:

$$G(f) = \exp\left(\frac{-(\log(f/f_0))^2}{2\sigma^2}\right). \quad (2.4)$$

Los filtros de Gabor dada su composición, su uso dentro del procesamiento de audio otorga diferentes ventajas:

- Los filtros de Gabor dadas sus ventanas de tiempo específicas, permiten analizar señales de audio capturando eventos sonoros transitorios o cambios rápidos en la señal.
- Dada su selectividad en frecuencia, son ideales para extraer características relevantes de la señal de audio, como patrones espectrales o tonos formantes.
- Se adaptan a diferentes escalas y frecuencias fácilmente, lo cual los hace útiles para procesar diferentes tipos de señales de audio.

Uno de los principales temas a tratar en el procesamiento digital de audio y más en específico en el procesamiento digital de la música, consiste en desarrollar sistemas de transcripción y análisis musical, lo cual implica convertir un audio musical a su correspondiente notación. Esto puede ser empleado en la enseñanza[4], la creación de bases de datos musicales[32], sistemas autónomos de acompañamiento e improvisación[29] o en la investigación de la música misma[25]. Un nicho de investigación en el desarrollo de sistemas de procesamiento musical es diseñar un algoritmo de análisis de los sonidos generados por un instrumento, específicamente acordes ejecutados en él, a fin de detectarlos automáticamente partiendo de las características de la señal de audio. El problema ha sido abordado por Müller [24] dónde propone dos caminos para la detección de acordes en un audio, mediante plantillas de los acordes a reconocer o utilizando algoritmos clasificadores, proponiendo en su libro cadenas de Markov. Como se mencionó anteriormente, los filtros de Gabor son ideales para extraer características relevantes de la señal de audio como patrones frecuenciales y transitorios temporales, por lo que su uso en esta clase de problemas puede ser justificable.

2.1.2. Teoría musical

La música se define como el arte de estructurar sonidos con el propósito de evocar una determinada emoción en el oyente[14]. Este arte se compone de tres elementos fundamentales: ritmo, melodía y armonía. El ritmo se refiere a la organización temporal de los sonidos y silencios dentro de una composición musical. La melodía consiste en una secuencia de notas dispuestas a lo largo del tiempo, mientras que la armonía se configura mediante la combinación simultánea de varias notas. Estos elementos están regidos por principios bien establecidos que garantizan la coherencia musical.

En la tradición musical occidental, las notas se organizan siguiendo una escala logarítmica de 12 notas, que se repiten cíclicamente en intervalos denominados octavas. Uno de los primeros sistemas para la construcción de escalas musicales fue propuesto por los pitagóricos, quienes establecieron divisiones geométricas de una cuerda en fracciones de 2, 3 y 4 partes iguales. Este sistema prevaleció hasta finales de la Edad Media. No obstante, en el siglo XVII, J.S. Bach introdujo modificaciones significativas en la escala musical previamente utilizada, dando origen a la *escala temperada*, que se ha convertido en el estándar en la música occidental[39].

Dentro de la escala temperada, entre una nota y su octava superior existen 11 divisiones intermedias en términos de frecuencia. Así, las 12 notas que conforman esta escala corresponden a C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯ y B. La unidad mínima de separación entre estas notas se denomina semitono, cuya relación de frecuencia se define por:

$$f_{n+1}/f_n = k$$

Donde f_{n+1} y f_n representan las frecuencias de dos notas consecutivas separadas por n semitonos, y la constante k , según la afinación moderna, equivale a la razón de frecuencia de un semitono, la cual se expresa como $\sqrt[12]{2} \approx 1.0595$. Regularmente, se especifica la octava a la cual una nota pertenece como un subíndice.

Por convención internacional, la Organización Internacional de Normalización ha definido la nota A_4 , con una frecuencia de 440 Hz, como el estándar de referencia para la afinación en la música occidental. A partir de este punto de referencia, cualquier nota de la escala temperada se encuentra separada del A_4 por un número entero de semitonos, denotado por i . Si una nota se encuentra por encima (notas subsecuentes en la escala de manera ascendente) de A_4 , i toma valores positivos; en caso contrario, adquiere valores negativos. De esta manera, la relación logarítmica entre las notas permite expresar su frecuencia fundamental nominal mediante la ecuación:

$$f_i = 440 \times 2^{\frac{i}{12}} \text{Hz}, \quad i \in \mathbb{Z} \quad (2.5)$$

Un acorde se define como la combinación simultánea de tres o más notas, independientemente de la octava en la que se encuentren. La tríada constituye la unidad armónica fundamental y se construye mediante la superposición de un intervalo de tercera y uno de quinta respecto a la nota fundamental, que da nombre al acorde[14].

Los acordes principales utilizados en la música tonal derivan de la armonización de la escala mayor, basada en la superposición de dos terceras diatónicas consecutivas (notas pertenecientes a la escala). En este contexto, sobre cada nota de la escala

mayor es posible construir una tríada.

Según la cualidad de los intervalos formados entre la nota fundamental y sus respectivas tercera y quinta, se identifican tres tipos principales de tríadas derivadas de la armonización de la escala mayor:

- **Acorde mayor:** Formado por la tónica, una tercera mayor y una quinta justa. Un ejemplo es el acorde de C mayor, compuesto por las notas C, E (tercera mayor) y G (quinta justa).
- **Acorde menor:** Compuesto por la tónica, una tercera menor y una quinta justa. Por ejemplo, el acorde de C menor está constituido por las notas C, D $\flat$ (tercera menor) y G (quinta justa).
- **Acorde disminuido:** Integrado por la tónica, una tercera menor y una quinta disminuida. Un ejemplo representativo es el acorde de C disminuido, conformado por C, D $\flat$ (tercera menor) y F $\flat$ (quinta disminuida).
- **Acorde suspendidos:** Formado por la tónica, una segunda mayor (acordes sus2) o una cuarta justa (sus4) y una quinta justa. Por ejemplo, el acorde C sus2 se compone de las notas C, D (segunda mayor) y G. El acorde C sus4 se forma con las notas C, F (cuarta justa) y G.

2.2. Metodología

Así pues, el objetivo principal es desarrollar un algoritmo de análisis de audio basado en filtros de Gabor que permita analizar cambios transitorios en el tiempo de la señal y poder reconocer patrones frecuenciales. Como caso de aplicación del modelo propuesto, se desarrolla un sistema capaz de detectar los acordes presentes en una pieza musical de manera precisa y robusta a través de herramientas matemático-estadísticas utilizando plantillas para la detección.

El algoritmo propuesto se encamina a la detección automática de los acordes musicales presentes en una grabación de audio de un único instrumento. Los acordes a detectar son acordes mayores, menores, aumentados, disminuidos y suspendidos tanto de segunda mayor como de cuarta justa. La Figura 2.1 muestra un diagrama del método.

Todas las etapas del método se agrupan en tres grandes secciones: El preprocesamiento, la detección de acordes y el postprocesamiento.

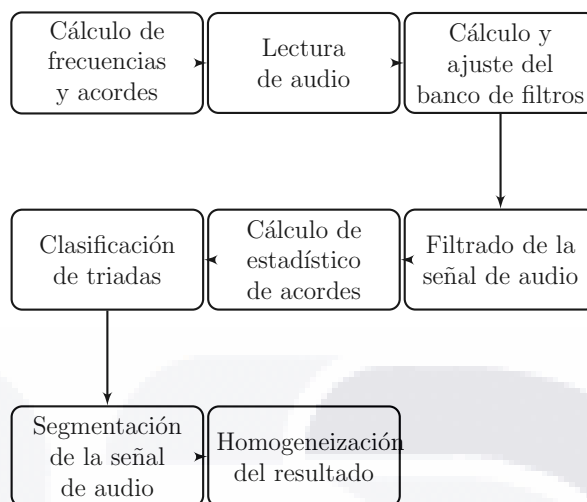


Figura 2.1: Modelo del algoritmo propuesto

2.2.1. Preprocesamiento

En esta sección se preparan tanto la señal de audio que será procesada como las estructuras que necesita el algoritmo internamente para continuar con la clasificación. Como paso inicial de la sección, se crean las plantillas para cada acorde posible, mayor, menor, disminuido, aumentado, suspendido 2 y suspendido 4 sobre cada una de las notas cromáticas. Posteriormente, se calculan las frecuencias teóricas de cada una de las notas musicales. Estas se generan dentro de un rango de octavas por cada nota que puede ser cambiado. Con esto se obtiene una lista en la que se incluye cada nota cromática (C, C $\sharp$, D, D $\sharp$, E, F, G, G $\sharp$, A, A $\sharp$ y B), y sus frecuencias sobre el grupo especificado de octavas.

Una vez generadas las plantillas y calculadas las frecuencias teóricas de las notas, se toma el archivo de audio en formato WAV a clasificar. A fin de disminuir la cantidad de datos a procesar se realiza un submuestreo de la señal de audio a 11025 muestras por segundo. En caso de tener una señal estéreo, los canales izquierdo y derecho son sumados en una sola señal. Con el propósito de resolver los efectos de borde se agrega medio segundo al principio y al final de la señal. Este es eliminado al final del proceso de clasificación.

Ya construidas las estructuras anteriores, se calcula la transformada rápida de Fourier de la señal de audio. Se genera un banco de filtros de Gabor a partir del modelo en frecuencia en su versión logarítmica representado en la Ecuación (2.4) donde f_0 corresponde a la frecuencia central del filtro y σ es el ancho del mismo o

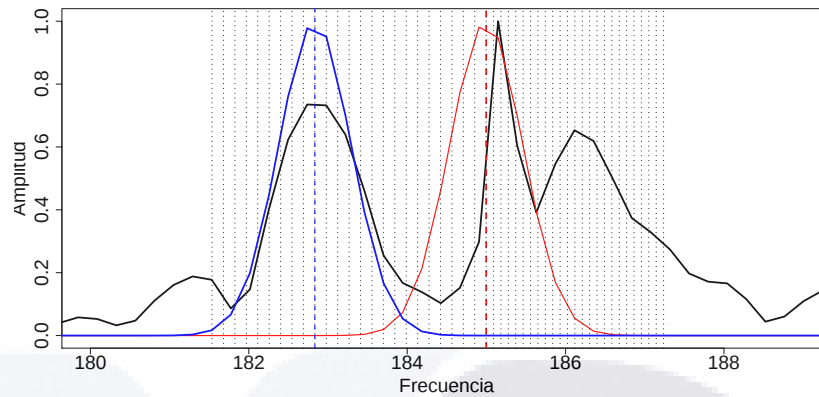


Figura 2.2: Filtros de Gabor correspondientes a una nota en particular antes y después del ajuste con respecto a la frecuencia nominal. Las líneas verticales punteadas corresponden a los pasos posibles dentro del proceso de ajuste, mientras que las dos líneas verticales de guiones corresponde a las frecuencias centrales del filtro original y ajustado.

banda de paso.

Al inicio, cada filtro del banco es centrado en cada una de las frecuencias de las notas musicales calculadas. Sin embargo, a fin de sopesar las ligeras desafinaciones que puedan presentarse en el audio, se realiza un proceso para ajustar las frecuencias centrales de cada uno de los filtros. Estas son adaptadas de acuerdo a los picos más cercanos a la frecuencia central nominal de las notas dentro del espectro de la señal de audio. Esto se realiza con un rango de movimiento en función de la frecuencia sobre la cual se encuentre. Debido a que las notas más bajas se encuentran más cerca entre sí, la libertad de ajuste es menor que en las frecuencias más altas, las cuales están más separadas y se corre menos riesgo de un traslape de los filtros. El rango de ajuste de cada filtro se obtiene, en primera instancia, calculando las diferencias entre la frecuencia central del filtro actual y la frecuencia de la nota anterior y de la nota siguiente. Estas dos diferencias son divididas en tres partes iguales y se toma el rango de la tercera parte más cercana a la frecuencia del filtro a ajustar, o de igual manera, se toma la última tercera parte de la primera diferencia y la primera tercera parte de la segunda diferencia. Así pues, estos dos intervalos son divididos a su vez en 25 partes iguales, las cuales, cada una de esas divisiones serán los pasos de ajuste que puede dar cada uno de los filtros. La Figura 2.2 muestra un ejemplo de un filtro en frecuencia antes y después de su ajuste.

El valor de σ del modelo se selecciona a prueba y error de acuerdo a los resultados que arrojaron los experimentos.

Al tener calculados los pasos de ajuste, cada filtro se coloca en un ciclo sobre

su frecuencia central con un movimiento hacia atrás o hacia adelante en frecuencia con cada paso calculado y se obtiene su respuesta. Debido a que los filtros deben posicionarse sobre el pico más cercano y para evitar en este punto contribuciones erróneas debido a la dispersión de energía en frecuencias vecinas, las cuales se localizan a niveles bajos de la señal en amplitud, se calcula como punto de corte el 80 % de la curva gaussiana del filtro y se obtiene la contribución del filtro en ese punto. Así pues, la frecuencia central de cada filtro se ajusta a aquella en donde se obtiene una mayor contribución.

Una vez generado el banco de filtros, se filtra la señal de audio. El proceso de filtrado de una señal (en particular nuestro filtrado debido a los filtros lineales empleados) se justifica desde el punto de vista matemático con una convolución. Así pues, por el teorema de la convolución, se obtiene la señal de audio filtrada por cada uno de los filtros de Gabor multiplicando el espectro de la señal por la respuesta en frecuencia del propio filtro que es el modelo elegido para construirlo. Posteriormente se realiza la transformada inversa de Fourier sobre la señal filtrada y se obtiene su módulo. Esto proporciona el comportamiento de cada nota a lo largo del tiempo dentro de la grabación.

Al tiempo que se va filtrando la señal con cada uno de los filtros del banco, se van acumulando las señales filtradas que pertenecen a la misma clase tonal, o de otra manera, se van sumando la señales que pertenecen a la misma nota musical sin importar la octava sobre la cual se encuentre. El proceso de filtrado genera una matriz de $12 \times N$ donde N es el número de muestras de la señal completa submuestreada y 12 filas que corresponden a las 12 notas cromáticas o clases tonales. A esta matriz se le conoce como *chroma*[24].

2.2.2. Detección de acordes

Una vez obtenida la matriz *chroma*, se analizan las contribuciones de cada una de las notas cromáticas en cada punto temporal de la señal. A partir de esto se busca la posible presencia de algún acorde en función de dichas contribuciones de las notas que forman al acorde mismo.

Así pues, a partir de las plantillas generadas previamente, se tienen grupos de tres notas que corresponden a cada uno de los acordes posibles sobre cada nota, 12 acordes mayores y 12 menores, 12 disminuidos, 12 aumentados, 12 sus2 y 12 sus4. A fin de obtener un estadístico que caracterice la contribución de cada triada de notas para cada acorde, se calcula una media geométrica de la amplitud de las tres notas que conforman el acorde en la señal en cada punto temporal. Ésta nos permite tener un valor que englobe a las tres notas de manera equitativa y que presenta

menos sensibilidad a valores atípicos[35]. Estos valores pueden ser causados por cambios de la dinámica de ejecución del instrumento grabado, por una mala técnica de interpretación o por la misma evolución de la señal a lo largo del tiempo.

Una vez obtenido el estadístico para cada acorde posible, se asigna la muestra a la clase o acorde que obtuvo la mayor contribución. Después de esta etapa, toda la señal de audio queda clasificada a lo largo del tiempo.

2.2.3. Postprocesamiento

A fin de aumentar la homogeneidad de los acordes detectados por el algoritmo, después del proceso de asignación, se homogeneiza por bloques el resultado. Esto se realiza cambiando el acorde asignado en diferentes puntos dentro de la señal que son clasificados erróneamente en función de los acordes detectados en las muestras vecinas. En primera instancia, se segmenta la señal de audio clasificada para posteriormente homogeneizar el resultado de la clasificación.

Segmentación de la señal de audio

La segmentación de la señal consiste en un procedimiento donde se trata de identificar los puntos en el tiempo dónde ocurrió un cambio de acorde sin importar a que acordes corresponde. Para esto es necesario identificar cambios en la distribución de energía en las frecuencias de las notas. Así pues, a partir de la matriz chroma se calcula una media aritmética de la respuesta de todas las notas por cada muestra o punto en el tiempo de la señal. Con esto se obtiene un vector con el estadístico del comportamiento en amplitud de las frecuencias de las notas a lo largo del tiempo. A fin de detectar cambios significativos en la frecuencia, lo cual implicaría un cambio de acorde y de la dinámica de ejecución, se obtiene la derivada del vector obtenido anteriormente y se buscan los puntos críticos en la derivada. Luego se obtiene el signo de toda la derivada y se detectan los cambios en ella. Una vez obtenido el vector de puntos críticos a lo largo de la señal, se obtienen aquellos que indiquen un punto de inflexión. En particular, son de mayor interés los cambios en valor de más a menos, por lo que se recorre el vector de promedios en función de los puntos críticos y se encuentran aquellos dónde el cambio establecido es detectado. Estos puntos indican los puntos donde muy probablemente ocurrió un cambio de acorde.

Homogeneización del resultado

Una vez obtenidos los puntos de cambio de los acordes en la señal de audio y a su vez la señal clasificada en acordes, se calculan los bloques que corresponden a un mismo acorde según los puntos de cambio. Una vez establecidos, se obtiene la moda

de los acordes detectados en cada bloque y se establece como acorde de la totalidad del mismo aquél que cuente con el mayor conteo. Este procedimiento contribuye a limpiar el resultado de clasificaciones erróneas o extremas dentro de los bloques que ya son considerados como un sólo acorde.

2.3. Experimentos

Se llevan acabo tres experimentos tratando de abordar diferentes conceptos del algoritmo propuesto. Para realizar las pruebas se utilizan grabaciones con una guitarra acústica Fender New Porter Player de cuerdas de metal, una guitarra clásica Yamaha C40 de cuerdas de nylon, un teclado Yamaha PSR E223 con su timbre ajustado como piano y un ukelele La Sevilla Svuke-200. Las grabaciones se realizan a través de una interfaz Behringer U-PHORIA UMC202HD conectado directamente para el caso de la guitarra acústica y el teclado, y a través de un micrófono Shure SV100 para la guitarra clásica y el ukelele.

2.3.1. Experimento 1

Como primer experimento se proponen cuatro progresiones de seis acordes diferentes ejecutadas en la guitarra acústica de cuerdas de metal. La duración de cada progresión es de 22 segundos aproximadamente. Se trata de abarcar todos los tipos de acordes propuestos a detectar. Los acordes elegidos se seleccionan de forma aleatoria sin ninguna relación armónica entre sí. Las cuatro progresiones propuestas son:

- Secuencia A: C Cm E Edim Dsus4 C[♯]aug
- Secuencia B: Am Baug Esus4 Dsus4 A[♯]m
- Secuencia C: F[♯] Gm Gdim D[♯] Bm Bsus2
- Secuencia D: Csus4 Eaug G Dbdim F Em

Las Figuras 2.3, 2.4, 2.5 y 2.6 muestran los resultados de la detección de los acordes propuestos para cada secuencia. El eje y de la gráfica indica la nota tónica correspondiente al acorde detectado mientras que el eje x corresponde a la duración. Cada tipo de acorde se representa por un color diferente.

Se observa en general que los acordes detectados por el algoritmo son los presentes en las progresiones propuestas. La duración de cada uno de los acordes dentro de las progresiones son aproximadamente la misma, lo cual nos permite observar que

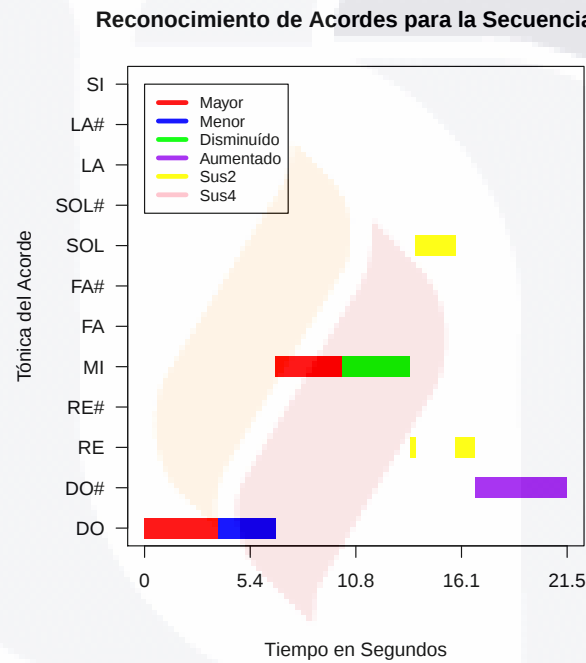
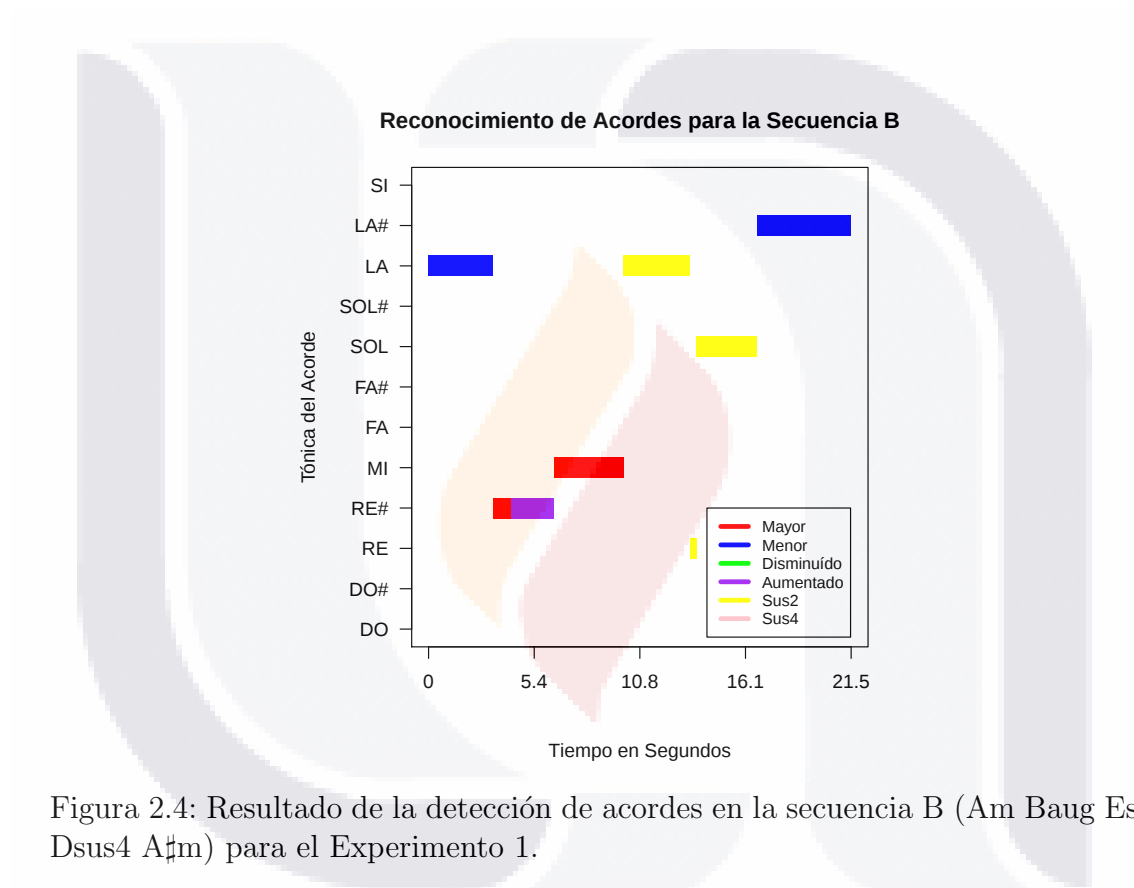


Figura 2.3: Resultado de la detección de acordes en la secuencia A (C Cm E Edim Dsus4 C#aug) para el Experimento 1.



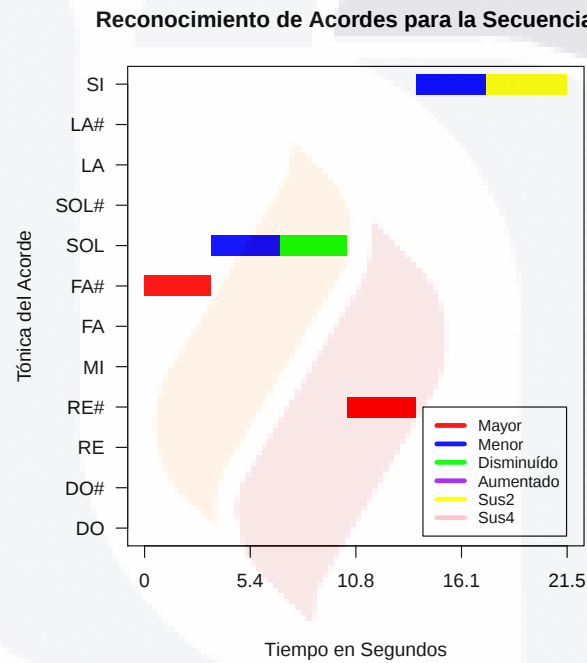
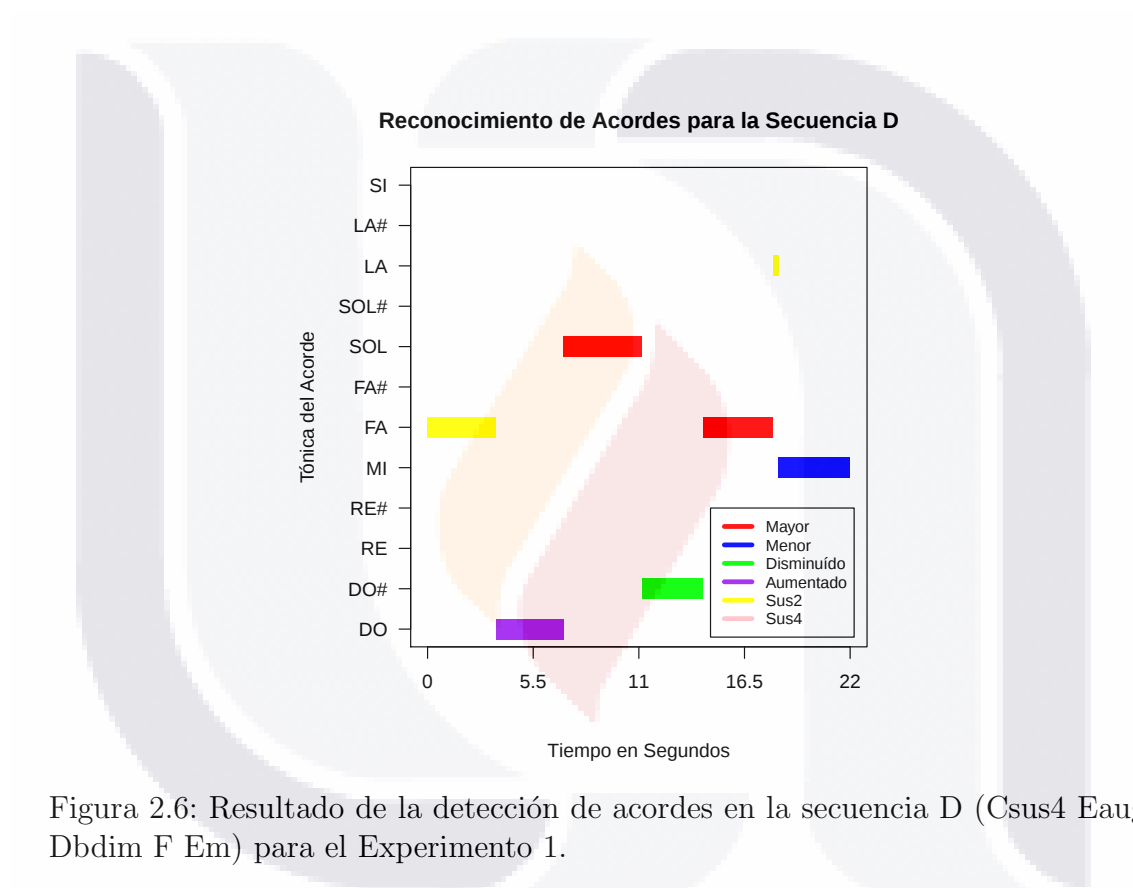


Figura 2.5: Resultado de la detección de acordes en la secuencia C (F $\sharp$ Gm Gdim D $\sharp$ Bm Bsus2) para el Experimento 1.



la mayoría de los errores que presenta la detección es en los inicios y finales de cada uno de los acordes.

Un caso singular se presenta en las progresiones A y B del experimento. Si se observan los acordes detectados en las secciones correspondientes a acordes del tipo sus4, se asignan acordes sus2 con una nota tónica diferente. Si se toman los acordes tanto el detectado como el que se pretendió tocar durante la grabación, y se analizan las notas que los componen, se aprecia que son exactamente las mismas. Por ejemplo, para el caso de la progresión A, el acorde sus4 propuesto es Dsus4 mientras que el acorde detectado por el algoritmo es un Gsus2. Analizando las notas que componen los dos acordes son D, G y A para Dsus4 y G, A y D para Gsus2. Así pues podría verse un acorde como una inversión del otro. Realmente dentro de la armonía musical se presentan una infinidad de ambigüedades acústicas dónde los acordes que pueden presentarse como únicos son las triadas mayores y las triadas menores. Así pues, podrían tomarse otros criterios al momento de nombrar un conjunto de notas tocadas al unísono, donde lo más natural sería tomar el contexto musical, el cual pude ser dado por una tonalidad en particular definida a priori. Para el algoritmo propuesto, y dado que no se supuso ninguna relación armónica entre los acordes a detectar, estos pueden ser vistos únicamente como una combinación de notas. Al ser un algoritmo secuencial, siempre dará como resultado el tipo de acorde que es procesado primero, en este caso los acordes sus2 antes que los acordes sus4.

2.3.2. Experimento 2

Como segundo experimento y siguiendo la conclusión obtenida a partir del Experimento 1, se proponen otras cuatro progresiones de acordes donde no se busca nombrar los acordes por tipo sino que se analizan como combinaciones de tres notas diferentes. Así mismo, y a fin de contar con archivos de audio con sus puntos de cambio exactos, se generan las progresiones a través de código partiendo de grabaciones de acordes individuales y recortándolos a longitudes específicas de 3 segundos. Las secuencias propuestas son:

- Secuencia A: Asus2 A#dim C#aug C#sus4 D# G#m
- Secuencia B: Bdim C C#m Csus2 Daug G#sus4
- Secuencia C: Am A# C#dim Dm Faug Gm
- Secuencia D: Bsus2 Csus4 D#aug E Edim F#m

Se calculan las combinaciones únicas de notas para formar cada uno de los acordes propuestos dando como resultado 52 triadas diferentes contrastando con los 72

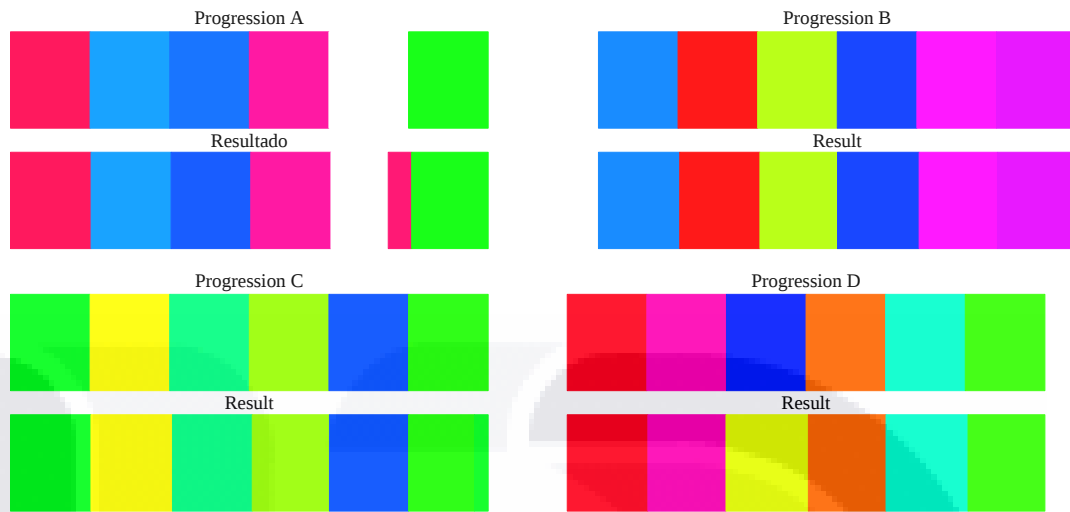


Figura 2.7: Resultado de la detección de acordes para las secuencias A (Asus2 A $\sharp$ dim Caug C $\sharp$ sus4 D $\sharp$ G $\sharp$ m), B (Bdim C C $\sharp$ m Csus2 Daug G $\sharp$ sus4), C (Am A $\sharp$ C $\sharp$ dim Dm Faug Gm) y D (Bsus2 Csus4 D $\sharp$ aug E Edim F $\sharp$ m) del Experimento 2.

acordes que pueden ser formados sobre cada nota cromática y de los 6 tipos propuestos. Esto indica que 20 de los acordes que pueden ser formados, serían equivalentes según el contexto. La Figura 2.7 muestra las gráficas de bloques de las triadas propuestas en la progresión y el resultado de la detección. Así mismo la Tabla 2.1 muestra las matrices de confusión para cada clasificación. La Tabla 2.2 muestra las medidas de exactitud, tasa de error, precisión y especificidad para la clasificación.

Se muestra en las gráficas que las triadas detectadas corresponden a las propuestas dentro de las progresiones. Sin embargo, desde el punto de vista temporal se observa un ligero desfase de los resultados con respecto a la secuencia real. Esto se hace más evidente en las matrices de confusión donde a pesar de ser los acordes correctos, el momento de inicio como del final de cada acorde presenta una ligera desviación.

2.3.3. Experimento 3

Como tercer experimento se proponen 4 grabaciones de una misma secuencia de acordes pero ejecutada en cuatro instrumentos diferentes, esto con la finalidad de comprobar el funcionamiento del algoritmo sin importar el timbre. Los instrumentos que se eligen para el experimento son una guitarra clásica, una guitarra eléctrica, un ukelele soprano y un teclado electrónico con el timbre configurado como piano de cola. El archivo de audio se forma de manera similar mediante código a partir de

	Progresión Real					
Resultado	4	21	35	37	55	58
4	71.94	0	0	0	0	0
21	0	96.05	0	0	0	0
35	0	0	98.76	1.76	0	0
38	0	0	0	98.24	1.81	0
55	2.54	0	0	0	98.19	0
57	25.52	3.95	0	0	0	0
58	0	0	1.24	0	0	100
	Progresión Real					
Resultado	1	14	36	39	50	51
1	97.92	2.84	0	0	0	0
14	0	97.16	0	0.32	0	0
36	2.08	0	100	0	0	0
39	0	0	0	99.68	0	2.83
50	0	0	0	0	98.81	0
51	0	0	0	0	1.19	97.17
	Progresión Real					
Resultado	11	15	20	22	26	38
11	98.93	0	0	0	3.43	0
15	0	96.33	0	0	0	0.62
20	0	0	82.78	0	0	0
22	1.07	0	17.16	100	0	0
26	0	3.67	0	0	96.57	0
38	0	0	0.06	0	0	99.38
	Progresión Real					
Resultado	5	19	29	40	54	60
5	97.11	0	0.46	0	0	0
12	2.89	0	0	100	0.6	0
19	0	96.55	0	0	0	0
29	0	3.45	99.54	0	0	0
54	0	0	0	0	98.09	0
60	0	0	0	0	1.31	100

Tabla 2.1: Tabla de confusión para las combinaciones de notas (números en comparativa) en las progresiones A, B, C y D del Experimento 2.

Progresión	Exactitud	Tasa de error	Precisión	Especificidad
Progresión A	0.94372512	0.05627488	0.61813524	0.97163196
Progresión B	0.994857647	0.005142353	0.984748051	0.996914588
Progresión C	0.985548	0.014452	0.9614419	0.9913288
Progresión D	0.94823381	0.05176619	0.70698967	0.9738687

Tabla 2.2: Medidas de calidad para el Experimento 3.

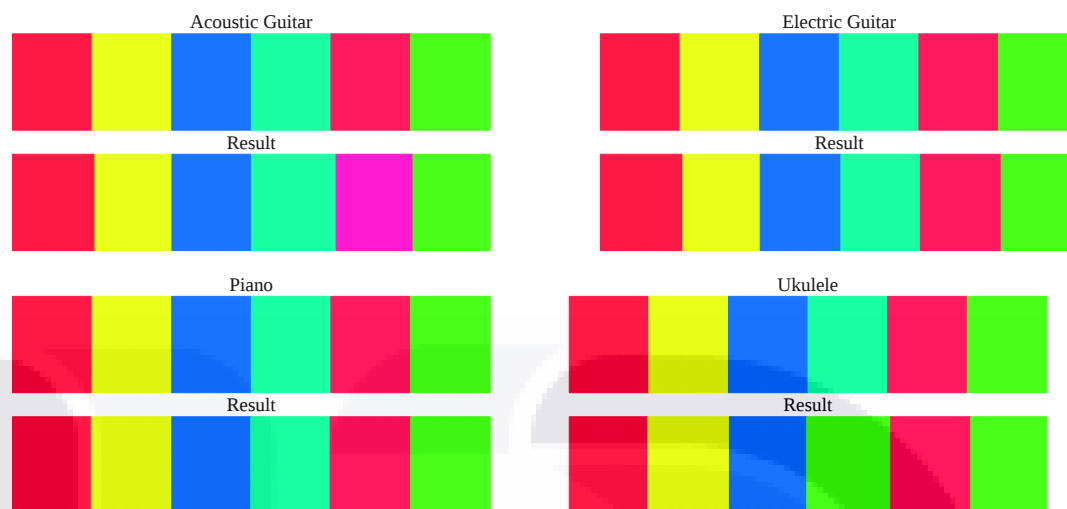


Figura 2.8: Resultado de la detección de acordes en la secuencia $A\sharp sus2$, B, Aug, Ddim, Esus4 y $F\sharp m$ para el Experimento 3 en guitarra clásica, guitarra eléctrica, piano y ukulele.

grabaciones individuales de cada acorde y recortados a una longitud de 3 segundos construyendo la progresión propuesta.

2.4. Conclusiones

Se propuso un algoritmo de análisis de audio basado en filtros de Gabor que permitió analizar cambios de la señal en el tiempo y sirvió de base para reconocer patrones espectrales en la señal. Se usó como ejemplo de aplicación un método de reconocimiento de acordes musicales presentes en un archivo de audio. Se utilizó la media geométrica como rasgo de caracterización espectral partiendo de la matriz chroma. Esta matriz fue calculada utilizando filtros de Gabor en su forma logarítmica. El algoritmo presentó un buen desempeño al clasificar progresiones de acordes mayores y menores en diferentes instrumentos, entre ellos la guitarra eléctrica, la guitarra acústica y el teclado simulando el timbre de un piano, con una efectividad de entre el 90 % y el 100 % para cada uno de los acordes.

El desempeño del algoritmo frente a diferentes instrumentos fue bueno y consistente presentando la mayoría de sus errores en el ukelele y la guitarra clásica, donde clasificó un acorde dentro de las progresiones por otro sin relación ni cercanía armónica.

	Progresión Real					
Resultado	12	19	27	37	58	59
12	96.45	0	0	0.29	0	0
19	0	96.91	0	0	0	0
27	0	0	99.87	0	6.45	0
37	0	0	0.13	99.71	0	0
53	0	3.09	0	0	93.55	0
59	3.55	0	0	0	0	100

	Progresión Real					
Resultado	12	19	27	37	58	59
12	96.51	0	0	0.69	0	0
19	0	96.65	0	0	0	0
27	0	0	97.99	0	1.79	0
37	0	0	2.01	99.31	0	0
58	0	3.35	0	0	98.21	0
59	3.49	0	0	0	0	100

	Progresión Real					
Resultado	12	19	27	37	58	59
12	100	0	0	0.14	0	1.01
19	0	99.85	0	0	0	0
27	0	0	99.01	0.66	0	0
37	0	0	0	99.19	0	0
58	0	0.15	0.99	0	100	0
59	0	0	0	0	0	98.99

	Progresión Real					
Resultado	12	19	27	37	58	59
12	100	0	0	1.17	0	1.45
19	0	95.65	100	1.7	3.77	0
37	0	0	0	97.13	0	0
58	0	4.35	0	0	96.23	0
59	0	0	0	0	0	98.55

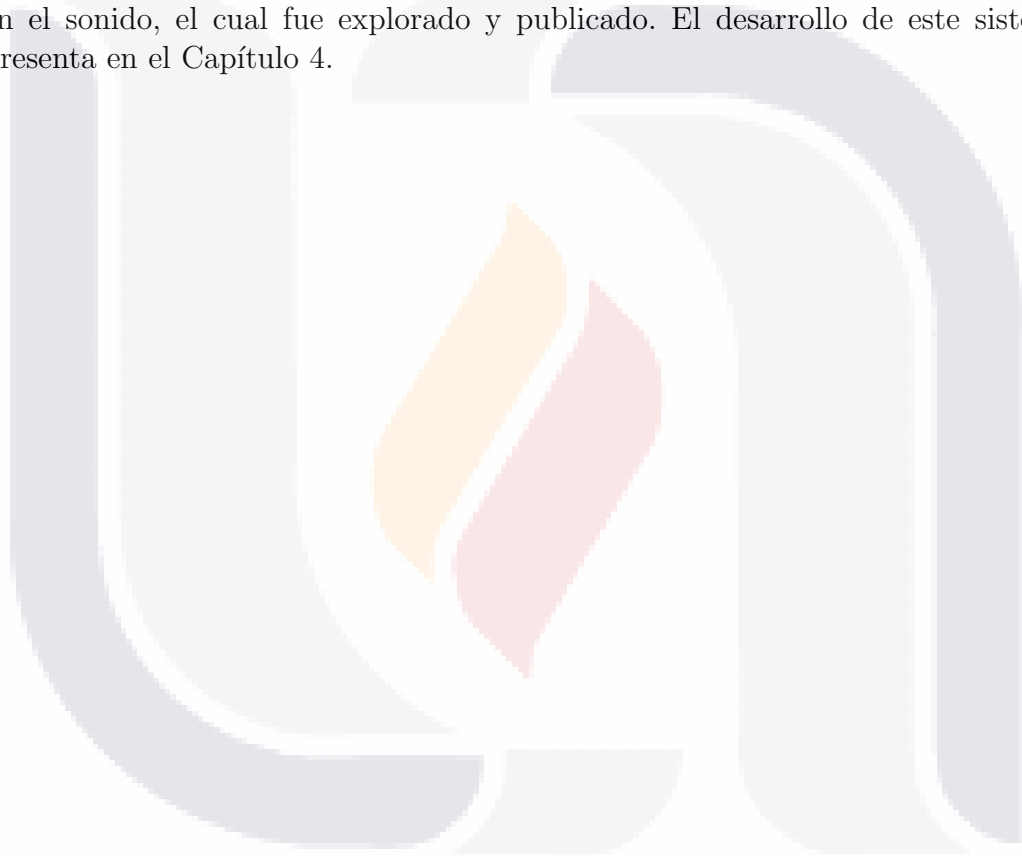
Tabla 2.3: Tabla de confusión para las combinaciones de notas (números en comparativa) en la progresión ejecutada en guitarra clásica, guitarra eléctrica, piano y ukelele del Experimento 3.

Instrumento	Exactitud	Tasa de error	Precisión	Especificidad
Guitarra clásica	0.9490199	0.0509801	0.7001073	0.9740137
Guitarra eléctrica	0.993702024	0.006297976	0.981392788	0.996221214
Piano	0.998358949	0.001641051	0.995116404	0.999015369
Ukelele	0.93752834	0.06247166	0.73445329	0.96251701

Tabla 2.4: Medidas de calidad para el Experimento 3.

En general, algunos de los errores de clasificación se presentaron al asignar un acorde como su acorde consecutivo en la progresión a clasificar. Esto se puede adjudicar a diferentes factores . En primer instancia la naturaleza de los filtros empleados, los cuales presentan incertidumbre en la localización en el tiempo, por lo que en los límites de cada cambio de acorde se genera una ligera mezcla de los mismos.

El sistema de análisis de audio mediante bancos de filtros de Gabor mostró buenos resultados y ayudó a establecer tanto el modelo básico como la metodología para ser empleado en un sistema de eco-localización y estimación de rango con base en el sonido, el cual fue explorado y publicado. El desarrollo de este sistema se presenta en el Capítulo 4.



Capítulo 3

Modelado y compensación del error en la estimación de la posición con encoders de bajo costo en robots de cuatro ruedas con redes neuronales artificiales

3.1. Marco teórico

El primer paso dentro del proceso de estimación de estados de un robot móvil en aplicaciones SLAM consiste en estimar su posición después de haberse accionado un control de movimiento en el mismo. Como se mencionó en el Capítulo 1, un método frecuentemente usado para la estimación de la posición de un vehículo a base de ruedas es la odometría en conjunto con algún modelo cinemático o dinámico para establecer la posición y pose del vehículo en un marco de referencia global [22][40][42][30]. Sin embargo, la presencia de agentes externos como el ruido, la calidad de los materiales del vehículo o sensores se pueden obtener estimaciones con errores considerables o modelos demasiado complejos. Así pues, se vuelve crucial encontrar un método que proporcione estimaciones precisas y de manera robusta a pesar de los antes mencionados factores de error.

Se ha demostrado que el uso de algoritmos de redes neuronales artificiales proporcionan una solución viable para estimaciones y regresiones en sistemas de control y robótica [20][19][34][41][21][33][15]. Con esto como referencia, se propone un método basado en redes neuronales para compensar el error de estimación de la distancia recorrida por el robot empleado en el desarrollo del tema de investigación.

Las redes neuronales artificiales son sistemas capaces de establecer relaciones no lineales entre los resultados y las observaciones individuales dentro de los procesos que los generan. Estas redes están compuestas por nodos individuales llamados neuronas, los cuales se conectan entre sí mediante pesos numéricos. Estos pesos se calculan a partir de los datos de entrada mediante operadores lineales y funciones de activación no lineales aplicadas a cada neurona. El objetivo de este proceso es minimizar la diferencia entre los resultados obtenidos y los valores esperados, lo que da lugar a una fase clave conocida como entrenamiento. Una vez determinados los pesos de las conexiones entre las neuronas, se genera un modelo que permite realizar predicciones a partir de datos de entrada nuevos.

3.1.1. Redes neuronales

En una red neuronal artificial (RNA) de tipo feed-forward, el modelo se construye mediante capas en las cuales todos los nodos de una capa están conectados a todos los nodos de la siguiente. La forma más básica de una red neuronal artificial consta de dos capas: una capa de entrada y una capa de salida, con al menos un nodo en cada una. Adicionalmente, el modelo puede incluir un número arbitrario de capas ocultas con cualquier cantidad de nodos en cada una. En este tipo de red, los datos fluyen desde la entrada hasta la salida, pasando por cada capa intermedia. El objetivo del entrenamiento de la red neuronal es calcular un conjunto de pesos para cada conexión, de manera que el error en la salida sea minimizado. La Figura 3.1 muestra una representación visual de una red neuronal artificial con 2 entradas y 1 capa oculta.

El modelo para una RNA se define mediante las siguientes ecuaciones:

$$y_j^{(1)} = g \left(\sum_{k=1}^K \omega_{jk}^{(1)} x_k + \theta_j^{(1)} \right), \quad j = 1, \dots, K_1 \quad (3.1)$$

$$z = \sum_{k=1}^{K_1} \omega_k^{(2)} y_j^{(1)} + \theta^{(2)} \quad (3.2)$$

donde x_k representa las K variables de entrada, $y_j^{(1)}$ son las K_1 variables de la capa oculta y z es la variable de salida. La matriz de pesos de la capa oculta se denota como $\omega_{jk}^{(1)}$, mientras que el vector de pesos asociado a la variable de salida es $\omega_k^{(2)}$. Los valores $\theta_j^{(1)}$ corresponden a los umbrales de la capa oculta, y $\theta^{(2)}$ es el término de sesgo para la salida. La función g representa la función de activación.

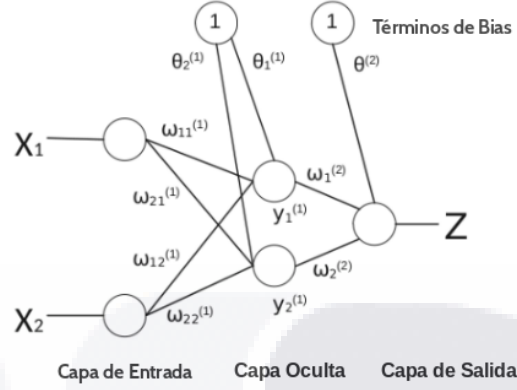


Figura 3.1: Representación de una red neuronal artificial de dos entradas y una capa oculta. En esta representación, los círculos vacíos representan las variables de cada capa, las líneas representan la conexión entre las neuronas que indican que el valor de una neurona en específico depende de neuronas previas. Los círculos con un valor unitario (1) dentro representan los términos de bias.

Asimismo, las ecuaciones (3.3), (3.4) y (3.5) representan una red neuronal con dos capas ocultas, donde la salida de la primera capa oculta sirve como entrada para la segunda.

$$y_j^{(1)} = g \left(\sum_{k=1}^K \omega_{jk}^{(1)} x_k + \theta_j^{(1)} \right), \quad j = 1, \dots, K_1 \quad (3.3)$$

$$y_j^{(2)} = g \left(\sum_{k=1}^{K_1} \omega_{jk}^{(2)} y_k^{(1)} + \theta_j^{(2)} \right), \quad j = 1, \dots, K_2 \quad (3.4)$$

$$z = \sum_{k=1}^{K_2} \omega_k^{(3)} y_k^{(2)} + \theta^{(3)} \quad (3.5)$$

A partir del análisis de estas ecuaciones, es posible generalizar el modelo para una red neuronal artificial de L capas.

El cálculo de los pesos para cada nodo puede abordarse desde diversas metodologías, que en esencia se reduce a un problema de optimización, en el que se busca minimizar el error entre la salida de la red neuronal y el conjunto de datos de entrenamiento. Uno de los algoritmos propuestos para resolver este problema es RProp.

RProp es un algoritmo de aprendizaje supervisado basado en un modelo de aprendizaje adaptativo local, diseñado para redes perceptrón multicapa. Su objetivo principal es reducir el impacto del tamaño de la derivada parcial en la actualización de los pesos.

El tamaño del ajuste en los pesos está determinado por el valor de actualización $\Delta_{ij}^{(t)}$, definido por:

$$\Delta\omega_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & \text{si } \frac{\partial E}{\partial \omega_{ij}}^{(t)} > 0 \\ \Delta_{ij}^{(t)} & \text{si } \frac{\partial E}{\partial \omega_{ij}}^{(t)} < 0 \\ 0 & \text{otro} \end{cases}$$

donde $\frac{\partial E}{\partial \omega_{ij}}^{(t)}$ denota la información del gradiente sumado sobre todos los patrones del conjunto de entrenamiento [28].

3.2. Metodología

El prototipo empleado en el proyecto de investigación corresponde a un vehículo de cuatro ruedas con tracción en cada llanta y donde la velocidad de cada motor es controlado por un valor de PWM. Cada llanta del vehículo cuenta con un encoder instalado. El procesamiento en el vehículo es realizado por un Arduino Mega 2560®, el cual está encargado de procesar y contabilizar la señal del encoder. La Figura 3.2 muestra una fotografía del prototipo empleado.

Dentro de la propuesta se asumen diferentes características del sistema empleado:

- El bajo costo de los componentes (encoders, procesamiento, etc.) significan un factor importante en el error obtenido de la estimación del recorrido con base a odometría.
- El error a ajustar se redujo únicamente a la estimación del recorrido lineal del vehículo.
- El ajuste de la estimación se hizo dentro de una ventana de movimiento definida.
- La precisión del modelo de encoder empleado (0.546 cm) produce un error inherente dentro de la estimación.

La Ecuación 3.6 muestra la expresión mediante la cuál puede estimarse la distancia recorrida por un vehículo dados los conteos odométricos de sus sensores. Sin embargo, como se muestra en la Tabla 3.1, dichas estimaciones presentan errores considerables conforme va aumentando el tiempo de recorrido del vehículo.

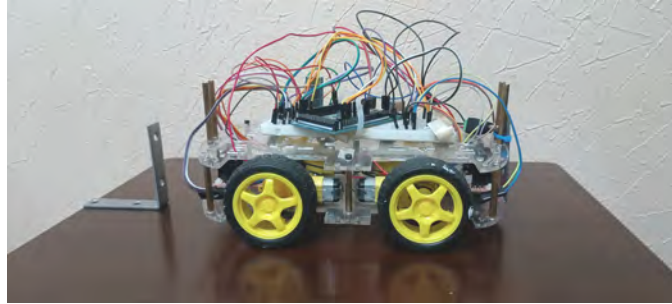


Figura 3.2: Fotografía del prototipo empleado.

WT1	WT2	WT3	WT4	Wm	ED(cm)	MD(cm)	Error
6	5	6	7	6	3.2	5.1	1.9
26	26	27	26	26.25	14.3	25.5	11.2
22	23	24	22	22.75	12.4	25.8	13.4
36	35	36	35	35.5	19.4	34.3	14.9
20	18	25	20	20.75	11.3	25.1	13.8
2	2	5	4	3.25	1.7	5.2	3.5
4	3	4	4	3.75	2.0	8.8	6.8
33	30	38	32	33.25	18.1	35.2	17.1
8	8	11	10	9.25	5.0	14.2	9.2
12	11	13	12	12	6.5	13.4	6.9

Tabla 3.1: WT1, WT2, WT3 y WT4 corresponden al número de tick para cada una de las llantas, W_m indica la media de los ticks de todas las llantas, ED es la distancia estimada mediante la expresión 3.6 y MD es la distancia medida[11].

$$\Delta l = D\pi \frac{ticks}{2n} \quad (3.6)$$

A fin de sopesar el error en la estimación de la distancia recorrida, se propone un modelo de red neuronal. La salida del modelo corresponde a la distancia recorrida por el vehículo en metros. Los experimentos se realizan siguiendo la configuración física del prototipo representada en la Figura 3.3 y se siguieron las siguientes restricciones:

- Las cuatro llantas del vehículo reciben el mismo control de velocidad PWM, que es un valor numérico entre 0 y 255.
- Cada una de las llantas tiene instalado un encoder óptico, cuya salida es conectada a un Arduino Mega 2560 para el conteo de tick.

- Los conteos recabados en cada movimiento del vehículo son transmitidos vía serial a la PC de procesamiento.
- Los datos recabados son procesados fuera de línea en **R** usando la biblioteca **neuralnet**, la cual contiene una implementación de modelos neuronales con retro-propagación. La configuración de la red neuronal usada es con una capa oculta de cinco neuronas. Fue elegida a prueba y error.

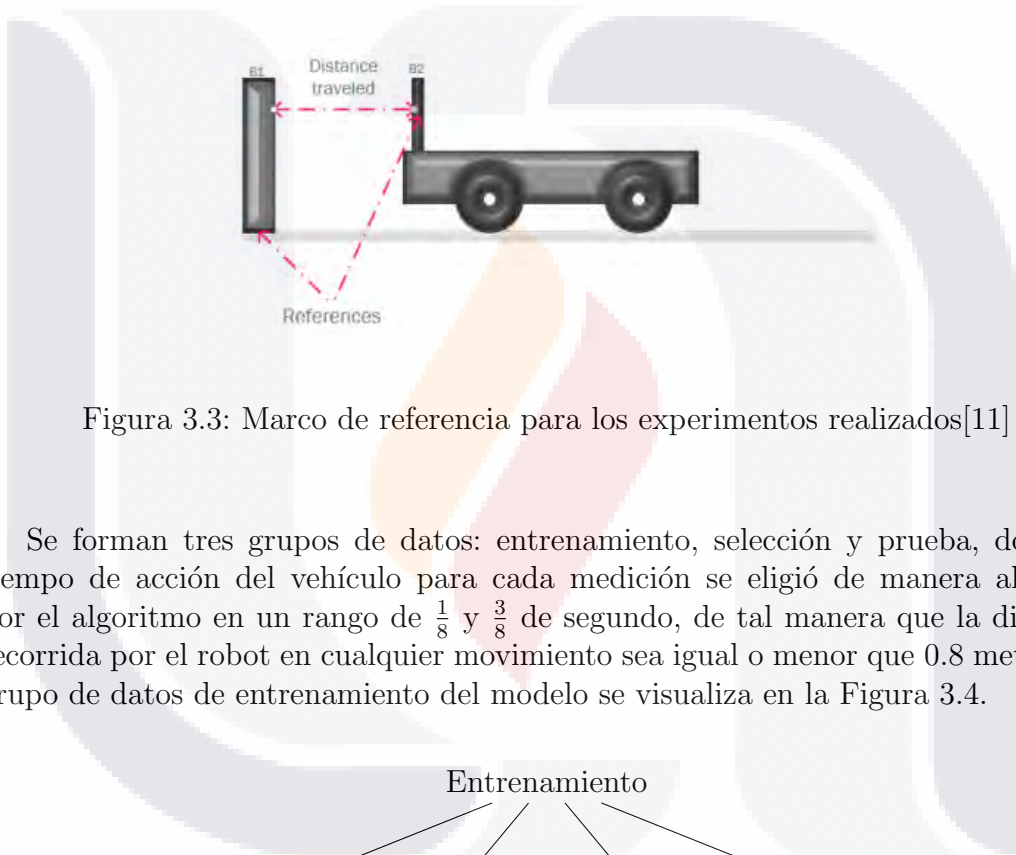


Figura 3.3: Marco de referencia para los experimentos realizados[11]

Se forman tres grupos de datos: entrenamiento, selección y prueba, donde el tiempo de acción del vehículo para cada medición se eligió de manera aleatoria por el algoritmo en un rango de $\frac{1}{8}$ y $\frac{3}{8}$ de segundo, de tal manera que la distancia recorrida por el robot en cualquier movimiento sea igual o menor que 0.8 metros. El grupo de datos de entrenamiento del modelo se visualiza en la Figura 3.4.

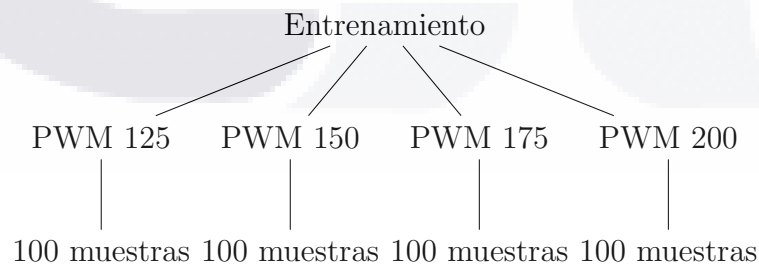


Figura 3.4: Composición del grupo de datos de entrenamiento del modelo de red neuronal artificial.

Estructura de las capas ocultas usadas para selección del modelo RNA
(3)
(4)
(5)
(6)
(7)
(3, 3)
(4, 4)
(5, 5)
(6, 6)
(7, 7)
(3, 3)
(4, 4)
(5, 5)
(6, 6)
(7, 7)

Tabla 3.2: Configuraciones de capas ocultas utilizadas en la selección del modelo de red neuronal. Los números entre paréntesis corresponden al número de neuronas en las capas ocultas del modelo.

3.3. Experimentos

El grupo de datos para la selección del modelo se construye con 115 nuevas mediciones donde el tiempo de acción y el PWM son seleccionados de manera aleatoria por el Arduino en cada movimiento. La selección del modelo se sustenta en el valor de MSE y el histograma de los errores de predicción, buscándose una distribución normal de los mismos. Se propone un modelo de red neuronal con cinco variables de entrada: El número de conteos de tick para cada una de las llantas más el valor de PWM aplicado a los motores para el movimiento. La salida del modelo corresponde a la distancia recorrida por el vehículo en metros. La configuración del modelo elegido es una red neuronal con una capa oculta y cinco neuronas. Ésta se elige de un grupo de posibilidades representadas en la Tabla 3.2. El valor de MSE de la red neuronal elegida es de 0.0004005325. La Figura 3.5 muestra el grafo correspondiente al modelo propuesto.

La Figura 3.6 muestra la distribución de los errores de estimación antes y después de usar la red neuronal elegida. En el grupo de datos para prueba, los parámetros como el PWM y el tiempo son elegidos aleatoriamente. Se tomaron 119 nuevas mediciones, comparando la distancia real recorrida por el vehículo con la distancia estimada por el modelo previamente entrenado y seleccionado. La Figura 3.7 muestra

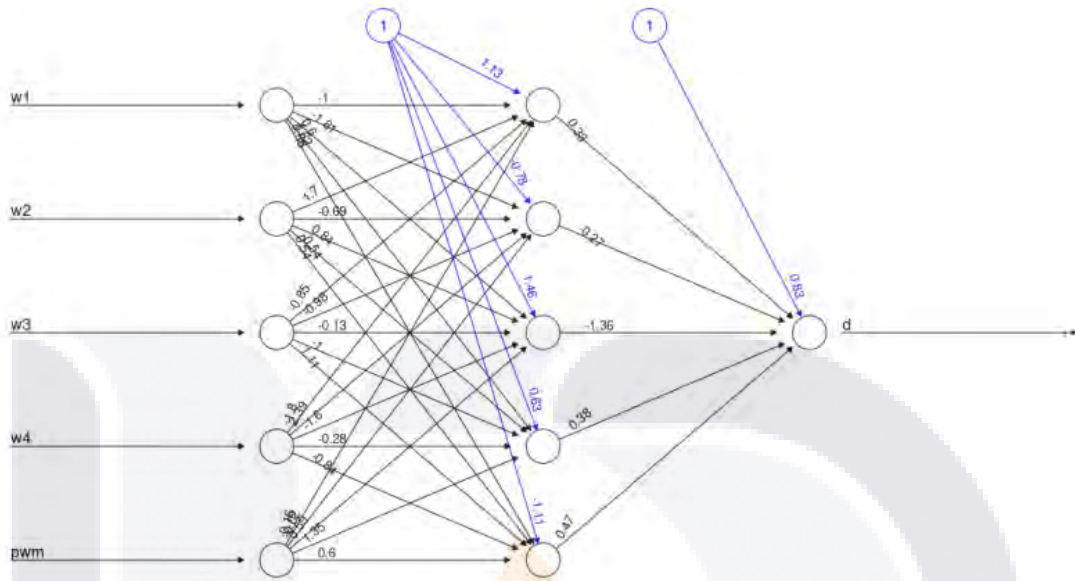


Figura 3.5: Grafo correspondiente al modelo de red neuronal propuesto.

una comparativa de la distribución de error para el grupo de datos de prueba de la red neuronal propuesta contra una función gaussiana y una función triangular, ambas centradas en la media de los errores obtenidos.

La Figura 3.8 muestra los resultados de estimación de la distancia lineal recorrida por el vehículo, donde la primera gráfica presenta la estimación basada en la expresión 3.6 mientras que la segunda muestra la salida de la red neuronal propuesta. Así mismo podemos mencionar las siguientes características que significan un resultado favorable del método propuesto:

- El comportamiento del error cuenta con una forma más acampanada, por lo cual es más fácil de tomar en cuenta en estimaciones futuras.
- La media global del error está al rededor de cero.
- Una prueba de normalidad de Shapiro-Wilk a los errores de las estimaciones del modelo arroja un estadístico de prueba $W=0.9820$ con un p-valor asociado de 0.1141, indicando poca significancia, y por ende no hay problema en asumir una distribución similar a la normal para los errores de las estimaciones.

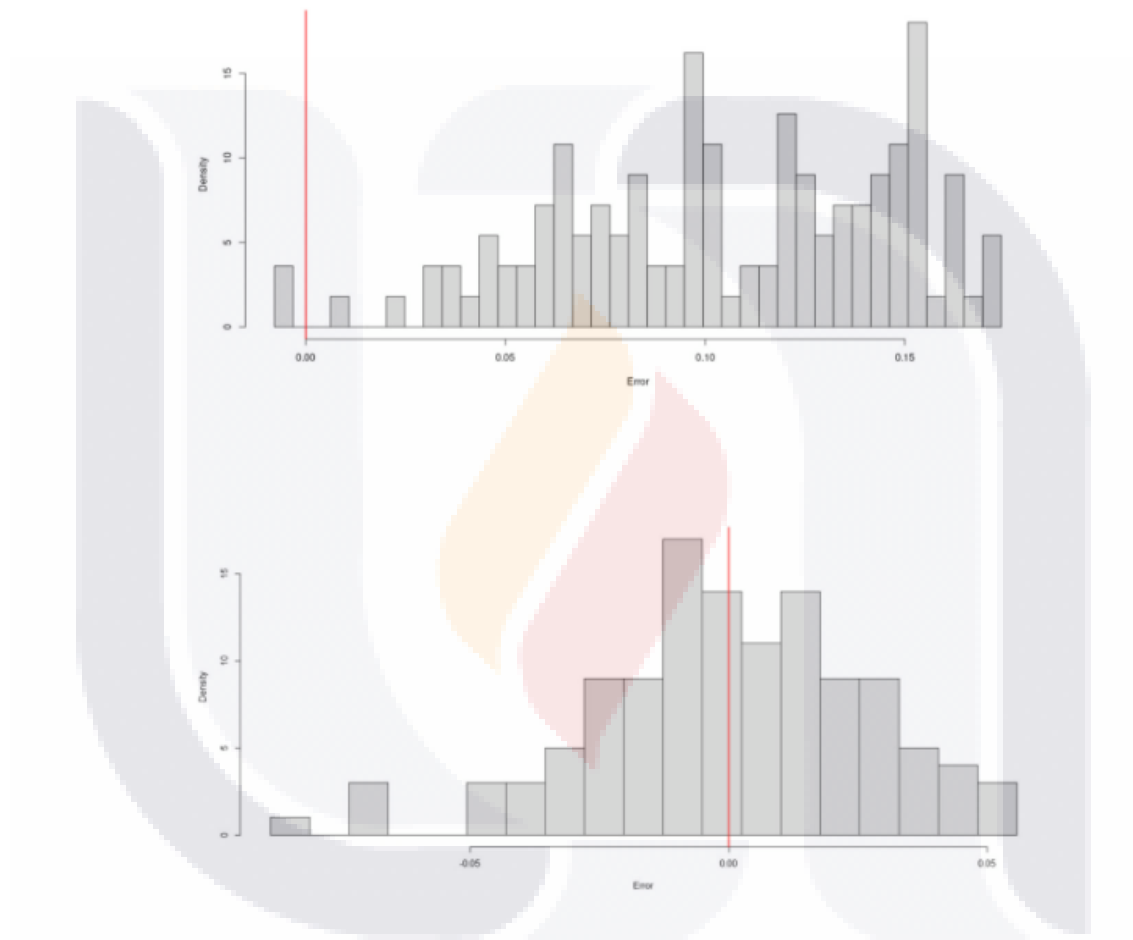


Figura 3.6: Histogramas de los valores de error de estimación (a) antes de la red neuronal propuesta, y (b) después de la red neuronal propuesta.

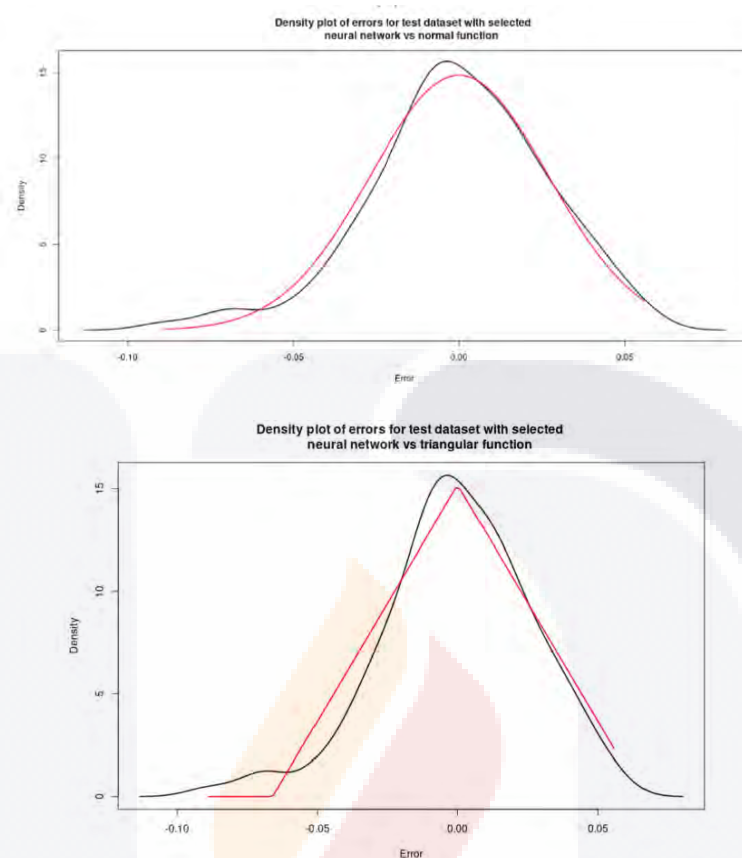


Figura 3.7: Comparación de la distribución de error para el grupo de datos de prueba de la red neuronal propuesta contra una función gaussiana y una función triangular centradas en la media de los errores.

3.4. Conclusiones

Como conclusión, la metodología mostró que la estimación de la posición de un robot está sometida a la capacidad de procesamiento y las características de los sensores. Sin embargo, si el comportamiento del error puede ser modelado, éste modelo podría ser usado para obtener estimaciones más precisas. Adicionalmente, el modelo odométrico probabilístico de estimación contempla que habrá errores en el cálculo de la posición del robot con respecto a un marco de referencia global. La propuesta de solución mostró un proceso práctico y preciso para la compensación de estimación de distancias empleando sensores de bajo costo o para su propia calibración. Así mismo el modelo de error resultante permitió ser introducido en el algoritmo de localización probabilístico propuesto en el proyecto de investigación.

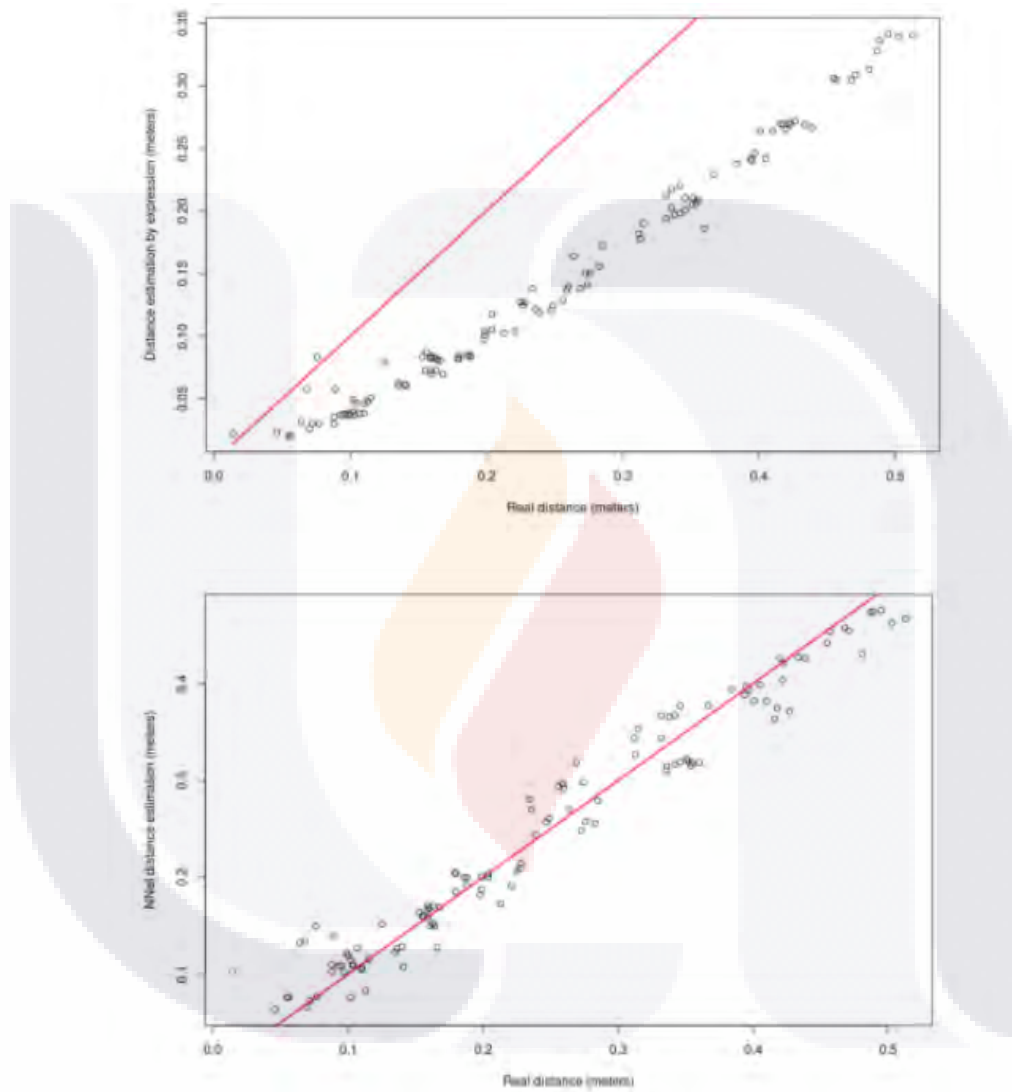


Figura 3.8: Resultados de la estimación de la distancia recorrida con la expresión lineal del encoder y con la red neuronal propuesta.



Capítulo 4

Método para la estimación de rango de obstáculos usando eco-localización con sonido de alta frecuencia

4.1. Marco teórico

Como se describió en la Sección 1.6, es requerido un modelo de mediciones con base en sensores de rango para introducir esta información en la actualización de los estados del mapa a construir. De igual manera como se mencionó en la Introducción del Capítulo 1, esta estimación del rango de obstáculos puede ser llevada a cabo de diferentes maneras, ya sea con láser, visión, o sonido.

Tomando en cuenta ventajas como que el procesamiento de señales de audio significaría una menor complejidad computacional en comparación con las imágenes o el video, los sensores láser o la visión computacional serían poco útiles en entornos de poca visibilidad y que las aplicaciones basadas en ultrasonido son poco adaptables a estimaciones de largas distancias dentro de un entorno de interiores, se propuso un sistema que emplea audio de alta frecuencia para la estimación de rango de obstáculos.

La localización acústica activa consiste en la identificación de la posición de un objeto a partir de la forma en que refleja el sonido [10]. Los sistemas básicos de localización acústica están constituidos por un emisor que genera de manera continua un pulso sonoro y un receptor que detecta los ecos producidos cuando dicho pulso incide sobre los objetos circundantes. Estos sistemas utilizan como referencia el instante en que se emite el pulso inicial, calculan el tiempo que tarda el eco en regresar al emisor (llamado *tiempo de vuelo*) y estiman la distancia con base en la

velocidad del sonido en el medio en el que opera el sistema[13]. La implementación precisa de estos sistemas conlleva múltiples desafíos, entre los cuales se destacan la selección óptima de la forma de onda y la frecuencia de los pulsos de referencia, el diseño y configuración de la infraestructura para la emisión y recepción de señales acústicas (incluyendo micrófonos, altavoces y otros dispositivos), el procesamiento de la señal del eco en entornos con presencia de ruido ambiental, así como la estimación precisa de la posición del objeto, entre otros factores [17].

Para la implementación eficiente de metodologías de localización acústica, es común el empleo de filtros de Gabor [9, 31, 16]. De igual manera, como se demostró en el artículo [12] y descrito en el Capítulo 2, el uso de filtros de Gabor para análisis y procesamiento de señales de audio es efectivo. Así pues, el objetivo principal del método propuesto publicado en el artículo [13] consistió en la estimación de la distancia de objetos estáticos en un entorno de interior, empleando el eco generado por dichos obstáculos en respuesta a pulsos de control emitidos a frecuencias elevadas.

4.1.1. Sonido y acústica

La acústica es la disciplina que se ocupa del análisis del sonido, incluyendo su generación, propagación a través de diferentes medios, y los diversos fenómenos que ocurren a lo largo de este proceso.

El *sonido* se define como un fenómeno ondulatorio que consiste en una perturbación que se transmite a través de un medio elástico, provocando fluctuaciones de presión o desplazamientos de las partículas que conforman dicho medio [27].

Las ondas sonoras pueden componerse de una sola frecuencia o estar integradas por una combinación de varios componentes espectrales.

La forma más sencilla de una onda sonora se describe mediante una onda sinusoidal, la cual puede representarse gráficamente en el plano cartesiano. En esta representación, el eje horizontal (abscisas) corresponde al tiempo, mientras que el eje vertical (ordenadas) representa el desplazamiento de las partículas del medio de propagación, o bien las variaciones en presión o densidad [8].

El punto máximo de la onda sinusoidal coincide con la región de mayor compresión del medio, mientras que los valores mínimos de la onda representan las fases de rarefacción del mismo.

Las ondas sonoras poseen ciertas características que permiten clasificarlas:

- **Intensidad:** Está relacionada con las variaciones de presión acústica y se mide habitualmente en decibeles en referencia a la fuente sonora.

- **Frecuencia:** Corresponde a la longitud de onda λ , que mide la distancia entre dos puntos del medio con idéntica presión.
- **Velocidad de propagación:** Representa la rapidez con la que las ondas se desplazan a través del medio. Por ejemplo, en aire a una temperatura estándar de 20°C, esta velocidad es aproximadamente $344 \frac{m}{s}$.

En el procesamiento de señales de audio, se deben tomar en cuenta principios y definiciones ampliamente desarrollados a lo largo de la historia de la física. A continuación se citan algunos de gran importancia dentro del campo de la localización acústica.

El principio de Huygens, establece que cada punto en el frente de onda actúa como una fuente de ondas secundarias esféricas que se propagan en todas las direcciones con la misma velocidad, frecuencia y longitud de onda que el frente principal del cual derivan.

Por otro lado, el *efecto Doppler* describe cómo el patrón de radiación acústica de una fuente sonora se altera cuando ésta se encuentra en movimiento. Si una fuente sonora con frecuencia f se aproxima al observador a una velocidad v , la frecuencia percibida f_d se calcula mediante:

$$f_d = \frac{f}{1 \pm \frac{v}{v_s}}$$

Aquí, el signo negativo corresponde a una fuente que se acerca, mientras que el positivo se utiliza para una fuente que se aleja. En términos angulares, donde θ es el ángulo entre la dirección de la fuente y el observador, el cambio de frecuencia se expresa como:

$$f_d = \frac{f}{1 - \frac{v \cos \theta}{v_s}}$$

donde v_s es la velocidad de propagación del sonido (aproximada en $344 \frac{m}{s}$). La *refracción* ocurre cuando las ondas sonoras cambian de dirección al atravesar diferentes medios debido a variaciones en su velocidad de propagación.

Cuando el sonido incide sobre una superficie, una parte de su energía es absorbida, mientras que la restante se refleja. La distribución de esta energía reflejada depende del ángulo de incidencia de la onda acústica.

La energía reflejada se clasifica en tres tipos principales [2]:

- **Reflexión especular:** La mayor parte de la energía reflejada rebota con un ángulo igual al de incidencia.
- **Reflexión dispersa:** Parte de la energía se refleja en múltiples direcciones alrededor del ángulo principal.
- **Retro-reflexión:** Una fracción menor de la energía regresa hacia la fuente original, siguiendo el mismo ángulo de incidencia.

El análisis de la energía retro-dispersada permite obtener información clave, como:

- La distancia entre la fuente y la superficie reflectante, basada en el tiempo entre el pulso inicial y el eco recibido.
- La velocidad del objeto reflectante, deducida a partir del cambio en frecuencia por el efecto Doppler.
- El ángulo de llegada de la onda reflejada, inferido mediante variaciones de fase.
- La naturaleza del objeto reflectante, estimada por la amplitud de la onda reflejada.

La *localización acústica* se refiere a la identificación de la posición y dirección de un sonido, ya sea a partir de su emisión directa o de sus reflexiones. Puede realizarse de dos formas:

- **Localización activa:** Consiste en emitir un sonido y analizar el eco recibido.
- **Localización pasiva:** Utiliza los sonidos generados por los objetos para determinar su ubicación.

Algunas de las señales más comúnmente empleadas en la localización acústica activa son:

- **Chirp:** Señales con frecuencias que varían en el tiempo, ideales para ambientes con ruido.
- **Bursts:** Pulsos de banda estrecha formados por ondas periódicas a una frecuencia específica, útiles para análisis de alta resolución.
- **Ricker:** Similar a la segunda derivada de un pulso gaussiano, empleado para lograr alta precisión.

La utilización de cada una queda definida por la aplicación o rango de frecuencias a trabajar.

4.1.2. La transformada wavelet

Las *wavelets* son transformadas matemáticas usadas para representar, analizar y procesar señales. Estas son funciones construidas sobre ondas de duración finita, media cero y un decaimiento rápido llamadas *kernels*. Existe un amplio conjunto de tipos de kernel wavelets donde su uso depende de la aplicación y el tipo de señal que será procesada. Las transformadas wavelets son clasificadas en dos tipos:

- Continua: Usadas para realizar un análisis temporal del espectro de la señal.
- Discreta: Empleadas en compresión o filtrado de ruido.

La transformada wavelet esta compuesta por dos operadores principales: escalamiento y desplazamiento. El escalamiento consiste en comprimir o expandir el kernel wavelet en el tiempo manteniendo una constante de proporcionalidad. Esto en el dominio de la frecuencia puede interpretarse como una función pasa-banda. Por otro lado, la operación de desplazamiento corresponde a transportar el kernel wavelet a lo largo de la señal procesada.

Filtrado de ruido empleando la transformada wavelet

El proceso de la eliminación de ruido usando la transformada discreta wavelet cuenta con tres pasos: descomposición multi-nivel, umbralización y reconstrucción de la señal.

La descomposición multi-nivel proporciona diferentes niveles de detalle de la señal procesada a diferentes resoluciones. Dos conjuntos de coeficientes son calculados en cada nivel de descomposición. Como se menciona en [23], la señal discreta en una resolución consecutiva a cierto nivel de detalle cD_j se calcula mediante una convolución de la señal del nivel previo cA_{j+1} con un filtro discreto G y tomando únicamente una muestra de cada dos (*down-sampling*). El filtro representado por G corresponde a un filtro pasa-altas, el cual a su vez tiene un correspondiente filtro relacionado en cuadratura, H [7], el cual es interpretado como un filtro pasa-bajas de la señal discreta del nivel anterior, cA_{j+1} . Este proceso resulta en la aproximación de los coeficientes del siguiente nivel cA_j . Así pues, la representación completa wavelet se compone de los dos conjuntos de coeficientes resultantes (cA_j, cD_j). Basado en lo anterior, la descomposición multi-nivel implica llevar acabo el proceso anterior en cascada, tomando la salida del filtro pasa-bajas como la entrada del mismo proceso.

La reconstrucción de la señal previa a un nivel dado se lleva acabo introduciendo ceros entre cada muestra de cA_j (*up-sampling*) y cD_j y convolucionando las señales resultantes con los filtros H y G respectivamente multiplicando por 2 [23]. Las Figuras 4.1 y 4.2 muestran los procesos de descomposición y reconstrucción previamente descritos:

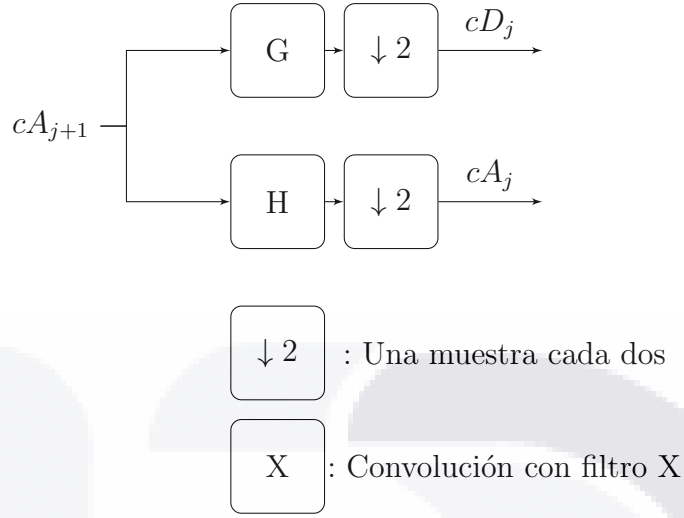


Figura 4.1: Esquema de la metodología de descomposición wavelet multi-nivel.

La umbralización (referido en inglés como *thresholding*) es el proceso en el cual se establece un valor de referencia en una señal, donde por encima o por debajo de este punto, se establece un valor particular en ese punto de la señal. El proceso de umbralizado puede ser clasificado en tres tipos:

- Hard-thresholding o umbralizado duro, los valores por debajo del valor de referencia son igualados a cero mientras que los valores superiores son multiplicados por una función identidad.
- Soft-thresholding o umbralizado suave, la señal es establecida en cero por debajo del valor de umbral, mientras que los valores superiores son atenuados en un factor proporcional al valor de referencia.
- Shrinkage, la señal es umbralizada basada en una función, manteniendo bajos los valores pequeños y manteniendo idénticos los valores superiores.

La Figura 4.3 muestra cómo cada método de umbralizado afecta a una función identidad $y = x$ con un umbral en $y = 0$.

La selección de un valor de umbralización dentro del proceso de filtrado de ruido empleando descomposición wavelet es un caso de estudio y depende del tipo de señal a procesar. De acuerdo al trabajo publicado en [5], se propone emplear un umbral universal, el cual es incorporado dentro de su método VisuShrink definido como:

$$t_n = \epsilon_1 \sqrt{2 \log(n)} \quad (4.1)$$

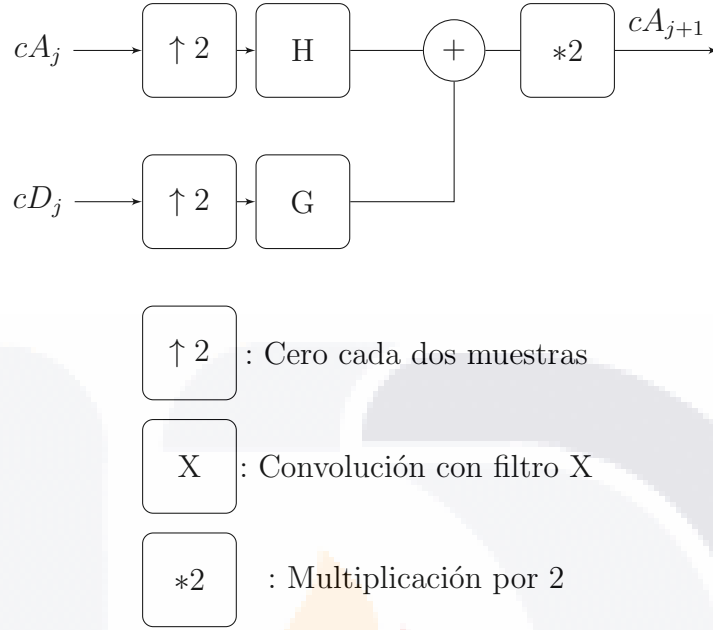


Figura 4.2: Esquema de la metodología de reconstrucción wavelet multi-nivel

Donde ϵ_1 es una estimación del nivel de ruido σ , y n es el número de muestras. La estimación del nivel de ruido, de acuerdo a [5] puede realizarse utilizando el estimador MAD definido como:

$$\epsilon_1 = \left[\frac{\text{median}|y_t|}{0.6745} \right]^2 \quad (4.2)$$

Donde y_t son las muestras en el nivel de más detalle de la descomposición wavelet de la señal. Así pues, de acuerdo a [5][6], para recuperar una señal a partir de datos sometidos a ruido, es suficiente con aplicar soft-thresholding a los coeficientes de detalle obtenidos mediante la función de descomposición wavelet, empleando el umbral constante t_n calculado previamente usando el estimador MAD.

Una vez sentadas las bases del procesamiento de señales, a continuación se desarrolla la metodología del sistema basado en audio de alta frecuencia para la estimación de rango de obstáculos.

4.2. Metodología

El algoritmo propuesto se estructura en cinco etapas:

- Preprocesamiento.

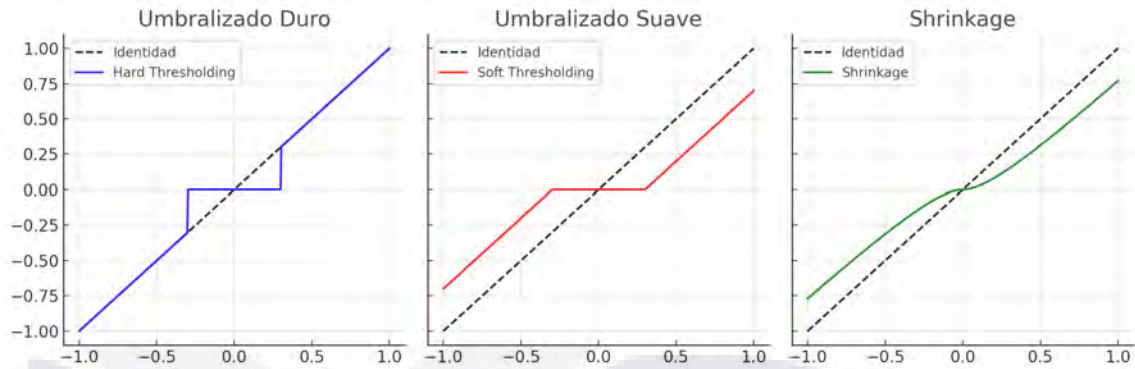


Figura 4.3: Gráficas que muestran cómo cada método de umbralizado afecta a una función identidad $y = x$.

- Filtrado de Gabor.
- Eliminación de ruido mediante transformada wavelet.
- Umbralización de la señal.
- Cálculo del tiempo de vuelo y estimación de la distancia en tiempo real.

La Figura 4.4 muestra un diagrama de la metodología propuesta con cada una de sus etapas.

Durante el preprocesamiento de la señal, la señal captada por el micrófono se acondiciona para su posterior procesamiento. Se normaliza la amplitud de la señal y su longitud se ajustó a la potencia de dos más próxima, descartando las muestras necesarias en la cola de la señal. Este ajuste es fundamental para garantizar una descomposición wavelet discreta adecuada, dado que dicho proceso requiere que la señal tenga una longitud en potencias de dos.

Para el proceso de filtrado en frecuencia se diseña un filtro unitario de Gabor centrado en la frecuencia principal del tren de pulsos utilizado como referencia. Este filtro permite enfocar la señal en la frecuencia esperada, facilitando la extracción de información relevante. El valor de σ del filtro de Gabor se selecciona mediante un método iterativo de prueba y error, buscando un equilibrio óptimo entre la precisión temporal y frecuencial.

Tras la aplicación del filtrado frecuencial, la señal se somete a un proceso de eliminación de ruido basado en la descomposición multi-nivel de wavelet discreta y la metodología VisuShrink, descrita en [5, 6]. Posteriormente, la señal se rectifica para mejorar su interpretación. Para suprimir variaciones espurias en el nivel base

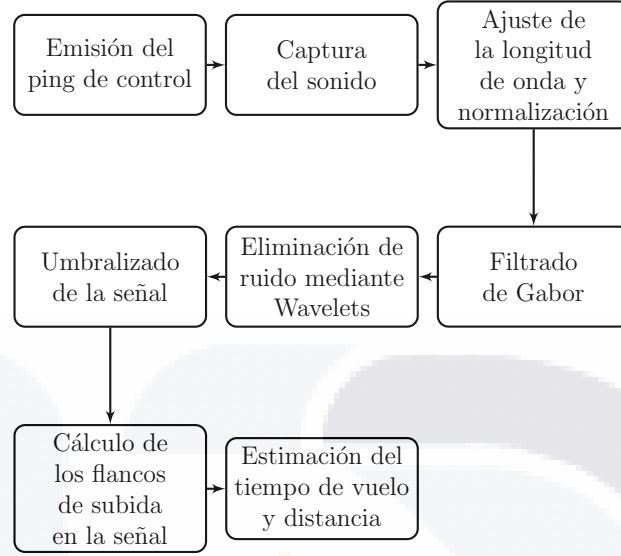


Figura 4.4: Diagrama de la metodología propuesta.

de la señal, se aplica un umbral duro, estableciendo en cero aquellas componentes cuya amplitud sea inferior al 2 % del valor máximo de la onda.

Para el cálculo de tiempo de vuelo, la señal procesada se separa en bloques bien definidos, correspondientes tanto al tren de pulsos del ping de control como a los ecos detectados. Para identificar los instantes clave de estos eventos, se aplica una función de detección de transiciones a lo largo de la señal, permitiendo determinar los instantes de transición ascendente y descendente dentro de cada bloque. La diferencia temporal entre el primer flanco ascendente (asociado al tren de pulsos de control) y los flancos ascendentes subsecuentes (correspondientes a los ecos reflejados por los objetos) permite calcular el tiempo de propagación de la onda en el aire.

Finalmente, la distancia entre la fuente emisora y el objeto reflectante que generó el eco se estima mediante la siguiente expresión:

$$D = \frac{S_{ERE} - S_{PRE}}{f_s} \times \frac{343}{2} \quad (4.3)$$

Donde D representa la distancia estimada al obstáculo, S_{ERE} denota la muestra correspondiente al flanco ascendente del eco detectado, S_{PRE} indica la muestra en la que se registró el flanco ascendente del pulso de control, y f_s es la frecuencia de muestreo. La constante 343 corresponde a la velocidad del sonido en el aire (m/s), y el factor de división por 2 considera el trayecto de ida y vuelta de la onda sonora.

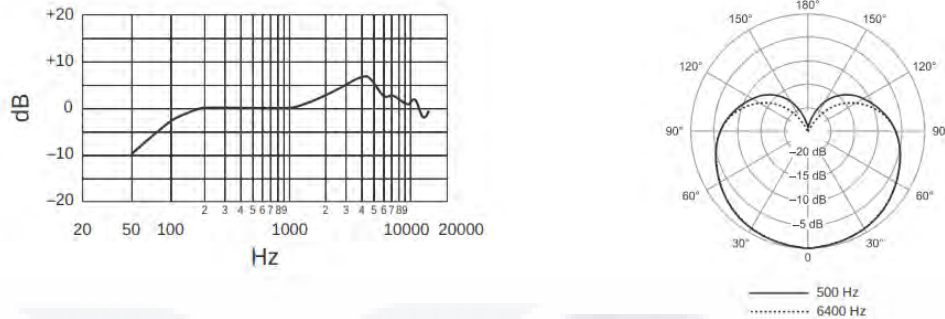


Figura 4.5: Respuesta en frecuencia y patrón polar del micrófono Shure® SV-100. Figura tomada del manual del dispositivo.

4.3. Experimentos

Los experimentos propuestos consisten en estimar la distancia de un obstáculo dispuesto a diferentes distancias del dispositivo de recolección de audio dentro de un ambiente interior. El obstáculo utilizado es una caja metálica de dimensiones de $44 \times 44 \times 47$ centímetros. Durante los experimentos, se utilizó un micrófono Shure® SV-100 con un patrón de captura cardioide, el cual capturó el sonido ambiental y se apuntó en dirección al obstáculo. La Figura 4.5 muestra el patrón y la respuesta en frecuencia del micrófono utilizado. La señal para el ping de control se genera mediante un script en el software R. Las ondas estudiadas son: sinusoidal, cuadrada, triangular y dientes de sierra. El tren de pulsos se formó con cinco ciclos de cada señal con una frecuencia fija de 15kHz y una frecuencia de muestreo de 44100 muestras por segundo. Estos trenes de pulsos son reproducidos a través de un altavoz activo Bose Color II®, grabando el sonido ambiental usando el software Audacity y una interfaz Behringer U-PHORIA UMC202HD®.

El algoritmo es implementado en Python utilizando la biblioteca *pywt* para calcular la onda diezmada mediante wavelets. La familia de wavelets seleccionada es Daubechies con una longitud de filtro de 2 coeficientes. El umbral para la reducción de ruido es determinado empleando el método VisuShrink [5, 6]. El filtro de Gabor unitario se construye a partir de las ecuaciones 2.3 y 2.4, centrado en 15 KHz con un valor de σ de 300. El parámetro σ se selecciona mediante un proceso de prueba y error. La Figura 4.6 muestra el espectro de la grabación del micrófono para la estimación de la distancia al obstáculo, utilizando un tren de pulsos sinusoidales de 15 KHz. En el mismo gráfico se presenta la respuesta del filtro de Gabor. La Figura 4.7 ilustra los resultados de cada una de las etapas del método propuesto: (a) señal original registrada por el micrófono, (b) señal filtrada por el filtro de Ga-

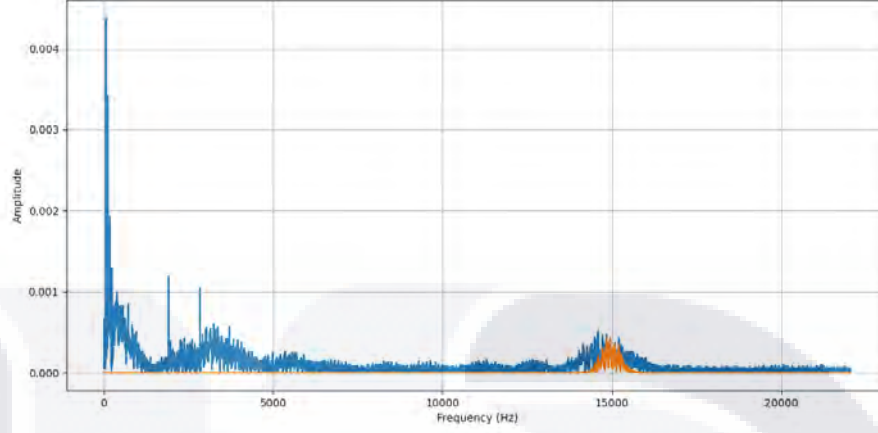


Figura 4.6: Se muestra en azul el espectro de la grabación correspondiente a la estimación de rango utilizando un tren de pulsos de una señal sinusoidal de 15KHz. En naranja se muestra la respuesta unitaria de un filtro de gabor centrado a 15 KHz con valor de σ 300[13].

bor propuesto, (c) señal procesada mediante descomposición wavelet y reducción de ruido con VisuShrink (posteriormente rectificada), y (d) señal umbralizada al 2 % del máximo de la señal. Esta figura representa el proceso completo utilizando una onda sinusoidal de 15 KHz.

Una vez identificados los bordes ascendentes de cada uno de los bloques en la señal, es posible estimar la distancia al obstáculo. En este caso, se asume que el primer bloque (de mayor amplitud) corresponde al pulso de control y el bloque inmediato siguiente representa el eco reflejado por el obstáculo. Con base en los resultados del procesamiento de la señal y la ecuación 4.3, se lleva a cabo la estimación del rango utilizando cada una de las formas de onda propuestas. En la Tabla 4.1 presentan los resultados de la estimación de la distancia entre el obstáculo y el sistema de adquisición de sonido. El objeto se sitúa a cuatro distancias distintas, realizando pruebas con cada una de las formas de onda.

La distancia se selecciona de manera que el pulso de control y los ecos no se solaparan entre sí. La distancia mínima teórica a la que un eco puede recibirse sin superposición se calcula mediante:

$$\delta = 2Tnv_s, \quad (4.4)$$

donde δ representa la distancia mínima sin solapamiento del eco, T es el período de la onda utilizada para el pulso, n es el número de ciclos y v_s es la velocidad del sonido a 20°C. Aplicando la ecuación 4.4, la distancia mínima en nuestra aplicación

Onda sinusoidal		
Distancia Real (cm)	Estimación (cm)	Error
130	127.16	2.84
140	134.16	5.84
150	144.27	5.73
160	153.22	6.78
Onda cuadrada		
Distancia Real (cm)	Estimación (cm)	Error
130	128.72	1.30
140	135.72	4.28
150	145.83	4.17
160	154.77	5.23
Onda dientes de sierra		
Distancia Real (cm)	Estimación (cm)	Error
130	128.72	1.30
140	136.50	3.50
150	144.66	5.34
160	156.33	3.67
Onda triangular		
Distancia Real (cm)	Estimación (cm)	Error
130	126.38	3.62
140	135.72	4.28
150	143.11	6.89
160	154.77	5.23

Tabla 4.1: Tabla de la estimación de la distancia a un obstáculo usando cada una de las formas de ondas propuestas. Se emplea un tren de pulsos compuesto por 5 ciclos completos de la onda a una frecuencia de 15 KHz.

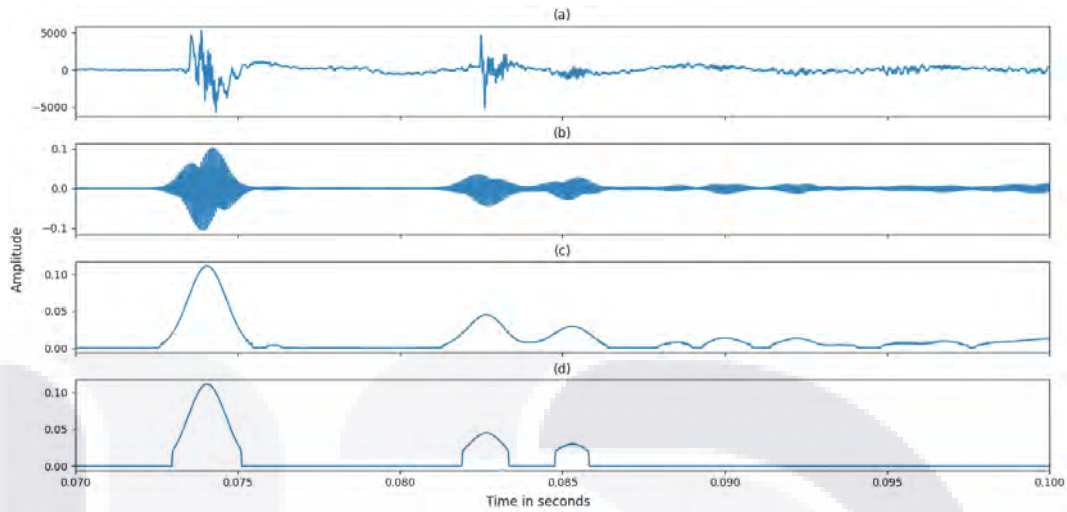


Figura 4.7: Resultado del procesamiento de una señal sinusoidal con un obstáculo a 30cm: (a) señal original obtenida a través del micrófono, (b) onda después del filtrado de Gabor en frecuencia, (c) señal después de la eliminación de ruido con wavelets y (d) onda umbralizada[13].

a 15 KHz es aproximadamente 22.86 cm. No obstante, para garantizar este parámetro, se selecciona una distancia considerablemente mayor para la realización de los experimentos. Para evaluar esta distancia mínima, se repite el experimento con el obstáculo ubicado a 30 cm del micrófono. La Figura 4.8 muestra las etapas de procesamiento de la señal grabada a esta distancia utilizando la misma onda sinusoidal de 15 KHz como pulso de control.

El resultado de la estimación en este experimento fue 24.88 cm, lo que implica un error de 5.12 cm. Aunque el error se encuentra dentro de los rangos previamente obtenidos, se observa que la separación entre el primer bloque (correspondiente al pulso de control) y el siguiente es muy reducida. Este experimento permitió determinar la distancia mínima medible utilizando las formas de onda propuestas. Asimismo, se realizan experimentos adicionales para comparar la estimación de distancia del método propuesto frente a sensores convencionales utilizados en robótica [1]. Los sensores seleccionados son el sensor ultrasónico HC-SR04 y el sensor infrarrojo SHARP GP2Y0A21YKOF. En estos experimentos, se realizan 5 mediciones de distancia en el rango de 20 cm a 200 cm con incrementos de 20 cm utilizando cada método. La Tabla 4.2 presenta los resultados obtenidos en cada experimento, mientras que la Figura 4.9 ilustra la comparación de la media de los errores absolutos para cada metodología.

Tabla 4.2: Resultados comparativos de las estimaciones obtenidas con el método propuesto, un sensor ultrasónico y un sensor infrarrojo. Cada experimento consistió en realizar 5 mediciones de distancia entre 20 cm y 200 cm con incrementos de 20 cm utilizando cada método. La tabla presenta el promedio de las estimaciones, la desviación estándar y el error absoluto medio para cada medición.

Media de las estimaciones			
Distancia real (cm)	Método propuesto (cm)	Ultrasónico	Infrarrojo
20	22843.33	21.54	17.19
40	41.45	40.39	44.24
60	56.77	60.11	74.87
80	75.05	80.43	126.65
100	92.00	99.35	152.87
120	112.15	117.92	134.06
140	130.73	138.82	119.24
160	154.38	136.68	96.99
180	172.50	177.55	91.11
200	196.38	197.46	105.04
Desviación estándar de las estimaciones			
Distancia real (cm)	Método propuesto (cm)	Ultrasónico	Infrarrojo
20	0.00	0.18	0.31
40	0.21	0.05	1.07
60	0.27	0.25	3.00
80	0.39	0.40	14.84
100	0.21	0.32	15.94
120	0.58	0.39	28.08
140	0.32	0.73	42.15
160	0.00	18.74	8.24
180	0.34	0.26	9.84
200	0.00	0.03	18.62
Media de los errores absolutos			
Distancia real (cm)	Método propuesto (cm)	Ultrasónico	Infrarrojo
20	22823.33	1.54	2.80
40	1.45	0.39	4.24
60	3.23	0.20	14.87
80	4.95	0.43	46.65
100	7.99	0.64	52.87
120	7.85	2.07	20.57
140	9.26	1.21	42.17
160	5.62	23.31	63.00
180	7.49	2.44	88.88
200	3.62	2.53	94.95

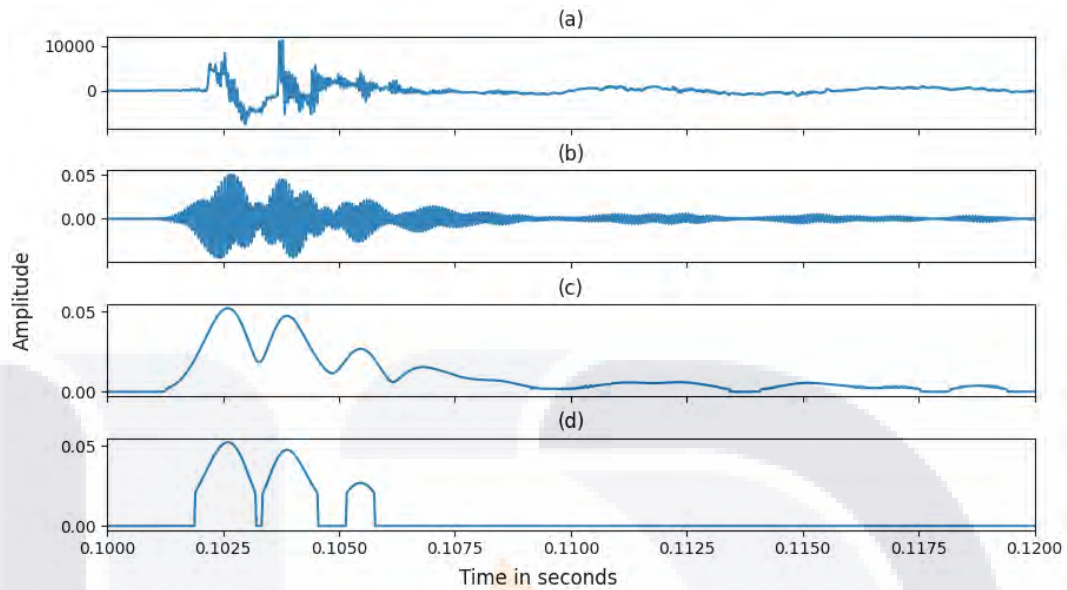


Figura 4.8: Resultado del procesamiento de una señal sinusoidal con un obstáculo a la mínima distancia permitida por el sistema: (a) señal original obtenida a través del micrófono, (b) onda después del filtrado de Gabor en frecuencia, (c) señal después de la eliminación de ruido con wavelets y (d) onda umbralizada[13].

4.4. Conclusiones

Los resultados obtenidos indican que la metodología propuesta permite una estimación precisa de la distancia a los obstáculos mediante el análisis de ecos de alta frecuencia, lo cual permite que el sistema sea utilizado como sensor de estimación de rango dentro del algoritmo de SLAM propuesto en el tema de investigación. Si bien el error de estimación fue similar para todas las formas de onda evaluadas, la onda diente de sierra mostró un mejor desempeño en términos de precisión. Sin embargo, es importante considerar otros factores como la amplitud de la onda emitida, lo que puede incrementar la ganancia del eco y facilitar su detección, en contraposición a las pérdidas energéticas y la susceptibilidad al ruido derivadas de la dispersión frecuencial en cada una de ellas.

El método propuesto se fundó en el procesamiento de señales de audio mediante filtrado de Gabor, cuya funcionalidad fue explorada en audio en el Capítulo 2, y la descomposición wavelet multinivel para la eliminación de ruido.

Por otro lado, esta aplicación brinda la oportunidad de ser implementada con elementos de bajo costo y fácil accesibilidad, así como su capacidad de poder adaptarse el algoritmo a diferentes formas de onda y frecuencias, esto con la finalidad de obtener una mayor precisión o adaptabilidad al entorno donde sea empleado.

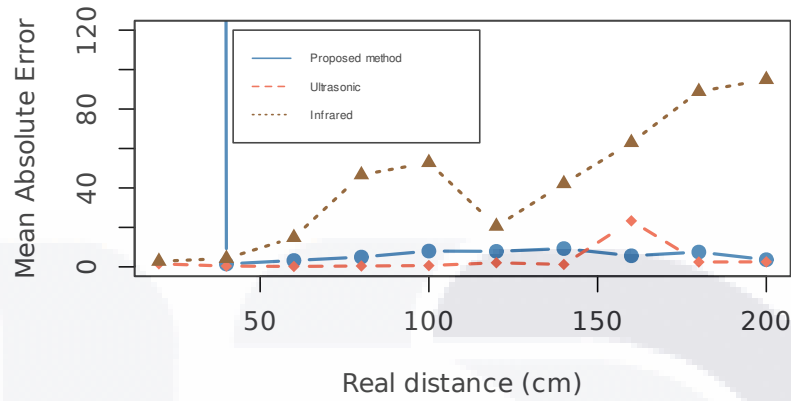


Figura 4.9: Comparación entre el método propuesto, un sensor ultrasónico y un sensor infrarrojo para la estimación de distancia. Los experimentos consistieron en realizar 5 mediciones de distancia entre 20 cm y 200 cm con incrementos de 20 cm utilizando cada método[13].

Los experimentos demostraron que la metodología propuesta en este trabajo permite obtener estimaciones precisas en un amplio rango de operación. Cabe destacar que las mediciones realizadas para la distancia mínima permitida por el método (24.88 cm) fueron consistentes y competitivas en comparación con el sensor ultrasónico, e incluso mostraron un desempeño superior respecto al sensor infrarrojo seleccionado para estos experimentos. La primera estimación con el método propuesto resultó en un error de valor por defecto (22843.33) debido a que la distancia se encontraba por debajo del rango mínimo medible por la metodología.

Capítulo 5

Conclusiones y trabajo a futuro

5.1. Conclusiones

Como se definió a lo largo de este trabajo, la robótica probabilística es un enfoque en el que se dota a los sistemas robóticos de mecanismos o métodos para manejar la incertidumbre en su percepción (sensores) y acción (controles) a través de modelos probabilísticos. Los métodos propuestos en este campo pretenden mejorar la toma de decisiones en entornos dinámicos e impredecibles, sustentados principalmente en estadística bayesiana y aprendizaje automático.

El proyecto de investigación propuesto permitió indagar en uno de los principales temas del campo de la robótica probabilística: El mapeo y localización simultánea (SLAM). Se estudiaron e implementaron algoritmos de vanguardia del tema, lográndose identificar puntos de acción para la generación de conocimiento inédito. Como se mencionó a lo largo de los capítulos, el SLAM consta de dos acciones importantes: la estimación de la posición del robot después de una instrucción de control, y la actualización de dicha estimación dada la percepción con la cual cuenta el robot. En el presente trabajo se presentaron tres propuestas que abordan estas dos fases de manera novedosa: la estimación de la distancia recorrida por un vehículo robótico de cuatro ruedas usando información odométrica y un modelo de redes neuronales artificiales; y un modelo de estimación de rango basado en sonido empleando filtros de Gabor y descomposición multinivel con wavelets. Estos dos últimos conceptos fueron analizados en profundidad en su aplicación para el procesamiento de señales de audio, produciendo un artículo de su aplicación dentro del campo de la música.

5.2. Trabajo a futuro

El campo de la robótica probabilística ha experimentado avances significativos en los últimos años, particularmente en la localización y mapeo simultáneo (SLAM). Con la integración de técnicas de aprendizaje profundo, computación en la nube y optimización de algoritmos probabilísticos, se han logrado importantes mejoras en la precisión y eficiencia de los sistemas autónomos. Sin embargo, aún persisten desafíos que deben ser abordados para garantizar una navegación robusta en entornos dinámicos, no estructurados y con restricciones computacionales. Así pues, el campo de la robótica probabilística es actualmente una importante área de oportunidad para el desarrollo y la investigación.

El trabajo presentado permitió abordar las bases del área de la robótica probabilística así como reveló el panorama sobre los temas que permiten continuar avanzado en los temas tratados. A continuación se mencionan algunas líneas de investigación y tópicos para continuar con el trabajo propuesto:

Integración de aprendizaje profundo en SLAM probabilístico

El uso de redes neuronales profundas para la estimación de características y reducción de ruido en sensores es un área prometedora. Se propone investigar modelos híbridos que combinen técnicas clásicas de SLAM con enfoques basados en aprendizaje profundo, como Graph Neural Networks (GNNs) para mejorar la optimización de mapas y redes convolucionales para la estimación de movimiento.

SLAM distribuido y colaborativo

En entornos donde múltiples robots interactúan, el desarrollo de un SLAM distribuido con intercambio eficiente de datos y actualización colaborativa de mapas es crucial. Se investigará el uso de inferencia probabilística distribuida para optimizar la fusión de información y mejorar la robustez del sistema ante fallos de comunicación.

SLAM para entornos no convencionales

A pesar de los avances, la navegación en entornos extremos como el espacio, el fondo marino o infraestructuras subterráneas sigue siendo un desafío. Se propone el desarrollo de modelos probabilísticos adaptativos que puedan ajustarse dinámicamente a condiciones ambientales adversas mediante técnicas de inferencia bayesiana reforzada.

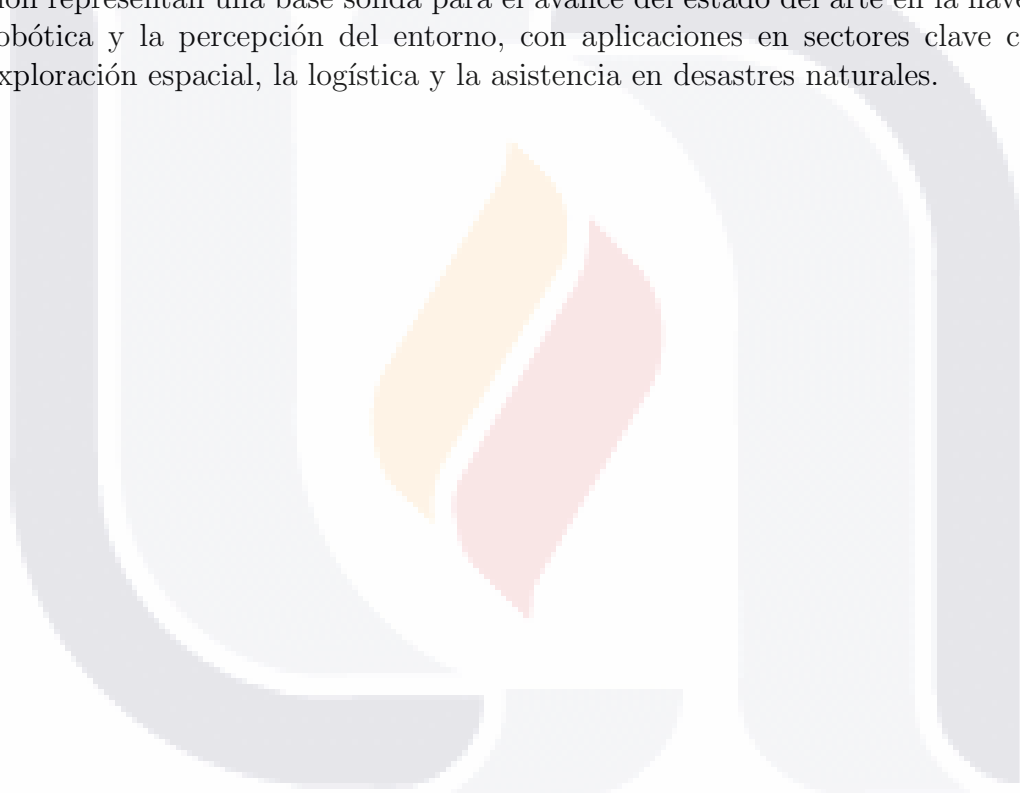
Modelado de incertidumbre en sensores no convencionales El uso de sensores alternativos como radares de onda milimétrica, sensores táctiles o espectroscopia para SLAM requiere un modelado probabilístico más preciso. Se explorará la implementación de filtros de partículas avanzados y redes bayesianas para mejorar la interpretación de datos en tiempo real.

Optimización computacional de algoritmos SLAM

Dado el alto costo computacional de los algoritmos de SLAM probabilístico, se

investigará el uso de técnicas de compresión de datos y algoritmos de baja latencia. En particular, se explorarán implementaciones en hardware especializado como GPUs y TPUs para acelerar la inferencia en sistemas embebidos de bajo consumo.

El futuro de la robótica probabilística y SLAM radica en la integración de metodologías interdisciplinarias que permitan superar las limitaciones actuales en términos de precisión, eficiencia computacional y robustez en entornos dinámicos. La combinación de modelos probabilísticos con aprendizaje automático, computación distribuida y nuevas técnicas de inferencia permitirá el desarrollo de robots más autónomos y capaces de operar en condiciones adversas. Estas líneas de investigación representan una base sólida para el avance del estado del arte en la navegación robótica y la percepción del entorno, con aplicaciones en sectores clave como la exploración espacial, la logística y la asistencia en desastres naturales.





Bibliografía

- [1] S Adarsh, S Mohamed Kaleemuddin, Dinesh Bose, and K I Ramachandran. Performance comparison of infrared and ultrasonic sensors for obstacles of different materials in vehicle/ robot navigation applications. *IOP Conference Series: Materials Science and Engineering*, 149(1):012141, sep 2016.
- [2] P. Blondel. *The Handbook of Sidescan Sonar*. Springer, 2009.
- [3] Wu Y.-Zeng W.-Du S. Chen, Y. Kinematics model estimation of 4w skid-steering mobile robots using visual terrain classification. *Hindawi Journal of Robotics*, 2023:207–221, 10 2023.
- [4] Chordify. Chordify application. <https://chordify.net/pages/about/>. Consultado Febrero 5, 2025.
- [5] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.
- [6] Johnstone-I. M. Donoho, D. L. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.
- [7] D. Esteban and C. Galand. Application of quadrature mirror filters to split band voice coding schemes. In *ICASSP '77. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 191–195, 1977.
- [8] A. Everest. *Master Handbook of Acoustics*. McGraw-Hill, 2001.
- [9] Markham A. He, Y. SoundSynp: Sound source detection from raw waveforms with multi-scale synperiodic filterbanks. In *International Conference on Artificial Intelligence and Statistics*, pages 9010–9023. PMLR, 2023.
- [10] Yuan He, Weiguo Wang, Luca Mottola, Shuai Li, Yimiao Sun, Jinming Li, Hua Jing, Ting Wang, and Yulei Wang. Acoustic localization system for precise drone landing. *IEEE Transactions on Mobile Computing*, 23(5):4126–4144, 2023.

- [11] Guerrero-Díaz-de-León-A. Macías-Díaz-J.E. Hernández, R. Artificial neural network-based positioning error modeling and compensation for low-cost encoders of four-wheeled vehicles. *Evolutionary Intelligence*, 2024.
- [12] Guerrero-Díaz-de-León J.A.-Macías-Díaz J.E. Hernández-Torres, R. A template-based algorithm by geometric means for the automatic and efficient recognition of music chords. *Evolutionary Intelligence*, 2024.
- [13] Guerrero-Díaz-de-León J.A.-Macías-Díaz J.E. Macias S. Hernández-Torres, R. A method for obstacle-range estimation based on echo-location using high-end sound with applications. *Evolutionary Intelligence*, 2025.
- [14] E. Herrera. *Teoria musical y Armonia moderna vol. I*. Antoni Bosch, 1990.
- [15] Y Huang, Z Li, and J Wang. Ann approach for scara robot inverse kinematics solutions with diverse datasets and optimisers. *Journal of Intelligent and Robotic Systems*, 2022.
- [16] Md Rashedul Islam, Jia Uddin, and Jong-Myon Kim. Acoustic emission sensor network based fault diagnosis of induction motors using a gabor filter and multiclass support vector machines. *Adhoc & Sensor Wireless Networks*, 34, 2016.
- [17] Renner Bernd-Christian Jiang, Y. Low-cost underwater swarm acoustic localization: a review. *IEEE access*, 12:25779–25796, 2024.
- [18] Krzysztof Kozłowski and Dariusz Pazderski. Modeling and control of a 4-wheel skid-steering mobile robot. *International Journal of Applied Mathematics and Computer Science*, 14, 01 2004.
- [19] P Kumar, S Sharma, and N Kumar. Artificial neural networks: a practical review of applications in robotics. *Journal of Intelligent Information Systems*, 2022.
- [20] Y Li, J Wang, and Z Li. A hybrid method using fabrik and custom ann in solving inverse kinematics of robotic manipulators. *International Journal of Machine Tools and Manufacture*, 2023.
- [21] X Liu, Z Li, and J Wang. Trajectory tracking solution of a robotic arm based on optimized ann. *IEEE Transactions on Industrial Informatics*, 2019.
- [22] Y Liu, M Li, and L Wang. Kinematic-icp: Enhancing lidar odometry with kinematic constraints for four-wheeled robots. *arXiv preprint arXiv:2410.10277*, 2024.

- [23] G. S. Mallat. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(7):674–693, 1989.
- [24] M. Müller. *Fundamentals of Music Processing*. Springer, Cham, 2015.
- [25] M. Müller, D. Ellis, A. Klapuri, and G. Richard. Signal processing for music analysis. *IEEE Journal of Selected Topics in Signal Processing*, 5:1088–1110, 2011.
- [26] Gupta N.-Reese-B.-et. al. Ordonez, C. Learning of skid-steered kinematic and dynamic models for motion planning. *Robotics and Autonomous Systems*, 95:207–221, 09 2017.
- [27] D. Raichel. *The Science and applications of Acoustics*. Springer, 2006.
- [28] Braun H. Riedmiller, M. A direct adaptive method for faster backpropagation learning: The rprop algorithm. *IEEE International Conference on Neural Networks*, 1993.
- [29] Ian S., Dan M., and Sumit B. Mysong: Automatic accompaniment generation for vocal melodies. In *CHI '08 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008.
- [30] R Sari, R Aryanto, and E Setiawan. Motion system of a four-wheeled robot using a pid controller based on inverse kinematics. *CSOL*, 2023.
- [31] Jens Schröder, Stefan Goetze, and Jörn Anemüller. Spectro-temporal gabor filterbank features for acoustic event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2198–2208, 2015.
- [32] Shazam. Shazam application. <https://www.shazam.com/es/apps>. Consultado Febrero 5, 2025.
- [33] P Sicard and R Gourdeau. Ann-based adaptive control of robotic manipulators with friction and joint elasticity. *IEEE Transactions on Robotics*, 2007.
- [34] S Singh, N Kumar, and S Sharma. A new artificial neural network approach in solving inverse kinematics problem of robotic arm. *Journal of Intelligent and Robotic Systems*, 2016.
- [35] S. Stevic. Geometric mean. In *International Encyclopedia of Statistical Science*. Springer, 2011.

- [36] R.L. STRATONOVICH. Application of the theory of markov processes to the optimum filtration of signals**radiotekh. i elektr., akad. nauk sssr., 5, no 11, 1751 (1960). (translated by o. m. blunn.). In P.I. Kuznetsov, R.L. Stratonovich, and V.I. Tikhonov, editors, *Non-Linear Transformations of Stochastic Processes*, pages 467–484. Pergamon, 1965.
- [37] Burgard W.-Fox-D. Thrun, S. *Probabilistic Robotics*. The MIT Press, 2006.
- [38] Leonard J. J. Thrun, S. *Simultaneous Localization and Mapping*. Springer Berlin Heidelberg, 2008.
- [39] N. Tomasini. *El fundamento matematico de la escala musical y sus raices pitagoricas*. Universidad de Palermo.
- [40] A Wicaksono, A Wahyudi, and E Setiawan. Implementation kinematics modeling and odometry of four omni wheel mobile robot on the trajectory planning and motion control based microcontroller. *ResearchGate*, 2023.
- [41] Y Zhang, Z Li, and J Wang. Artificial neural networks for inverse kinematics problem in robotic arm. *Journal of Robotics*, 2023.
- [42] Y Zhang, Z Li, and Y Wang. Kinematic analysis and odometry-based navigation of an omnidirectional wheeled mobile robot on uneven surfaces. *Springer*, 2023.