



UNIVERSIDAD DE GUANAJUATO

---

---

CAMPUS IRAPUATO - SALAMANCA  
DIVISIÓN DE INGENIERÍAS

*Diseño de algoritmos robustos UFIR-Kalman  
para localización de un robot holonómico  
usando tecnología RFID*

**TESIS**

QUE PARA OBTENER EL GRADO DE:

**DOCTOR EN INGENIERÍA ELÉCTRICA**

PRESENTA:

***M.I. Jorge Andrés Ortega Contreras***

DIRECTORES:

**Dr. Yuriy S. Shmaliy**  
**Dr. José A. Andrade Lucio**

Salamanca, Guanajuato

Diciembre 2023.

Salamanca, Gto., a 29 de noviembre del 2023

**M. en I. HERIBERTO GUTIÉRREZ MARTIN  
COORDINADOR DE ASUNTOS ESCOLARES  
P R E S E N T E.-**

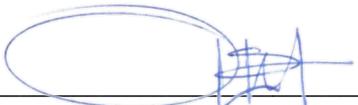
Por medio de la presente, se otorga autorización para proceder a los trámites de impresión, empastado de tesis y titulación al alumno(a) **Jorge Andrés Ortega Contreras** del **Programa de Doctorado en Ingeniería Eléctrica** y cuyo **NUA** es: 142451 del cual soy director. El título de la tesis es: **Diseño de algoritmos robustos UFIR-Kalman para localización de un robot holonómico usando tecnología RFID.**

Hago constar que he revisado dicho trabajo y he tenido comunicación con los sinodales asignados para la revisión de la tesis, por lo que no hay impedimento alguno para fijar la fecha de examen de titulación.

**A T E N T A M E N T E**

  
\_\_\_\_\_  
**DR. YURIY SHMALIY  
DIRECTOR DE TESIS  
SECRETARIO**

  
\_\_\_\_\_  
**DR. JOSÉ AMPARO ANDRADE LUCIO  
DIRECTOR DE TESIS**

  
\_\_\_\_\_  
**DR. OSCAR GERARDO IBARRA MANZANO  
PRESIDENTE**

*Granados Lieberman D.*  
\_\_\_\_\_  
**DR. DAVID GRANADOS LIEBERMAN  
VOCAL**

  
\_\_\_\_\_  
**DR. JUAN GABRIEL AVIÑA CERVANTES  
VOCAL**

  
\_\_\_\_\_  
**DR. ARTURO GARCÍA PÉREZ  
VOCAL**

# Agradecimientos

A mi madre, por su amor incondicional durante toda mi vida.

A mi padre, por haber sido y seguir siendo mi ejemplo a seguir.

A mis compañeros del grupo de investigación, Karen y Gabriel. Sin su talento y amistad este proceso no habría sido igual.

A mis amigos David, Francisco y Óscar por seguir esta etapa conmigo.

A Víctor, por su fé en mí.

A mis asesores, los doctores José A. Andrade Lucio y Yuriy S. Shmaliy por su paciencia y fuerte compromiso para llevar este proyecto hasta su fin.

A los Doctores J. Gabriel Aviña Cervantes, Edgar Alvarado por sus palabras de aliento e impulso a mi desarrollo profesional.

Al Dr. Sergio Ledesma, con su guía me ayudó a recuperar el camino.

A la Ing. Consuelo Galván Anguiano, porque desde joven me mostró el inicio del camino que yo elegí vivir.

# Agradecimientos Institucionales

Expreso mi gratitud hacia la Universidad de Guanajuato, especialmente a los profesores y personal de la División de Ingenierías del Campus Irapuato-Salamanca por la formación y el apoyo recibido. En particular, a todos los miembros del grupo de Procesamiento Digital de Señales de la División por su invaluable apoyo en los proyectos y procesos administrativos.



Este trabajo fue realizado gracias al apoyo recibido a través del Consejo Nacional de Humanidades, Ciencia y Tecnología CONAHCYT con la Convocatoria de Investigación Científica Básica 2017-2018, bajo el número de solicitud A1-S-10287.



**CONAHCYT**

CONSEJO NACIONAL DE HUMANIDADES  
CIENCIAS Y TECNOLOGÍAS

*Jorge Andrés Ortega Contreras*

# Resumen

Los robots móviles tienen la capacidad de moverse dentro de un entorno que puede o no ser conocido, existen diversas configuraciones cinemáticas que le ofrecen ventajas a unos sobre otros. En general, los robots que no poseen restricciones holonómicas son capaces de cambiar su postura en cualquier instante de tiempo, esto les concede una mayor maniobrabilidad y les ha abierto las puertas como plataformas robóticas omnidireccionales en múltiples aplicaciones. Una de las tareas más relevantes para robots móviles es la localización, con el fin de cumplir cualquiera de las actividades que se le asignen el robot debe poder percibir su entorno y saber donde se encuentra para poder moverse a su destino. En este trabajo de tesis se estudiarán tecnologías de radiofrecuencia empleadas como medio de observación a fin de localizar un robot holonómico dentro de un espacio desconocido empleando elementos de referencia con posición fija y conocida. Se presentará el análisis para la autolocalización del robot omnidireccional de tres ruedas (TWOR) en entornos de etiquetas de identificación por radiofrecuencia (RFID). El modelo TWOR no lineal se mejora significativamente mediante el uso de interpretación geométrica y representación de tiempo incremental en tiempo discreto. Este tipo de robot puede realizar rotación y traslación en cualquier postura. El segmento de longitud de arco descrito por las omniruedas controla la trayectoria del robot. Presentamos un modelo cinemático simplificado en espacio de estados. La posición y el rumbo de TWOR se estiman mediante mediciones de distancia a etiquetas RFID y un giroscopio digital en presencia del típico ruido de medición de color (CMN). Además se describe el desarrollo de un sistema de seguimiento para un robot móvil omnidireccional de tres ruedas cuyo sistema de observación se basa en un sensor de alcance de banda ultraancha.

Las tecnologías inalámbricas como los sistemas basados en etiquetas de identificación por radiofrecuencia (RFID) han atraído el interés de muchos consumidores debido a su bajo costo y bajo (o nulo) consumo de energía y un amplio rango de distancia que los convirtió en estándar para la navegación y seguimiento de objetos en interiores. En diseños prácticos de sistemas de navegación de robots móviles, se encuentran varias soluciones híbridas eficientes, como el esquema de localización que combina la información disponible de las redes basadas en etiquetas RFID con mediciones basadas en láser. Modelos RFID de potencia variable son empleados para la localización sobre

---

redes de etiquetas pasivas de frecuencia ultra-alta (UHF) en entornos complejos.

Varios factores pueden afectar la precisión de la navegación del robot, como las no linealidades en sus componentes, las perturbaciones, el ruido ambiental y los sensores. De esta manera, se han orientado muchos esfuerzos a minimizar los errores aleatorios y las incertidumbres del modelo. La odometría es un método utilizado para una fácil implementación en hardware y software, y son de costo relativamente bajo. Pero este método podría acumular considerables derivas en la estimación. Por ellos, se propone un enfoque novedoso para la predicción de la respuesta de impulso finito  $H_2$ -FIR en espacio-estado de tiempo discreto. Se ha demostrado que el predictor FIR óptimo  $H_2$  con sesgo y el  $H_2$ -OUFIR restringido en sesgo derivado bajo perturbaciones y errores de medición tiene la máxima verosimilitud y es equivalente al predictor OUFIR bajo ruido gaussiano. Este algoritmo se desarrolló utilizando el método de Euler hacia atrás minimizando la norma de Frobenius ponderada al cuadrado. También se diseña un algoritmo de filtrado subóptimo  $H_2$ -FIR con restricción de sesgo que utiliza la desigualdad de matriz lineal. El rendimiento del predictor  $H_2$ -OUFIR se investiga mediante simulaciones y experimentalmente en comparación con los predictores de Kalman.

En entornos hostiles, el seguimiento se organiza asumiendo perturbaciones, errores iniciales y errores de datos que requieren algoritmos sólidos. En este artículo, desarrollamos, en un espacio de estados de tiempo discreto, un filtro robusto de respuesta de impulso finito óptimo a posteriori  $H_2$ -OFIR de sistemas perturbados bajo errores iniciales y de medición. La derivación se proporciona utilizando un nuevo enfoque de estimación del estado de respuesta de impulso finito  $H_2$ -FIR minimizando la norma de Frobenius al cuadrado de la función de transferencia ponderada. El filtro robusto  $H_2$ -OFIR está diseñado para matrices de error de bloque completo y sus formas recursivas se muestran para matrices de error diagonales. También se presenta el algoritmo de filtrado subóptimo  $H_2$ -FIR utilizando la desigualdad de matriz lineal. Se muestra que, en el seguimiento de vehículos en movimiento basado en el sistema de posicionamiento global, el filtro  $H_2$ -OFIR supera a los filtros Kalman y FIR imparcial en términos de precisión y robustez. Además, la capacidad de operar en horizontes cortos hace que el filtro  $H_2$ -OFIR sea computacionalmente eficiente.

# Abstract

Mobile robots can move within an environment that may or may not be familiar, there are various cinematographic settings that offer advantages to some over others. Robots that do not have holonomic restrictions can change their posture at any instant of time, this gives them greater maneuverability and has opened the doors to them as omnidirectional robotic platforms in multiple applications. One of the most relevant tasks for mobile robots is localization. In order to carry out any of the activities assigned to the robot, it must be able to perceive its environment and know where it is in order to move to its destination. In this thesis work, we will study radiofrequency technologies used as a means of observation in order to locate a holonomic robot within an unknown space using reference elements with a fixed and known position. Analysis for self-localization of three-wheeled omnidirectional robot (TWOR) in radio frequency identification (RFID) tag environments will be presented. It significantly improved the nonlinear TWOR model by using geometric interpretation and discrete-time incremental time representation. This type of robot can perform rotation and translation in any posture. The arc length segment described by the omni wheels controls the trajectory of the robot. We present a simplified cinematographic model in state space. We estimate the position and heading of TWOR using distance measurements to RFID tags and a digital gyroscope in the presence of typical color measurement noise (CMN). The development of a tracking system for a three-wheeled omnidirectional mobile robot whose observation system is based on an ultra-wideband range sensor is described.

Wireless technologies such as systems based on radio frequency identification (RFID) tags have attracted the interest of many consumers due to their low cost and low (or no) power consumption and a wide range of distance that make them standard for Indoor object navigation and tracking. In practical designs of mobile robot navigation systems, they find several efficient hybrid solutions, such as the localization scheme that combines the information available from RFID tag-based networks with laser-based measurements. Variable power RFID models are used for localization over ultra-high frequency (UHF) passive tag networks in complex environments.

Several factors can affect the robot's navigation accuracy, such as

---

nonlinearities in its components, disturbances, environmental noise, and sensors. In this way, many efforts have been directed to minimizing random errors and model uncertainties. Odometry is a method used for easy implementation in hardware and software, and its cost is relatively low. But this method could accumulate considerable deviations in the estimate. Therefore, a novel approach is proposed for the prediction of the finite impulse response  $H_2$ -FIR in discrete-time state space. The optimal  $H_2$  biased and constrained unbiased FIR predictor  $H_2$ -OUFIR derived under perturbations and measurement errors has been shown to have maximum likelihood form and be equivalent to the OUFIR predictor under Gaussian noise. This algorithm was developed using the backward Euler method by minimizing the squared weighted Frobenius norm. A bias-constrained suboptimal  $H_2$ -FIR filtering algorithm using linear matrix inequality is also designed. The performance of the  $H_2$ -OUFIR predictor is investigated through simulations and experimentally in comparison with Kalman and unbiased predictors.

In hostile environments, monitoring is organized assuming disturbances, initial errors, and data errors that require robust algorithms. In this paper, we develop, in a discrete-time state space, a robust  $H_2$  a posteriori optimal finite impulse response  $H_2$ -OFIR filter of perturbed systems under initial and measurement errors. The derivation uses a new approach to estimate the  $H_2$  finite impulse response  $H_2$ -FIR state by minimizing the Frobenius norm squared of the weighted transfer function. The robust  $H_2$ -OFIR filter is designed for full block error matrices and its recursive forms are shown for diagonal error matrices. The suboptimal filtering algorithm  $H_2$ -FIR using linear matrix inequality is also presented. It is shown that in global positioning system-based tracking of moving vehicles, the  $H_2$ -OFIR filter outperforms the Kalman and unbiased FIR filters in terms of accuracy and robustness. Additionally, the ability to operate at short horizons makes the  $H_2$ -OFIR filter computationally efficient.

**Keywords:**  $H_2$ -OUFIR,  $H_2$ -OFIR, Kalman filter,  $H_2$ -FIR, colored noise, omnidirectional robot.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Justificación . . . . .	1
1.2. Motivación y descripción del problema . . . . .	2
1.2.1. Hipótesis de Trabajo . . . . .	3
1.3. Objetivos . . . . .	3
<b>2. Fundamento Teórico</b>	<b>5</b>
2.1. Estimación, Suavizado y Predicción . . . . .	5
2.2. Filtro Kalman . . . . .	6
2.3. Ruido . . . . .	7
2.4. Teoría de estimación Robusta . . . . .	10
2.5. Localización de un Robot . . . . .	11
2.6. Sensores RFID y UWB . . . . .	12
<b>3. Predictor <math>H_2</math>-FIR bajo perturbaciones y errores de medición</b>	<b>16</b>
3.1. Resumen . . . . .	16
3.2. Modelo y Formulación del problema . . . . .	17
3.3. Predictor $H_2$ -OUFIR en la forma de máxima verosimilitud . . . . .	22

3.4. Equivalencia con el predictor OUFIR . . . . .	24
3.5. Predictor subóptimo $H_2$ -FIR usando LMI . . . . .	25
<b>4. Filtrado Robusto <math>H_2</math>-OFIR</b>	<b>26</b>
4.1. Resumen . . . . .	26
4.2. Modelo y Formulación del problema . . . . .	28
4.3. Filtrado $H_2$ -FIR estándar . . . . .	29
4.4. Filtrado Robusto $H_2$ -OFIR . . . . .	33
4.4.1. Filtrado Robusto $H_2$ OFIR en batch . . . . .	33
4.4.2. Forma recursiva para $\mathcal{Q}_N$ y $\mathcal{R}_N$ diagonales . . . . .	36
4.5. Filtro subóptimo $H_2$ -FIR usando LMI . . . . .	38
<b>5. Implementación de un Robot Omnidireccional de Tres Ruedas</b>	<b>41</b>
5.1. Diseño y planificación . . . . .	42
5.2. Componentes de Hardware . . . . .	45
5.2.1. NVIDIA Jetson TX2 . . . . .	46
5.2.2. RealSense Camera . . . . .	48
5.2.3. TETRIX® MAX DC Motor Expansion Controller . . . . .	48
5.2.4. Motores DC Torquenado . . . . .	49
5.2.5. Interfaz NXT-Serial . . . . .	50
5.2.6. DWM1000 . . . . .	51
5.2.7. Lidar . . . . .	52
5.3. Programación y software . . . . .	54
5.3.1. Computation Graph level . . . . .	54
5.3.2. Tópicos . . . . .	56

<b>6. Localización del robot omnidireccional en una red RFID</b>	<b>62</b>
6.1. Modelado cinemático del robot . . . . .	64
6.2. Representación en espacio de estados . . . . .	65
6.2.1. Ecuación de Transición de Estado . . . . .	65
6.2.2. Ecuación de Observación . . . . .	67
6.3. Filtros extendidos para ruido coloreado . . . . .	69
6.3.1. Modelo Extendido del espacio de estados . . . . .	69
6.3.2. Modificación del sistema para Ruido Coloreado . . . . .	69
6.3.3. Filtro de Kalman Extendido para ruido coloreado . . . . .	70
6.3.4. Filtro FIR Extendido para ruido coloreado . . . . .	71
<b>7. Localización del TWOR en una red de sensores UWB</b>	<b>74</b>
7.1. Descripción del robot móvil . . . . .	75
7.1.1. Modelo de Observación . . . . .	77
7.2. Seguimiento del TWOR . . . . .	78
7.2.1. Caracterización del sensor UWB en LOS . . . . .	78
7.2.2. Localización del Tag en la red UWB . . . . .	80
<b>8. Resultados</b>	<b>83</b>
8.1. Predictor H2-FIR . . . . .	83
8.2. Estimador H2-OFIR . . . . .	86
8.2.1. Seguimiento entre vehículos . . . . .	87
8.3. AutoLocalización en un entorno con RFID-tags . . . . .	96
8.3.1. Trayectoria circular en un ambiente interior . . . . .	96
8.3.2. Simulación con datos experimentales . . . . .	99

8.3.3. Consumo de tiempo y Complejidad Computacional . . . . .	101
8.4. Localización del TWOR en una red UWB . . . . .	104
<b>9. Conclusiones</b>	<b>106</b>
<b>Bibliografía</b>	<b>107</b>
<b>A. Algoritmos</b>	<b>119</b>
<b>B. Códigos en Matlab</b>	<b>122</b>
<b>C. Producción Académica</b>	<b>126</b>
C.1. Artículos en revista . . . . .	126
C.2. Artículos en Congreso . . . . .	127
C.3. Capítulos en Libro . . . . .	129
C.4. Producción de Capital Humano . . . . .	129

# Índice de figuras

2.1. Etapas en el filtro de Kalman . . . . .	7
2.2. Ejemplos de ruido coloreado . . . . .	9
2.3. Localización simultánea y mapeo . . . . .	13
2.4. Medición de Rango de dos vías, adaptado de Lian [1] . . . . .	15
4.1. Estructura general de un filtro robusto $H_2$ -OFIR . . . . .	35
5.1. Ruedas omnidireccionales . . . . .	42
5.2. Ejes de los rodamientos y de la rueda omnidireccional . . . . .	43
5.3. Plataformas Omnidireccionales, a) De tres ruedas, b) De cuatro ruedas. En rojo la dirección activa de la rueda, en azul la pasiva. . . . .	44
5.4. . . . .	46
5.5. Tarjeta de desarrollo NVIDIA Jetson TX2 . . . . .	47
5.6. Intel Realsense T265 . . . . .	48
5.7. . . . .	49
5.8. Motor DC Torquenado . . . . .	50
5.9. Arduino Mega2560 en la breakoutboard MEGAPRO . . . . .	51
5.10. Módulo BU01-DB de AiThinker . . . . .	52
5.11. Sensor de distancia láser VL53L0X . . . . .	53

5.12. Conector J21 en la Jetson TX2 . . . . .	53
5.13. Representación de los niveles de cómputo del grafo . . . . .	55
5.14. Ventana de aplicaciones de inicio . . . . .	56
5.15. Salida en terminal del launchfile para la T265 . . . . .	57
5.16. Salida en terminal del nodo para el sensor VL53L0x . . . . .	58
5.17. Salida en terminal de roserial . . . . .	60
5.18. Salida en terminal del nodo para el transceptor BU01 . . . . .	61
6.1. Plataforma de desarrollo para robótica móvil Festo Robotino [2]. . . . .	64
6.2. Mobile Robot . . . . .	65
7.1. Mobile Robot . . . . .	78
7.2. Robot Móvil . . . . .	79
7.3. Covarianza del UWB . . . . .	80
8.1. RMSEs producidos por los predictores H2-OUFIR, UFIR y Kalman como función de $\phi$ bajo perturbaciones de Markov-Gauss. . . . .	85
8.2. Ruido de medición no-Gaussiano . . . . .	85
8.3. Errores de estimación producidos por los algoritmos $H_2$ -OUFIR, UFIR y Kalman . . . . .	86
8.4. Seguimiento de un vehículo frontal por un vehículo trasero . . . . .	87
8.5. Ruido de medición de radar $\xi_k$ , m. (a) Casi WGN con $\varphi = 0.05$ . (b) Fuertemente coloreado con $\varphi = 0.95$ . . . . .	87
8.6. Respuesta al escalón del los filtros en la coordenada $y$ para dos valores extremos de factor de color (a) $\varphi = 0.05$ . (b) $\varphi = 0.95$ . . . . .	90
8.7. Respuesta al impulso del los filtros en la coordenada $y$ para dos valores extremos de factor de color (a) $\varphi = 0.05$ . (b) $\varphi = 0.95$ . . . . .	91

8.8. Trayectoria medida por el GPS para un vehículo realizando una maniobra suave . . . . .	92
8.9. Valores típicos de RMSE producidos por los filtros $H_2$ -OFIR por lotes, Kalman, y UFIR a lo largo de la dirección en x como función de $N$ para maniobras suaves considerando un vehículo bajo ruido moderadamente coloreado $\varphi = 0.5$ . . . . .	93
8.10. Trayectoria medida por el GPS para un vehículo realizando una maniobras rápidas . . . . .	94
8.11. Valores típicos de RMSEs producidos por los filtros $H_2$ -OFIR por lotes, Kalman, y UFIR a lo largo de la dirección en x como función de $N$ para maniobras rápidas considerando un vehículo bajo ruido moderadamente coloreado $\varphi = 0.5$ . . . . .	95
8.12. a) Ambiente interior con etiquetas RFID y trayectoria de prueba del TWOR. b) TWOR y los dispositivos asociados . . . . .	97
8.13. RMSE producido por los algoritmos cEKF, EFIR y cEFIR . . . . .	99
8.14. Trayectoria estimada del robot y medición de referencia. . . . .	100
8.15. RMSE producido por los algoritmos cEKF, EFIR y cEFIR . . . . .	101
8.16. Consumo en tiempo de los algoritmos . . . . .	103

# Índice de tablas

5.1. Lista de materiales . . . . .	45
8.1. RMSEs bajo perturbaciones de Markov-Gauss. . . . .	84
8.2. RMSEs (m) Producido por los filtros FIR para $N_{opt} = 152; 400$ . . . .	98
8.3. RMSEs (m) Producido por el algoritmo cEKF para $\Psi_{opt} = 0.05; 0.95$ .	99
8.4. RMSEs (m) Producido por los filtros FIR para $N_{opt} = 82; 245$ . . . . .	100
8.5. RMSEs (m) Producido por el algoritmo cEKF para $\Psi_{opt} = 0.05; 0.95$ .	101
8.6. Número de FLOPs para los algoritmos cEKF y cEFIR . . . . .	102



# Capítulo 1

## Introducción

### 1.1. Justificación

La navegación en interiores es un problema que ha adquirido importancia, a la fecha, para ubicar a un objeto en un mapa interior se pueden emplear tecnologías como: Redes WiFi, estaciones bluetooth, estaciones de telefonía móvil y luces LED. En su conjunto forman una red de información. Cualquier tecnología de ubicación en interiores depende de este conjunto de "balizas" que tienen una posición fija, empleando una metodología de medición [3, 4] es posible estimar la posición dentro de la red.

La tecnología RFID (Radio-Frequency IDentification) se basa en la utilización de etiquetas (tag) que integran una antena y un chip. Las más comunes son las "etiquetas pasivas", que se alimentan mediante las ondas electromagnéticas emitidas por un receptor, que estimula el circuito devolviendo los datos almacenados en el chip. Debido su consumo energético tienen ventaja sobre el resto de las tecnologías antes mencionadas, y actualmente son ampliamente usadas para ubicación en interiores [5].

Existen diversos factores que influyen en el desempeño de una red RFID [6], según R. Aliberti [7] el ruido del sistema de observación a través de tag RFID no es blanco, por ello un filtro de Kalman pierde precisión, la solución a través de un filtro de partículas puede lograr mejores resultados a un mayor costo computacional.

La estimación del estado es un problema que puede ser muy complicado por la no linealidad del modelo en espacio de estados para sistemas que se pueden encontrar en la vida diaria, además de la presencia de ruido de observación y del proceso, con estadísticas desconocidas. Por estos motivos es importante atender el problema de localización en interiores mediante una variación de los algoritmos UFIR/Kalman. El sistema de localización mediante UHF-RFID que se propone utilizará el RSSI y

lateración hiperbólica [8] para la construcción de la función de observación, se mejorará su desempeño al corregir el efecto de pérdidas por polarización debido a la orientación del sensor RFID respecto a cada tag, adicionalmente se modelará el ruido de observación mediante un proceso estacionario de Markov-Gauss.

## 1.2. Motivación y descripción del problema

En los artículos “An Electrically Small Spherical UHF RFID Tag Antenna With Quasi-Isotropic Patterns for Wireless Sensor Networks” [9] y “Design of semi-active RFID antenna” [10] Se discute el diseño de antenas UHF (890-960 MHz) para RFID que logran disminuir el espacio necesario, además la búsqueda de patrones de radiación isotrópicos incrementan la detectabilidad de la posición del tag independientemente de su posición con respecto al lector.

En “Constrained and quantized Kalman filtering for an RFID robot localization problem” [11] se considera un robot moviéndose en un ambiente conocido. Dicho ambiente está instrumentado con una serie de tag RFID distribuidos, que en combinación con la odometría del robot forman la red de información con la cual se estima la posición. El ruido de un sistema de este tipo es altamente no gaussiano. Este artículo presenta una variación del algoritmo de Kalman que alcanza un rendimiento comparable a los resultados obtenidos mediante un filtro de partículas con menor costo computacional.

En el artículo “Accurate Self-Localization in RFID Tag Information Grids Using FIR Filtering” [12], se propone un nuevo algoritmo de filtrado con respuesta finita al impulso, se verifica que dicha metodología tiene ventajas sobre el algoritmo extendido de Kalman (EKF) al no requerir conocimiento a priori de las estadísticas de ruido del sistema.

En “Self-Localization in RFID-Tag Information Networks Using Extended UFIR Filtering” [13] se comprueba que el filtro extendido con respuesta infinita al impulso (EUFIR) produce una estimación sin bias, a la vez que ignora las características del ruido del sistema, se prueba su rendimiento contra el filtro EKF y se verifica que el EUFIR tiene un mejor rendimiento.

En el artículo “Triangulation-based indoor robot localization using extended FIR/Kalman filtering” [14] se propone una combinación de un filtro EFIR y Kalman para la localización de un robot móvil a través de triangulación, si bien se ha probado la distintiva ventaja del algoritmo EFIR ante las características del ruido, a su vez requiere de un intervalo óptimo de promediado. Se muestra que el algoritmo EFIR/Kalman es más preciso que el EKF bajo ciertas características de ruido y condiciones iniciales.

En "Extended UFIR filtering of nonlinear models corrupted by white gaussian" [15] se examina el comportamiento de un filtro EFIR para un modelo en espacio de estados discreto no lineal y corrompido por ruido aditivo Gaussiano. El algoritmo es presentado en la forma de Kalman, pero ignorando las estadísticas del ruido, estas con calculadas por un promediado de N puntos. Se comparan dos filtros EFIR1 y EFIR2 de primer y segundo orden respectivamente con un filtro extendido de Kalman EKF. Se comprueba que ambos algoritmos producen errores similares bajo condiciones ideales.

El filtro de Kalman es ampliamente usado para la estimación del estado para sistemas lineales bajo ruido Gaussiano, en "State estimation under non-Gaussian Levy noise: A modified Kalman filtering method" [16] se discute como en presencia de ruido de Levy, el filtro de Kalman pierde efectividad, y se propone una modificación del algoritmo que trabaja para ruido con covarianza infinita.

Un vehículo que no tiene restricciones no holonómicas es capaz de moverse en cualquier dirección para cualquier orientación en el artículo "Design of an omnidirectional universal mobile platform" [17] se desarrolla la dinámica para un robot de este tipo.

### 1.2.1. Hipótesis de Trabajo

Es posible encontrar una mejora para los algoritmos UFIR/Kalman para minimizar el error en presencia de ruido de observación no gaussiano. Las pérdidas en el RSSI causadas por el mal acoplamiento por la polarización propia de la antena del tag pueden ser corregidas mediante el conocimiento a priori de la posición de los tags y la información obtenida por el IMU y/o mediante el uso de la antena correcta.

## 1.3. Objetivos

### Objetivo General

Implementar una mejora de los algoritmos de UFIR/Kalman para ruido de observación coloreado y someterlo a prueba para la localización de un robot holonómico en una red de información UHF-RFID.

### Objetivos Particulares

- Desarrollar un algoritmo de posicionamiento basados en el Radio Signal Strength Indicator (RSSI) y lateración hiperbólica.
- Modelar a partir de datos experimentales el ruido de observación del sistema como un proceso de Markov-Gauss.
- Construir un robot omnidireccional que incluya un lector de tag RFID, una unidad de medición inercial (IMU, Inertial Measurement Unit) y conectividad inalámbrica (WiFi/ Zigbee).
- Proponer una mejora a los algoritmos UFIR/Kalman para un ruido coloreado en el proceso de observación.

# Capítulo 2

## Fundamento Teórico

### 2.1. Estimación, Suavizado y Predicción

Estimación, predicción y suavizado son tres conceptos fundamentales en el análisis de series temporales y el procesamiento de señales, y están relacionados con la obtención de información a partir de datos secuenciales. El problema que discutiremos se puede formular de la siguiente manera: Considere dos procesos estocásticos reales  $\{s(t), t \in T\}$  y  $\{n(t), t \in T\}$  que corresponden a la señal y el ruido respectivamente. Supongamos que la suma  $y(t) = s(t) + n(t)$  es observable y medible. En el momento  $t$  tenemos una realización  $y(\tau), t_0 \leq \tau \leq t$ , de la variable medida. La mejor estimación del valor de la señal en el tiempo  $t_1 < t$ , resulta en el problema de suavizado o interpolación. Si  $t_1 = t$ , tenemos un problema de filtrado, y si  $t_1 > t$ , es un problema de predicción. En una formulación un poco más general, se requiere estimar una funcional de la señal como  $ds/dt$  o  $\int s dt$  a partir de observaciones de la señal además de ruido. A menudo es conveniente tener una notación para el conjunto de observaciones que están disponibles para estimar  $s(t_1)$ . El vector  $y$  está definido como:

$$y = [y^T(t_0), y^T(t_0 + 1), \dots, y^T(t)]^T \quad (2.1)$$

Para procesos de tiempo continuo,  $y$  es una función definida en el intervalo  $(t_0, t)$ . Sean  $y \in \mathbf{Y}$  y  $s \in \mathbf{S}$  un estimador (interpolador, filtro, predictor) que es una función que asigna  $\mathbf{Y}$  a  $\mathbf{S}$ . El valor de esta función para un valor particular de  $y$  se llama estimación y se denota por  $\hat{s}$ . Para describir completamente cualquiera de estos problemas debemos especificar lo siguiente:

- El criterio que define la mejor estimación
- Los procesos de señal y ruido

- Las restricciones sobre los estimadores admisibles

Los procesos de señal y ruido pueden caracterizarse mediante funciones de covarianza, densidades espectrales, diferencias estocásticas o ecuaciones diferenciales. También existen muchas posibilidades para definir la "mejor" estimación, por ejemplo, es posible definir una función de pérdida  $l$ , que es una función real con las propiedades  $l \geq 0$ ,  $l(x) = l(-x)$  y  $l$  no decrecientes para  $x > 0$ . La pérdida es entonces la variable estocástica  $l(s - \hat{s})$  y la mejor estimación  $\hat{s}$  es la que minimice la pérdida promedio  $E[l(s - \hat{s})]$ .

La estimación se refiere al proceso de determinar el valor o los parámetros de una variable o proceso desconocido a partir de un conjunto de observaciones o datos. En el contexto de series temporales, la estimación implica calcular valores o parámetros basados en los datos observados hasta el momento sin necesidad de hacer predicciones sobre valores futuros. La predicción se refiere al proceso de hacer suposiciones sobre los valores futuros de una variable o proceso basándose en datos pasados y presentes. En el contexto de series temporales, la predicción implica utilizar un modelo matemático o estadístico para prever cómo se comportará la serie temporal en el futuro. El suavizado se refiere al proceso de eliminar o reducir el ruido o la variabilidad en una serie temporal o señal con el fin de identificar patrones subyacentes o tendencias mediante la estimación de una función de densidad de probabilidad desconocida, de forma recursiva, y a lo largo del tiempo utilizando mediciones entrantes incrementales. Algunas de las técnicas de suavizado más comunes son: el suavizado promedio móvil, el suavizado exponencial, el suavizado de Kernel, el Rauch-Tung-Striebel (RTS), los suavizadores gaussianos, (suavizador de Kalman extendido o suavizadores de punto sigma) o los suavizadores de partículas.

## 2.2. Filtro Kalman

El filtro Kalman fue concebido por Rudolf E. Kalman en la década de 1960 en su publicación "A new approach to linear filtering and prediction problems" y es usado en numerosas áreas como navegación, robótica, control de procesos, fusión de datos de sensores y procesamiento de señales. La eficiencia del filtro de Kalman se debe a sus bajos requisitos computacionales, sus propiedades recursivas bien diseñadas, la representación del estimador óptimo para sistemas lineales unidimensionales asumiendo estadísticas de error gaussiano y su idoneidad para la implementación en tiempo real. El filtro de Kalman puede considerarse una extensión del desarrollo original de mínimos cuadrados de Gauss para estimar parámetros desconocidos de un modelo. Hasta mediados del siglo XX, los sistemas eran generalmente estáticos y las mediciones eran constantes con respecto al tiempo. Sin embargo, actualmente, los sistemas pueden ser dinámicos con mediciones realizadas desde plataformas móviles en intervalos de

tiempo constantes, como en la navegación autónoma. En consecuencia, el filtro de Kalman es el algoritmo más adecuado para aplicar al proceso de estimación vinculando mediciones en tiempo real y estimando el estado de los parámetros del sistema a través de series de tiempo.

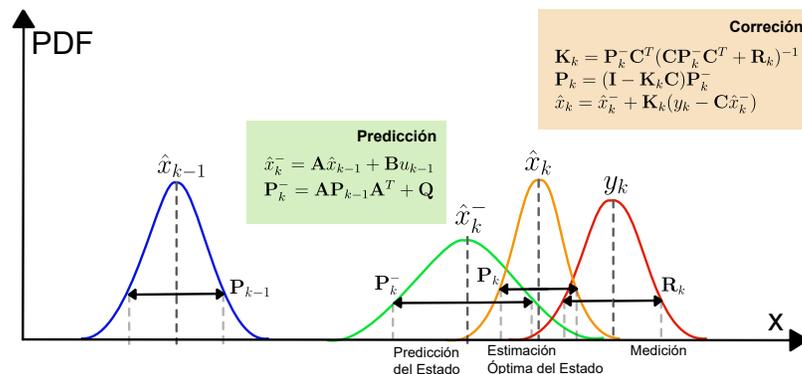


Figura 2.1. Etapas en el filtro de Kalman

El filtro Kalman está diseñado para sistemas lineales, pero los problemas científicos desafiantes, por ejemplo en la navegación por satélite, no son lineales y, por lo tanto, fue necesario implementar una versión especial del filtro Kalman llamada Filtro Kalman extendido (EKF). El filtro Kalman se basa en conceptos de probabilidad y tiene dos etapas, la predicción, donde utiliza el modelo matemático del sistema para anticipar cuál será el estado del sistema en el siguiente intervalo de tiempo, considerando el ruido de proceso  $\mathbf{Q}$ ; y la actualización o corrección, en la que se calcula una corrección que combina la predicción previa con las nuevas mediciones. Esto resulta en una estimación actualizada y más precisa del estado del sistema. La medición tienen una precisión limitada, por lo que se introduce una matriz de covarianza,  $\mathbf{R}$ . Un filtro de Kalman pierde eficiencia en situaciones en las que sus suposiciones fundamentales ya no se cumplen o están gravemente comprometidas. Las principales condiciones bajo las cuales un filtro de Kalman puede perder eficiencia son modelo Incorrecto, errores de medición grandes, alta no linealidad, falta de datos relevantes, cambios en la dinámica del sistema y sesgo en las mediciones o en el modelo.

## 2.3. Ruido

Se denomina ruido a las perturbaciones no deseadas o aleatorias que se superponen a una señal o proceso. Estas perturbaciones pueden ser causadas por una variedad de fuentes, como interferencia electromagnética, errores de medición, fluctuaciones térmicas, componentes electrónicos imperfectos o cualquier otro factor que

introduzca variabilidad no deseada en una señal o sistema. El ruido en señales y sistemas se considera una señal no deseada que puede degradar la calidad de la información que se transmite o procesa. El procesamiento de señales y sistemas a menudo incluye técnicas para filtrar, atenuar o eliminar el ruido, así como para modelar y comprender sus efectos en las señales.

El ruido coloreado, también conocido como ruido de color o ruido no blanco, es un tipo de señal aleatoria en la que la densidad espectral de potencia (PSD, por sus siglas en inglés) no es constante en todas las frecuencias, a diferencia del ruido blanco, que tiene una PSD constante en todas las frecuencias. En otras palabras, el ruido coloreado exhibe una distribución de energía que varía en función de la frecuencia.

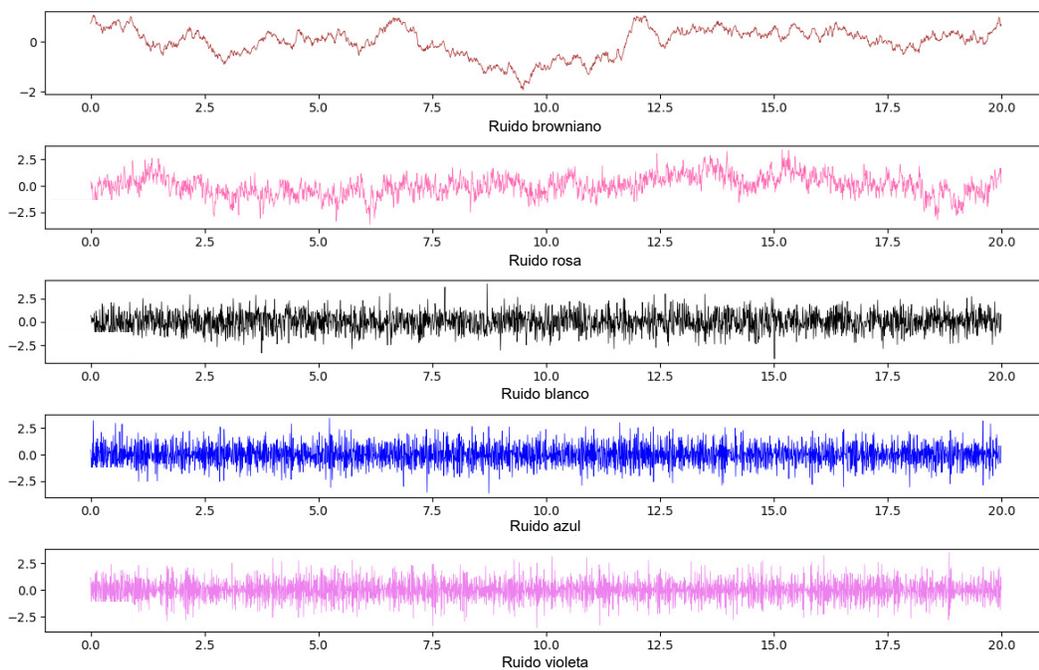
El término "coloreado" se deriva de la idea de que el ruido de este tipo parece tener un color característico en un espectrograma o una representación de frecuencia, en contraste con el ruido blanco, que no muestra un patrón de color en tales representaciones.

El ruido coloreado puede manifestarse en varias formas dependiendo de su PSD en función de la frecuencia. Es posible encontrarlo en diversas áreas, como la electrónica, la acústica, la señalización de comunicaciones y el procesamiento de señales. Se utiliza en pruebas de equipos, en la generación de señales de audio y en otras aplicaciones donde se necesita un espectro de frecuencia específico o un comportamiento de ruido particular. La elección del tipo de ruido coloreado depende de la aplicación y de los requisitos específicos de la señal. El proceso de "decolorar" el ruido, en el contexto de procesamiento de señales o análisis de datos, generalmente se refiere a la eliminación o reducción de la componente de ruido presente en una señal o conjunto de datos. La decoloración del ruido es esencial para mejorar la calidad de la señal o los datos y facilitar un análisis más preciso. Algunos ejemplos comunes de ruido coloreado incluyen:

- Ruido Rosa: Este tiene una PSD que disminuye proporcionalmente con la frecuencia, lo que significa que contiene más energía en las bajas frecuencias que en las altas. Tiene igual poder en bandas que son proporcionalmente amplias. Esto significa que el ruido rosa tendría igual potencia en el rango de frecuencia de 40 a 60 Hz que en la banda de 4000 a 6000 Hz. La densidad de potencia espectral, en comparación con el ruido blanco, disminuye 3,01 dB por octava. Por esta razón, el ruido rosa a menudo se denomina "ruido  $1/f$ ".
- Ruido Rojo: También es conocido como ruido Browniano. El ruido rojo tiene una PSD que aumenta con la frecuencia, lo que significa que contiene más energía en las altas frecuencias que en las bajas. Tiene una densidad de potencia que disminuye 6,02 dB por octava al aumentar la frecuencia (densidad de potencia proporcional a  $1/f^2$ ) en un rango de frecuencia excluido el cero. E
- Ruido Azul: El ruido azul tiene una PSD que aumenta a medida que aumenta la

frecuencia. Contiene más energía en las altas frecuencias y menos en las bajas. El ruido azul también se llama ruido azul. La densidad de potencia del ruido azul aumenta  $10\log_{10}2 = 3.01\text{dB}$  por octava con frecuencia creciente (densidad proporcional a  $f$ ) en un rango de frecuencia finito.

- Ruido Violeta: El ruido violeta tiene una PSD que aumenta proporcionalmente al cuadrado de la frecuencia. La densidad de potencia del ruido violeta aumenta 6,02 dB por octava al aumentar la frecuencia. Es menos común y se utiliza en aplicaciones específicas.



**Figura 2.2.** Ejemplos de ruido coloreado

## 2.4. Teoría de estimación Robusta

La teoría de estimación robusta es un enfoque que se centra en la obtención de estimaciones de un modelo estadístico que sean resistentes (robustas) a la presencia de valores atípicos o errores en los datos. Los valores atípicos son observaciones inusuales que pueden tener un efecto desproporcionado en las estimaciones de parámetros cuando se utilizan métodos estadísticos estándar. La idea fundamental detrás de la estimación robusta es minimizar el impacto de los valores atípicos o errores en las estimaciones, lo que resulta en estimaciones más confiables y estables en presencia de datos contaminados. Algunos métodos y conceptos clave asociados con la teoría de estimación robusta incluyen:

- **Mínimos Cuadrados Robustos:** En lugar de utilizar el método de los mínimos cuadrados ordinarios (OLS), que es sensible a valores atípicos, los métodos de mínimos cuadrados robustos buscan minimizar una función de pérdida que penaliza menos a los residuos atípicos. Ejemplos de estos métodos incluyen la regresión robusta y la estimación por cuantiles (quantile regression).
- **Funciones de Influencia:** Las funciones de influencia se utilizan para medir cómo una observación individual afecta a una estimación estadística. En la estimación robusta, se busca limitar el efecto de observaciones atípicas al limitar su influencia en la estimación.
- **M-estimadores:** Los M-estimadores son una clase de estimadores robustos que buscan maximizar una función objetivo que es una función de una medida de ajuste y una medida de dispersión. Ejemplos incluyen el estimador de Huber y el estimador de Tukey.
- **Medidas de Robustez:** Se utilizan medidas de robustez para cuantificar cuán robusto es un estimador en particular ante la presencia de valores atípicos. Estas medidas ayudan a seleccionar el estimador adecuado para una situación dada.
- **Bootstrap Robusto:** El método bootstrap robusto es una técnica que se utiliza para estimar la incertidumbre de un estimador robusto al muestrear repetidamente a partir de los datos disponibles, lo que proporciona intervalos de confianza robustos.

La teoría de estimación robusta es especialmente útil en situaciones donde los datos pueden contener valores atípicos o errores que no siguen la distribución normal o los supuestos habituales de los modelos estadísticos. Se aplica en diversas áreas, como econometría, estadísticas aplicadas, análisis de datos y más, para obtener estimaciones confiables incluso en presencia de datos problemáticos. Las estadísticas sólidas buscan proporcionar métodos que emulen los métodos estadísticos populares,

pero que no se vean excesivamente afectados por valores atípicos u otras pequeñas desviaciones de los supuestos del modelo. En estadística, los métodos de estimación clásicos se basan en gran medida en supuestos que a menudo no se cumplen en la práctica. En particular, a menudo se supone que los errores de los datos se distribuyen normalmente, al menos aproximadamente, o que se puede confiar en que el teorema del límite central produce estimaciones distribuidas normalmente. Desafortunadamente, cuando hay valores atípicos en los datos, los estimadores clásicos a menudo tienen un desempeño muy pobre, cuando se los juzga utilizando el punto de ruptura y la función de influencia, que se describen a continuación.

El efecto práctico de los problemas observados en la función de influencia se puede estudiar empíricamente examinando la distribución muestral de los estimadores propuestos bajo un modelo de mezcla, donde se mezcla una pequeña cantidad (1 a 5% suele ser suficiente) de contaminación. Por ejemplo, se puede utilizar una combinación de 95% de distribución normal y 5% de distribución normal con la misma media pero una desviación estándar significativamente mayor (que representa valores atípicos).

## 2.5. Localización de un Robot

La localización de robots es el proceso de determinar dónde se encuentra un robot móvil con respecto a su entorno, que puede o no ser desconocido. Es una de las competencias fundamentales que requiere un robot autónomo, ya que el conocimiento de la propia ubicación del robot es necesaria para la toma de decisiones. Cuando el entorno es conocido, el problema de localización se reduce a estimar la pose (posición y orientación) del robot conforme al sistema coordinado conocido utilizando la información recopilada de diversos sensores que deben ser capaces de observar el entorno y el propio movimiento del robot. Esta información puede ser recuperada a través de diversos medios como lo son el GPS, cámaras, sensores de velocidad, un modelo odométrico, etc.

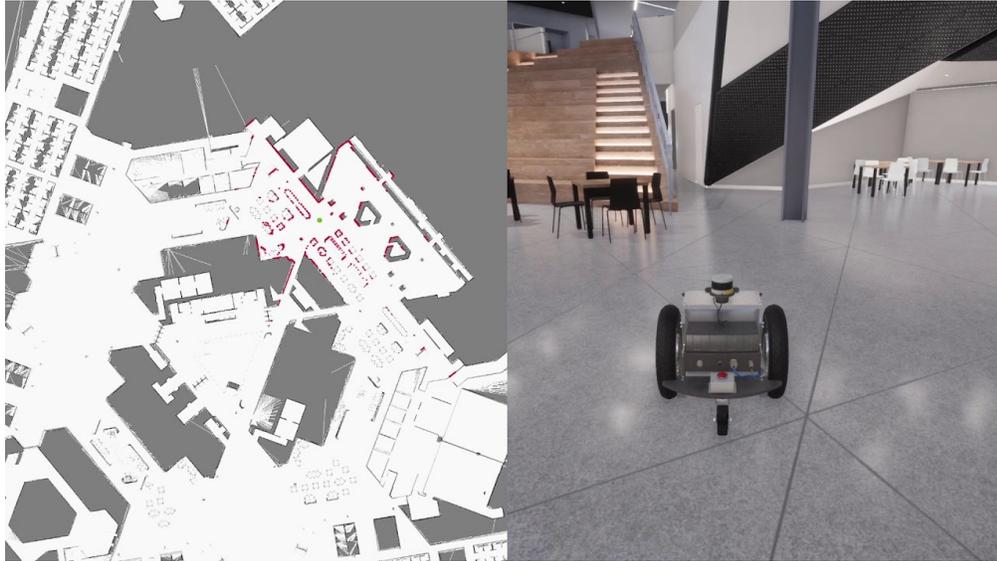
La formulación del problema de localización depende del tipo de mapa disponible así como de las características de los sensores utilizados para observar su entorno. En una posible formulación, el mapa contiene ubicaciones de algunos puntos de referencia o características relevantes del entorno y el robot es capaz de medir el alcance y/o la orientación de estas características en relación con el robot. Una forma común es el mapa de cuadrícula de ocupación, este proporciona las regiones ocupadas y libres de un entorno, en este caso los sensores miden la distancia a la región ocupada más cercana en una dirección determinada.

El problema de seguimiento de posición suele utilizar el mismo enfoque general. La estimación previa de la posición del robot se actualiza continuamente

realizando repetidamente los pasos de predicción y percepción. Las mediciones de los sensores actualizan la creencia de posición durante el paso de predicción. Con la información recopilada del entorno por los sensores el paso de percepción utiliza la creencia proporcionada en el paso de predicción para corregirla. Se debe conocer la ubicación inicial del robot para que funcione el seguimiento de la posición. En general, el robot siempre podría localizarse con precisión los datos percibidos por los sensores no tuvieran error. Sin embargo, dependiendo de la tecnología con la que funcionan pueden presentar ruidos de distintas naturalezas. Por ejemplo, el GPS no es fiable en interiores, y suele estar acompañado de un ruido rosa. Los giroscopios pueden modelarse bien de forma estocástica considerando un ruido blanco. Otras dificultades que no suelen estar modeladas son todas aquellas perturbaciones aleatorias que pueden surgir por la interacción de diversos agentes en el mapa. Además, los mapas suelen representar el espacio de trabajo como una colección de objetos estáticos. Esto es un problema porque es muy probable que el entorno contenga más barreras. Estos pueden ser obstáculos estáticos o dinámicos. El seguimiento de la posición supone que se conoce la postura inicial del robot. La localización del robot se puede lograr adaptando el ruido en el movimiento del robot. El ruido del movimiento (o ruido de proceso) suele ser pequeño. La incertidumbre de pose a menudo se aproxima mediante una distribución unimodal. El problema del seguimiento de la posición es un problema local, ya que la incertidumbre es local y se limita a una región cercana a la verdadera pose del robot. Los entornos estáticos son aquellos donde el único estado variable es la pose del robot. Todos los demás objetos dentro del área de trabajo permanecen en el mismo lugar. Los entornos estáticos tienen algunas propiedades matemáticas interesantes que los hacen susceptibles de una estimación probabilística eficiente. En cambio, los entornos dinámicos poseen objetos distintos del robot cuya ubicación o configuración cambia con el tiempo. De particular interés son los cambios que persisten en el tiempo y que afectan a más de una lectura de sensor. Algunas metodologías han sido ideadas para localización al robot en un entorno desconocido mientras que al mismo tiempo se realiza el mapa, se conocen varios algoritmos para resolverlo en un tiempo, al menos aproximadamente, manejable para ciertos entornos, uno de los más efectivos es el SLAM (Simultaneous Location and mapping). Los algoritmos SLAM se basan en conceptos de geometría computacional, mapeo, odometría y visión por computadora.

## 2.6. Sensores RFID y UWB

La Identificación por Radio Frecuencia (RFID) o tecnología RFID, es una tecnología que permite identificar objetos mediante ondas de radio en el rango de UHF (868 MHz) a través de identificadores únicos, de los que se pueden recuperar o escribir algunos datos y pudiendo leer/escribir cientos de objetos en un intervalo corto de tiempo. El RFID no necesita una línea de visión directa, por lo que solo es imprescindible que la etiqueta RFID este dentro del rango de lectura del lector o



**Figura 2.3.** *Localización simultánea y mapeo*

antena RFID. Sin embargo, su orientación puede afectar al valor del RSSI (Received Signal Strength Indicator) recibido por la antena.

Las soluciones RFID necesitan de 3 elementos para su funcionamiento: una antena RFID, un lector RFID y una etiqueta RFID. Dependiendo del uso y necesidades que se tenga se necesitará un tipo de hardware u otro. El lector RFID es el sistema encargado de transmitir la señal a la antena para que emita ondas de radiofrecuencia. Cuando una de estas ondas llega a una etiqueta RFID, esta se activa y devuelve los datos que contiene en su interior hacia la antena. Estos datos llegan después al lector, que transforma los datos, en información. Las antenas utilizan para leer etiquetas RFID a través de la interacción con su campo eléctrico. Existen antenas de largo alcance, para montaje en el piso, universales y con distinta polarización. Finalmente las etiquetas (tags) consisten en un chip acoplado a una antena, estos poseen un identificador único que dependiendo de la clase (LF, HF, EPC) puede ser de 32-64bits de largo y un espacio de memoria EEPROM (4-32kB) donde se puede almacenar información. Las etiquetas RFID pueden ser activas, semipasivas o pasivas. Las etiquetas pasivas no requieren ninguna fuente de alimentación interna y son dispositivos puramente pasivos. Los otros dos tipos necesitan alimentación alrededor de 3V a 300mAh. Las etiquetas pueden ser muy versátiles pues existen muchas presentaciones, tarjetas, llaveros, estampas, etc.

Algunos de las ventajas de la tecnología RFID se podrían resumir de la siguiente manera:

- La tecnología RFID no necesita línea de visión.
- Las etiquetas pueden leerse y regrabar millones de veces.

- Los datos de las etiquetas RFID pueden encriptarse.
- Los lectores RFID pueden leer cientos de tags a la vez.
- Las etiquetas RFID se pueden imprimir con la información que se quiera.
- Los sistemas RFID pueden integrarse con el ERP u otros sistemas internos para poder gestionar toda la información.

Suponiendo que la impedancia y la polarización coinciden entre el lector y la etiqueta, es posible derivar las fórmulas de la potencia recibida por el chip en la etiqueta y la potencia el lector obtendrá la re-radiación de la etiqueta.

$$P_L = P_{in} G_{reader} G_{tag} \lambda^2 / (4\pi R)^2$$

$$P_{rec} = P_{in} G_{reader}^2 G_{tag}^2 \lambda^4 / (4\pi R)^4$$

Si ocurre un desajuste entre la impedancia y la polarización, las ecuaciones se modifican con los factores  $p$  (factor de desajuste de polarización) y  $\Gamma_{tag}$  (coeficiente de reflexión de la etiqueta).

$$P_L = (1 - |\Gamma_{tag}|^2) p P_{in} G_{reader} G_{tag} \lambda^2 / (4\pi R)^2$$

$$P_{rec} = p P_{in} G_{reader}^2 G_{tag}^2 \lambda^4 / (4\pi R)^4$$

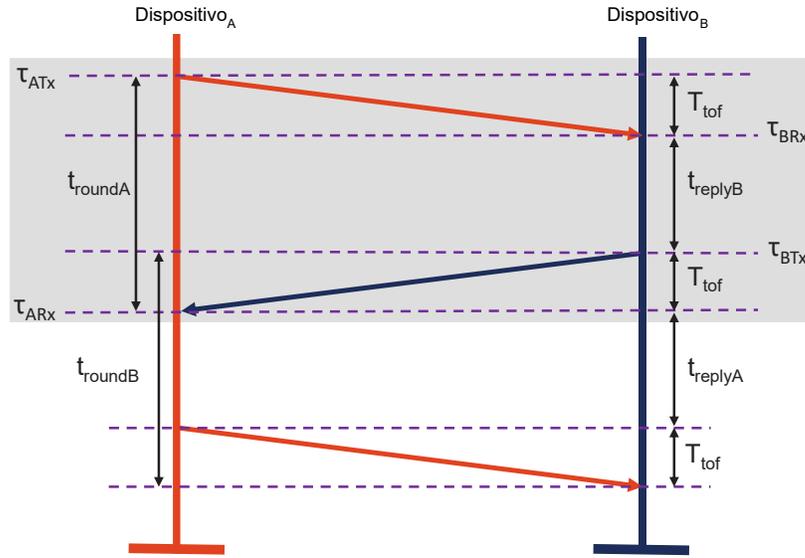
Usando la potencia recibida por la etiqueta y el lector, podemos predecir el rango de lectura de la siguiente manera

$$R_{max} = \text{mín} \{ [p P_{in} G_{reader}^2 G_{tag}^2 \lambda^4 / (4\pi R)^4]^{0.25}, \quad (2.2)$$

$$\sqrt{(1 - |\Gamma_{tag}|^2) p P_{in} G_{reader} G_{tag} \lambda / (4\pi)} \} \quad (2.3)$$

La banda ultraancha (UWB) se ha considerado una opción prometedora. tecnología para sistemas de localización inalámbricos precisos, especialmente en entornos donde no esta disponible un sistema global de navegación por satélite (GNSS). El método alternativo de medición de alcance bidireccional de doble vía (AltDSTWR) es frecuentemente utilizado para la medición de distancias de rango, se considera como el método más confiable entre diferentes esquemas disponibles debido a la mínima influencia del error en el reloj. La diferencia en el tiempo de llegada (TDoA) aprovecha la diferencia en tiempos de propagación de señales de diferentes transmisores, medidos en el receptor. Esto es diferente de la técnica ToF, donde se utiliza el tiempo absoluto de propagación de la señal. El TDoA de al menos tres transmisores es necesario para calcular la ubicación exacta del receptor como intersección de los tres (o más) hiperboloides. El sistema de ecuaciones de hipérbola se pueden resolver mediante métodos lineales regresión o linealizando la ecuación usando series de Taylor. La

precisión de la estimación de TDoA depende del ancho de banda de la señal, de la frecuencia de muestreo del receptor y de la existencia de una línea de visión directa entre los transmisores y el receptor. La sincronización estricta también es requerido, pero a diferencia de las técnicas ToF donde la sincronización es necesario entre el transmisor y el receptor, en el TDoA sólo se requiere sincronización entre los transmisores. Los errores de deriva de reloj en los osciladores rondan las 20 ppm según el estándar IEEE 802.51.4-2011. Por simplicidad, pero sin pérdida de generalidad, los errores de retraso mencionados para la estimación de  $T_{tof}$  en los sistemas de comunicación inalámbrica se puede modelar como una ecuación lineal simple.



**Figura 2.4.** Medición de Rango de dos vías, adaptado de Lian [1]

Considere la figura 2.4, el método AltDS-TWR permite obtener los tiempos de comunicación de ambos dispositivos como se ve en las ecuaciones (2.4) y (2.5) como sigue:

$$t_{roundA} = 2T_{tof} + t_{replyB} \quad (2.4)$$

$$t_{roundB} = 2T_{tof} + t_{replyA} \quad (2.5)$$

Sin embargo, en lugar de combinar las ecuaciones en el método AltDS-TWR se multiplican (2.4) y (2.5) como:

$$t_{roundA}t_{roundB} = (2T_{tof} + t_{replyB})(2T_{tof} + t_{replyA})$$

Simplificando la ecuación, el tiempo de vuelo  $T_{tof}$  se calcula como:

$$T_{tof} = \frac{t_{roundA}t_{roundB} - t_{replyA}t_{replyB}}{t_{roundA} + t_{replyA} + t_{roundB} + t_{replyB}} \quad (2.6)$$

# Capítulo 3

## Predictor $H_2$ -FIR bajo perturbaciones y errores de medición

### 3.1. Resumen

A la estimación usando la norma  $H_2$  también se le llama estimador por mínimos cuadrados [18], [19]. Este estimador se obtiene al minimizar la norma de Frobenius al cuadrado y su implementación esta disponible en forma recursiva [20–25]. La teoría del estimador  $H_2$  fue generalizada en [19,26–28], el problema de minimización por  $H_2$  es convexo y por ello existe una solución cerrada [29,30]. Cabe destacar que para un proceso gaussiano, el Filtro  $H_2$  se convierte en el filtro de Kalman [19]. Una desventaja es que las recursiones están disponibles para matrices de error diagonales, y por ello información importante contenida en las componentes cruzadas es descartada perdiendo así precisión. Otros resultados pueden ser encontrados en [31–34]. Las estructuras de respuesta finita al impulso (FIR) por lotes funcionan en horizontes finitos de los datos más recientes con matrices de error completas. Existen dos tipos de predictores  $H_2$ -FIR, el de receding-horizon(RH) [35] y el óptimo (OFIR) de un paso [36,37]. Algunos otros como el  $H_2$ ,  $H_1$ , y los filtros predictivos mixtos  $H_2/H_1$  RH FIR se pueden encontrar en [38–40]. Aunque la arquitectura filtro  $H_2$ -FIR por lote es capaz de mejorar el rendimiento y demostrar la misma robustez que los estimadores  $H_1$  [40], tienen dos serios inconvenientes: 1) la derivación estándar sugiere minimizar una norma de Frobenius no ponderada al cuadrado que tiene limitaciones prácticas y 2) el ruido de medición y los errores iniciales se ignoran, mientras que en práctica perturban simultáneamente al sistema.

En esta sección se presenta una mejora a la estimación de  $H_2$ -FIR mediante la introducción de la norma ponderada de Frobenius, se deriva un filtro óptimo  $H_2$  con restricción de sesgo ( $H_2$ -OUFIR) bajo perturbaciones y errores de medición, se obtiene la forma de Máxima Verosimilitud (ML) y su equivalencia con el predictor OUFIR bajo ruido gaussiano, además, se desarrollaran soluciones sub-óptimas usando desigualdades de matrices lineales (LMI). Se presentan algunos casos de entornos simulados usando el filtro Kalman y predictores FIR sin sesgo (UFIR) como puntos de referencia.

### 3.2. Modelo y Formulación del problema

La estimación del estado usando un acercamiento  $H_2$  es determinada en el dominio de la frecuencia para minimizar la magnitud de la función de transferencia perturbación/error en un sistema lineal invariante en el tiempo (LTI). Basándonos en la predicción de un paso [41], que coincide con el método backward-Euler, consideraremos el siguiente modelo:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \mathbf{E}\mathbf{u}_k + \mathbf{B}\mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{D}\mathbf{w}_k + \mathbf{v}_k\end{aligned}\quad (3.1)$$

Donde  $\mathbf{x}_k \in \mathbb{R}^K$  es el vector de estados del sistema,  $\mathbf{u}_k \in \mathbb{R}^L$  es el vector de entrada,  $\mathbf{y}_k \in \mathbb{R}^P$  es el vector de salida del sistema, y las matrices son  $\mathbf{F} \in \mathbb{R}^{K \times K}$ ,  $\mathbf{E} \in \mathbb{R}^{K \times L}$ ,  $\mathbf{B} \in \mathbb{R}^{K \times M}$ ,  $\mathbf{H} \in \mathbb{R}^{P \times K}$  y  $\mathbf{D} \in \mathbb{R}^{P \times M}$ . El vector  $\mathbf{w}_k \in \mathbb{R}^M$  representa una perturbación aleatoria limitada y con media cero,  $\mathbf{v}_k \in \mathbb{R}^P$  es un error de medición con media cero que no es obligatoriamente gaussiano.

Para diseñar un estimador FIR y llegar a un rendimiento óptimo o sub-óptimo, el sistema 3.1 se extiende sobre un horizonte  $[m, k]$  de  $N$  puntos, desde  $m = k - N + 1$  a  $k$ , que se construye como indican las siguientes ecuaciones.

$$\mathbf{x}_{k+1} = \mathbf{F}^N \mathbf{x}_m + \bar{\mathbf{S}}_N \mathbf{U}_{m,k} + \bar{\mathbf{D}}_N \mathbf{W}_{m,k}, \quad (3.2)$$

$$\mathbf{Y}_{m,k} = \mathbf{H}_N \mathbf{x}_m + \mathbf{L}_N \mathbf{U}_{m,k} + \mathbf{T}_N \mathbf{W}_{m,k} + \mathbf{V}_{m,k}, \quad (3.3)$$

Los vectores aumentados se definen como:

$$\begin{aligned}\mathbf{U}_{m,k} &= [\mathbf{u}_m^T \ \mathbf{u}_{m+1}^T \ \cdots \ \mathbf{u}_k^T]^T, \\ \mathbf{W}_{m,k} &= [\mathbf{w}_m^T \ \mathbf{w}_{m+1}^T \ \cdots \ \mathbf{w}_k^T]^T, \\ \mathbf{V}_{m,k} &= [\mathbf{v}_m^T \ \mathbf{v}_{m+1}^T \ \cdots \ \mathbf{v}_k^T]^T, \\ \mathbf{Y}_{m,k} &= [\mathbf{y}_m^T \ \mathbf{y}_{m+1}^T \ \cdots \ \mathbf{y}_k^T]^T\end{aligned}$$

y las matrices aumentadas se definen como:

$$\begin{aligned}
 \mathbf{H}_N &= \bar{\mathbf{H}}_N \mathbf{F}_N, \\
 \mathbf{L}_N &= \bar{\mathbf{H}}_N \mathbf{S}_N, \\
 \mathbf{T}_N &= \mathbf{G}_N + \bar{\mathbf{T}}_N, \\
 \mathbf{G}_N &= \bar{\mathbf{H}}_N \mathbf{D}_N, \\
 \bar{\mathbf{H}}_N &= \text{diag}(\mathbf{H} \ \mathbf{H} \ \dots \ \mathbf{H}), \\
 \bar{\mathbf{T}}_N &= \text{diag}(\mathbf{D} \ \mathbf{D} \ \dots \ \mathbf{D}), \\
 \mathbf{F}_N &= [ \ \mathbf{I} \ \mathbf{F}^T \ \dots \ (\mathbf{F}^{N-2})^T \ (\mathbf{F}^{N-1})^T \ ]^T, \\
 \mathbf{S}_N &= \begin{bmatrix} \mathbf{E} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{F}\mathbf{E} & \mathbf{E} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{F}^{N-2}\mathbf{E} & \mathbf{F}^{N-3}\mathbf{E} & \dots & \mathbf{E} & \mathbf{0} \\ \mathbf{F}^{N-1}\mathbf{E} & \mathbf{F}^{N-2}\mathbf{E} & \dots & \mathbf{F}\mathbf{E} & \mathbf{E} \end{bmatrix},
 \end{aligned}$$

La matriz  $\mathbf{D}_N$  surge de sustituir la matriz  $\mathbf{E}$  por  $\mathbf{B}$  en la matriz  $\mathbf{S}_N$ . La matriz  $\bar{\mathbf{S}}_N$  es el último vector renglón de la matriz  $\mathbf{S}_N$ , el mismo caso sucede con las matrices  $\bar{\mathbf{D}}_N$  y  $\bar{\mathbf{D}}_N$ . Para derivar el predictor  $H_2$ -OFIR en la forma por lotes [42] se propone el estimado predecido con la ecuación 3.4.

$$\begin{aligned}
 \hat{\mathbf{x}}_{k+1} &= \mathcal{H}_N \mathbf{Y}_{m,k} + \bar{\mathcal{H}}_N \mathbf{U}_{m,k}, \\
 &= \mathcal{H}_N \mathbf{H}_N x_m + (\mathcal{H}_N \mathbf{L}_N + \bar{\mathcal{H}}_N) \mathbf{U}_{m,k} + \mathcal{H}_N \mathbf{T}_N \mathbf{W}_{m,k} + \mathcal{H}_N \mathbf{V}_{m,k} \quad (3.4)
 \end{aligned}$$

Las matrices  $\mathcal{H}_N$  y  $\bar{\mathcal{H}}_N$  corresponden, respectivamente, a las ganancias homogénea y forzada de un predictor FIR. Típicamente se requiere que la predicción no tenga sesgo (unbiased), por ello se usará la restricción  $E\{\mathbf{x}_{k+1}\} = E\{\hat{\mathbf{x}}_{k+1}\}$  con la ecuación de estado (3.2) y con la predicción estimada (3.4) obteniendo lo siguiente

$$\begin{aligned}
 E\{\hat{\mathbf{x}}_{k+1}\} &= \mathcal{H}_N \mathbf{H}_N x_m + (\mathcal{H}_N \mathbf{L}_N + \bar{\mathcal{H}}_N) \mathbf{U}_{m,k} \\
 E\{\mathbf{x}_{k+1}\} &= \mathbf{F}^N x_m + \bar{\mathbf{S}}_N \mathbf{U}_{m,k}
 \end{aligned}$$

Debido a las propiedades de linealidad del sistema, la respuesta puede ser determinada por superposición considerando la solución homogénea ( $\mathbf{U}_{m,k} = \mathbf{0}$ ) y la forzada ( $x_m = \mathbf{0}$ ), obteniendo las ganancias para la restricción sin sesgo.

$$\mathbf{I} = \mathcal{H}_N \mathbf{C}_N \quad (3.5)$$

$$\bar{\mathcal{H}}_N = \bar{\mathbf{S}}_N - \mathcal{H}_N \mathbf{L}_N \quad (3.6)$$

Definimos la matriz  $\mathbf{C}_N = \mathbf{H}_N \mathbf{F}^{-N}$ , que será usada para derivar el diseño predictor óptimo  $H_2$ -OUFIR. El error de predicción se define como  $\varepsilon_{k+1} = x_{k+1} - \hat{x}_{k+1}$ ,

sustituyendo las ecuaciones (3.2) y (3.4) y considerando las restricciones de sesgo se obtiene

$$\varepsilon_{k+1} = (\mathbf{F}^N - \mathcal{H}_N \mathbf{H}_N) x_m + (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N) \mathbf{W}_{m,k} - \mathcal{H}_N \mathbf{V}_{m,k}$$

Definiremos la matriz residual de sesgo  $\mathcal{B}_N$ , la matriz de error residual del sistema  $\mathcal{W}_N$ , y la matriz de error residual de observación  $\mathcal{V}_N$  como

$$\begin{aligned} \mathcal{B}_N &= \mathbf{F}^N - \mathcal{H}_N \mathbf{H}_N, \\ \mathcal{W}_N &= \bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N, \\ \mathcal{V}_N &= \mathcal{H}_N \end{aligned}$$

Para obtener el error de predicción

$$\varepsilon_{k+1} = \mathcal{B}_N x_m + \mathcal{W}_N \mathbf{W}_{m,k} - \mathcal{V}_N \mathbf{V}_{m,k} \quad (3.7)$$

Finalmente el predictor de estado estará dado por

$$\hat{x}_{k+1} = \mathcal{H}_N \mathbf{Y}_{m,k} + (\hat{\mathbf{S}}_N - \mathcal{H}_N \mathbf{L}_N) \mathbf{U}_{m,k} \quad (3.8)$$

Y la matriz de error de predicción  $\mathbf{P}_{k+1} = E\{\varepsilon_{k+1} \varepsilon_{k+1}^T\}$

$$\mathbf{P}_{k+1} = \mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T \quad (3.9)$$

Donde las matrices de error estan dadas por

$$\begin{aligned} \chi_m &= E\{x_m x_m^T\}, \\ \mathcal{Q}_N &= E\{\mathbf{W}_{m,k} \mathbf{W}_{m,k}^T\}, \\ \mathcal{R}_N &= E\{\mathbf{V}_{m,k} \mathbf{V}_{m,k}^T\}, \end{aligned}$$

Estas matrices no son conocidas con precisión pero se asume que están limitadas. El problema se formula como sigue: Se desea mejorar el filtro  $H_2$  introduciendo una norma ponderada de Frobenius y a partir de ello, derivar un predictor óptimo con sesgo restringido ( $H_2$ -OUFIR). Este operará en una ventana  $[m, k]$  para producir una estimación en  $k + 1$  bajo perturbaciones y errores de medición. A diferencia de la solución estándar [35] se derivará el predictor  $H_2$ -OUFIR por la minimización de la media cuadrática de la norma de Frobenius definida por (3.10).

$$\begin{aligned} \|\mathcal{J}(z)\|_F^2 &= E_z \{E_k \{ \text{tr}[\mathcal{J}(z) \varpi_k \varpi_k^* \mathcal{J}^*(z)] \} \} \\ &= \frac{1}{2\pi} \int_0^{2\pi} \text{tr} [\mathcal{J}(e^{j\omega\tau}) E\{\varpi_k \varpi_k^*\} \mathcal{J}^*(e^{j\omega\tau})] d\omega\tau, \end{aligned} \quad (3.10)$$

Donde  $\tau$  es el tiempo de muestreo,  $\mathcal{J}^*(z)$  es la traspuesta conjugada de  $\mathcal{J}(z)$  y  $\varpi_k$  es el vector de peso. La ganancia del predictor  $\mathcal{H}_N$  está definida para sistemas estables ( $\mathbf{D} = 0$ ) por el siguiente teorema.

**Teorema 3.2.1.** *Dado el sistema estable representado por el modelo en (3.1) en el que  $\mathbf{D} = 0$ , la ganancia para el predictor  $H_2$ -OUFIR está dado por*

$$\begin{aligned} \mathcal{H}_N &= (\mathbf{C}_{NN}^{T-1} \mathbf{C}_N)^{-1} \mathbf{C}_{NN}^{T-1} \\ &\quad + \bar{\mathbf{D}}_N \mathcal{Q} \mathbf{G}_N^{T-1} [\mathbf{I} - \mathbf{C}_N (\mathbf{C}_{NN}^{T-1} \mathbf{C}_N)^{-1} \mathbf{C}_{NN}^{T-1}] \end{aligned} \quad (3.11)$$

Donde  $\mathbf{C}_N = \mathbf{G}_N \mathcal{Q}_N \mathbf{G}_N^T + \mathcal{R}_N$ , y la matriz de error

$$\mathbf{P}_k = \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T \quad (3.12)$$

las ecuaciones anteriores consideran las representaciones como matriz aumentada definidas al inicio de esta sección.

*Demostración.* Representando la ecuación (3.7) como

$$\varepsilon_{k+1} = \varepsilon_{x(k+1)} + \varepsilon_{w(k+1)} + \varepsilon_{v(k+1)} \quad (3.13)$$

Donde los errores están dados como  $\varepsilon_{x(k+1)} = \mathcal{B}_N x_m$ ,  $\varepsilon_{w(k+1)} = \mathcal{W}_N \mathbf{W}_{m,k}$  y  $\varepsilon_{v(k+1)} = -\mathcal{V}_N \mathbf{V}_{m,k}$ . Para errores sin correlación la ecuación 3.10 resulta en

$$\|\mathcal{J}(z)\|_F^2 = \|\mathcal{J}_x(z)\|_F^2 + \|\mathcal{J}_w(z)\|_F^2 + \|\mathcal{J}_v(z)\|_F^2 \quad (3.14)$$

Las funciones de error en 3.7 ni sus respectivas transformada  $z$  son, por lo general, conocidas. Asignaremos las funciones de transferencia como  $\mathcal{J}_x(z) = \mathcal{J}_w(z) = \mathcal{J}_v(z) = \mathbf{I}$  y como factor de peso  $\varpi_k$  se usarán los encontrados en la ecuación de predicción de error, lo que convierte a la ecuación 3.10 en

$$\|\mathcal{J}(z)\|_F^2 = \text{tr}(\mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T) \quad (3.15)$$

El método propuesto en [35] requiere mayor esfuerzo y considera la ecuación (3.15) como una desigualdad. Para garantizar el rendimiento  $H_2$  y hacer que el predictor este limitado en sesgo, se sustituye la ecuación (3.5) en (3.15) removiendo el término  $\chi_m$ . Considere el problema de optimización en el que se busca minimizar la norma de Frobenius sujeto a la restricción de sesgo.

$$\begin{aligned} \text{mín} \quad & \|\mathcal{J}(z)\|_F^2 \\ \text{s.t.} \quad & \mathbf{I} = \mathcal{H}_N \mathbf{C}_N \end{aligned}$$

Este da lugar a la siguiente función potencial.

$$J(\mathcal{H}_N, \Lambda) = \text{tr}(\mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T) + \text{tr} \Lambda (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) \quad (3.16)$$

Donde  $\Lambda$  son multiplicadores de Lagrange. Debido a que el problema  $H_2$  es cóncavo podemos encontrar la ganancia del estimador a través del siguiente problema de minimización.

$$\mathcal{H}_N = \arg \text{mín}_{\mathcal{H}_N, \Lambda} J(\mathcal{H}_N, \Lambda) \quad (3.17)$$

Conforme al método de los multiplicadores de Lagrange resolveremos

$$\begin{aligned} \frac{\partial}{\partial \mathcal{H}_N} \left( \text{tr} \left( (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N) \mathcal{Q}_N (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)^T + \right. \right. \\ \left. \left. \mathcal{H}_N \mathcal{R}_N \mathcal{H}_N^T \right) + \text{tr} \Lambda (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) \right) = 0 \\ \frac{\partial}{\partial \Lambda} \left( \text{tr} \Lambda (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) \right) = 0 \end{aligned}$$

Expandiendo las expresiones anteriores

$$\begin{aligned} \frac{\partial}{\partial \mathcal{H}_N} \left( \text{tr} (\bar{\mathbf{D}}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T - \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{T}_N^T \mathcal{H}_N^T - \mathcal{H}_N \mathbf{T}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T \right. \\ \left. + \mathcal{H}_N \mathbf{T}_N \mathcal{Q}_N \mathbf{T}_N^T \mathcal{H}_N^T + \mathcal{H}_N \mathcal{R}_N \mathcal{H}_N^T) + \text{tr} \Lambda (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) \right) = 0 \\ \frac{\partial}{\partial \Lambda} \left( \text{tr} (\Lambda - \Lambda \mathcal{H}_N \mathbf{C}_N) \right) = 0 \end{aligned}$$

Debido a las condiciones estabilidad del sistema ( $\bar{\mathbf{T}}_N = \mathbf{0}$ ,  $\mathbf{T}_N = \mathbf{G}_N$ )

$$\begin{aligned} \frac{\partial}{\partial \mathcal{H}_N} \left( \text{tr} (\bar{\mathbf{D}}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T - \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T \mathcal{H}_N^T - \mathcal{H}_N \mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T \right. \\ \left. + \mathcal{H}_N \mathbf{G}_N \mathcal{Q}_N \mathbf{G}_N^T \mathcal{H}_N^T + \mathcal{H}_N \mathcal{R}_N \mathcal{H}_N^T) + \text{tr} \Lambda (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) \right) = 0 \\ \frac{\partial}{\partial \Lambda} \left( \text{tr} (\Lambda) - \text{tr} (\Lambda \mathcal{H}_N \mathbf{C}_N) \right) = 0 \end{aligned}$$

Aplicando las identidades de primer orden para las trazas

$$\frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{X}\mathbf{A}) = \mathbf{A}^T \quad (3.18)$$

$$\frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{A}\mathbf{X}^T) = \mathbf{A} \quad (3.19)$$

$$\frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{A}\mathbf{X}\mathbf{B}) = \mathbf{A}^T \mathbf{B}^T \quad (3.20)$$

$$\frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{X}\mathbf{A}\mathbf{X}^T) = \mathbf{X}\mathbf{A}^T + \mathbf{X}\mathbf{A} \quad (3.21)$$

$$\begin{aligned} -\bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T - (\mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T)^T + \mathcal{H}_N ((\mathbf{G}_N \mathcal{Q}_N \mathbf{G}_N^T) + (\mathbf{G}_N \mathcal{Q}_N \mathbf{G}_N^T)^T) + \\ \mathcal{H}_N (\mathcal{R}_N + \mathcal{R}_N^T) - \Lambda^T \mathbf{C}_N^T = 0 \\ \mathbf{I} - (\mathcal{H}_N \mathbf{C}_N)^T = 0 \end{aligned}$$

Debido a las condiciones de simetría de las matrices  $\mathcal{Q}_N$  y  $\mathcal{R}_N$  llegamos a

$$-2\bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T + 2\mathcal{H}_N \mathcal{R}_N - \Lambda^T \mathbf{C}_N^T = 0 \quad (3.22)$$

$$\mathbf{I} - \mathbf{C}_N^T \mathcal{H}_N^T = 0 \quad (3.23)$$

$$\Omega_N = \mathbf{G}_N \mathcal{Q}_N \mathbf{G}_N^T + \mathcal{R}_N \quad (3.24)$$

Cabe destacar la equivalencia entre (3.23) y (3.5). De la ecuación (3.22)

$$\begin{aligned}\Lambda^T \mathbf{C}_N^T &= -2\bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T + 2\mathcal{H}_N \Omega_N, \\ \mathbf{C}_N \Lambda &= 2(\Omega_N \mathcal{H}_N^T - \mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T),\end{aligned}$$

premultiplicando por  $\mathbf{C}_N^T \Omega_N^{-1}$

$$\begin{aligned}\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N \Lambda &= 2\mathbf{C}_N^T \Omega_N^{-1} (\Omega_N \mathcal{H}_N^T - \mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T), \\ &= 2(\mathbf{C}_N^T \Omega_N^{-1} \Omega_N \mathcal{H}_N^T - \mathbf{C}_N^T \Omega_N^{-1} \mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T), \\ &= 2(\mathbf{I} - \mathbf{C}_N^T \Omega_N^{-1} \mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T)\end{aligned}\tag{3.25}$$

Premultiplicando (3.25) por  $(\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1}$

$$\Lambda = 2(\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} (\mathbf{I} - \mathbf{C}_N^T \Omega_N^{-1} \mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T)\tag{3.26}$$

La ecuación (3.26) corresponde a los coeficientes que combinados linealmente con el gradiente de la restricción de sesgo igualan al gradiente de la norma de Frobenius en su mínimo local. Finalmente, sustituyendo este resultado en (3.22) se obtiene la ganancia del predictor.

$$\begin{aligned}-\bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T + \mathcal{H}_N \Omega_N - ((\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} (\mathbf{I} - \mathbf{C}_N^T \Omega_N^{-1} \mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T))^T \mathbf{C}_N^T &= 0 \\ \mathcal{H}_N &= \left( \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T + ((\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} (\mathbf{I} - \mathbf{C}_N^T \Omega_N^{-1} \mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T))^T \mathbf{C}_N^T \right) \Omega_N^{-1} \\ &= \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T \Omega_N^{-1} + (\mathbf{I} - \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T \Omega_N^{-1} \mathbf{C}_N) (\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Omega_N^{-1} \\ &= \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T \Omega_N^{-1} \left( \mathbf{I} - \mathbf{C}_N (\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Omega_N^{-1} \right) + (\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Omega_N^{-1}\end{aligned}$$

Tomando la ecuación (3.9) y aplicando la restricción de sesgo (3.5) se obtiene

$$\begin{aligned}\mathbf{P}_{k+1} &= \mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T \\ &= (\mathbf{F}^N - \mathcal{H}_N \mathbf{H}_N) \chi_m (\mathbf{F}^N - \mathcal{H}_N \mathbf{H}_N)^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T \\ &= (\mathbf{F}^N - \mathcal{H}_N \mathbf{C}_N \mathbf{F}^N) \chi_m (\mathbf{F}^N - \mathcal{H}_N \mathbf{C}_N \mathbf{F}^N)^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T \\ &= (\mathbf{F}^N - \mathbf{F}^N) \chi_m (\mathbf{F}^N - \mathbf{F}^N)^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T \\ &= \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T\end{aligned}$$

Comprobando así la ecuación (3.12). □

### 3.3. Predictor $H_2$ -OUFIR en la forma de máxima verosimilitud

El siguiente teorema afirma que la ganancia (3.11) del predictor  $H_2$ -OUFIR es equivalente a la ganancia de un predictor FIR de máxima verosimilitud (ML).

### 3.3. PREDICTOR $H_2$ -OUFIR EN LA FORMA DE MÁXIMA VEROSIMILITUD

---

**Teorema 3.3.1.** *Dado un sistema LTI estable representado con el modelo (3.1) en el que  $\mathbf{D} = 0$ , la ganancia del predictor  $H_2$ -OUFIR es equivalente a la ganancia del predictor por máxima verosimilitud*

$$\mathcal{H}_N = (\mathbf{C}_N^T \Sigma_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Sigma_N^{-1} \quad (3.27)$$

Donde, la matriz de error  $\Sigma_N$  esta dada por

$$\Sigma_N = (\mathbf{G}_N - \mathbf{C}_N \bar{\mathbf{D}}_N) \mathcal{Q}_N (\mathbf{G}_N - \mathbf{C}_N \bar{\mathbf{D}}_N)^T + \mathcal{R}_N \quad (3.28)$$

*Demostración.* La predicción de un estimado por ML puede ser encontrado como

$$\hat{\mathbf{x}}_{k+1} = \arg \max_{x_{k+1}} p(\mathbf{Y}_{m,k} | \mathbf{x}_k) \quad (3.29)$$

Tomando la predicción de la ecuación (3.2) podemos definir el estado inicial como

$$\mathbf{x}_m = \mathbf{F}^{-N} \mathbf{x}_{k+1} - \mathbf{F}^{-N} \bar{\mathbf{S}}_N \mathbf{U}_{m,k} - \mathbf{F}^{-N} \bar{\mathbf{D}}_N \mathbf{W}_{m,k},$$

sustituyendo esta expresión en (3.3) tenemos

$$\begin{aligned} \mathbf{Y}_{m,k} &= \mathbf{H}_N \left( \mathbf{F}^{-N} \mathbf{x}_{k+1} - \mathbf{F}^{-N} \bar{\mathbf{S}}_N \mathbf{U}_{m,k} - \mathbf{F}^{-N} \bar{\mathbf{D}}_N \mathbf{W}_{m,k} \right) \\ &\quad + \mathbf{L}_N \mathbf{U}_{m,k} + \mathbf{T}_N \mathbf{W}_{m,k} + \mathbf{V}_{m,k}, \\ \mathbf{Y}_{m,k} - \mathbf{C}_N \mathbf{x}_{k+1} &= \mathcal{N}_{m,k} \end{aligned}$$

Donde  $\mathcal{N}_{m,k} = (\mathbf{T}_N - \mathbf{C}_N \bar{\mathbf{D}}_N) \mathbf{W}_{m,k} + (\mathbf{L}_N - \mathbf{C}_N \bar{\mathbf{S}}_N) \mathbf{U}_{m,k} + \mathbf{V}_{m,k}$  representa una componente de ruido aleatorio. La verosimilitud de  $x_{k+1}$  puede ser representada como una distribución gaussiana multidimensional de la forma

$$p(\mathbf{Y}_{m,k} | \mathbf{x}_k) \propto \exp \left( -\frac{1}{2} (\mathbf{Y}_{m,k} - \mathbf{C}_N \mathbf{x}_{k+1})^T \Sigma_N^{-1} (\mathbf{Y}_{m,k} - \mathbf{C}_N \mathbf{x}_{k+1}) \right) \quad (3.30)$$

Debido a las condiciones de estabilidad  $\mathbf{T}_N = \mathbf{G}_N$ , la matriz de error corresponde a  $\Sigma_N = E\{\mathcal{N}_{m,k} \mathcal{N}_{m,k}^T\}$ . Resolviendo el operador obtenemos

$$\begin{aligned} \Sigma_N &= (\mathbf{G}_N - \mathbf{C}_N \bar{\mathbf{D}}_N) E\{\mathbf{W}_{m,k} \mathbf{W}_{m,k}^T\} (\mathbf{G}_N - \mathbf{C}_N \bar{\mathbf{D}}_N)^T + E\{\mathbf{V}_{m,k} \mathbf{V}_{m,k}^T\} \\ &= (\mathbf{G}_N - \mathbf{C}_N \bar{\mathbf{D}}_N) \mathcal{Q}_N (\mathbf{G}_N - \mathbf{C}_N \bar{\mathbf{D}}_N)^T + \mathcal{R}_N \end{aligned}$$

Comprobando así la ecuación (3.28). El problema de optimización se puede resolver minimizando la forma cuadrática en (3.30) como:

$$\hat{\mathbf{x}}_{k+1} = \arg \max_{x_{k+1}} \left( -\frac{1}{2} (\mathbf{Y}_{m,k} - \mathbf{C}_N \mathbf{x}_{k+1})^T \Sigma_N^{-1} (\mathbf{Y}_{m,k} - \mathbf{C}_N \mathbf{x}_{k+1}) \right) \quad (3.31)$$

Derivando el argumento en (3.31) e igualando a cero podemos encontrar la predicción estimada.

$$\begin{aligned} \frac{d}{dx_{k+1}} \left( -\frac{1}{2} (\mathbf{Y}_{m,k} - \mathbf{C}_N \mathbf{x}_{k+1})^T \Sigma_N^{-1} (\mathbf{Y}_{m,k} - \mathbf{C}_N \mathbf{x}_{k+1}) \right) &= 0 \\ \frac{d}{dx_{k+1}} \left( -\mathbf{Y}_{m,k}^T \Sigma_N^{-1} \mathbf{C}_N \mathbf{x}_{k+1} - x_{k+1}^T \mathbf{C}_N^T \Sigma_N^{-1} \mathbf{Y}_{m,k} + x_{k+1}^T \mathbf{C}_N^T \Sigma_N^{-1} \mathbf{C}_N \mathbf{x}_{k+1} \right) &= 0 \\ -\mathbf{C}_N^T \Sigma_N^{-1} \mathbf{Y}_{m,k} - (\mathbf{Y}_{m,k}^T \Sigma_N^{-1} \mathbf{C}_N)^T + ((\mathbf{C}_N^T \Sigma_N^{-1} \mathbf{C}_N)^T + (\mathbf{C}_N^T \Sigma_N^{-1} \mathbf{C}_N)) \mathbf{x}_{k+1} &= 0 \\ \mathbf{x}_{k+1} &= (\mathbf{C}_N^T \Sigma_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Sigma_N^{-1} \mathbf{Y}_{m,k} \end{aligned}$$

Probando así la ganancia descrita en (3.27).  $\square$

### 3.4. Equivalencia con el predictor OUFIR

Por el teorema de Parseval, la minimización del error en el espectro de potencia puede ser calculada de forma equivalente en el dominio del tiempo, minimizando el error cuadrático medio (MSE). La equivalencia entre los predictores  $H_2$ -OUFIR y OUFIR está dada por el siguiente teorema.

**Teorema 3.4.1.** *Dado un sistema LTI estable representado con el modelo (3.1) en el que  $\mathbf{D} = 0$ , para el proceso gaussiano, la ganancia del predictor  $H_2$ -OUFIR es equivalente a la del predictor OUFIR.*

*Demostración.* El rendimiento del predictor OUFIR está garantizado por la condición de ortogonalidad  $E\{\varepsilon_{k+1} \mathbf{Y}_{m,k}^T\} = 0$

$$\begin{aligned} 0 = E \left\{ \left( (\mathbf{F}^N - \mathcal{H}_N \mathbf{H}_N) x_m + (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N) \mathbf{W}_{m,k} - \mathcal{H}_N \mathbf{V}_{m,k} \right) \right. \\ \left. (\mathbf{H}_N \mathbf{x}_m + \mathbf{L}_N \mathbf{U}_{m,k} + \mathbf{T}_N \mathbf{W}_{m,k} + \mathbf{V}_{m,k})^T \right\} \end{aligned}$$

Aplicando la restricción de sesgo (3.5) se puede remover el término que incluye a  $\chi_m$ , al aplicar el operador y asumiendo que las matrices  $\mathbf{W}_{m,k}$  y  $\mathbf{V}_{m,k}$  no tienen correlación obtenemos

$$0 = ((\mathbf{F}^N - \mathcal{H}_N \mathbf{H}_N) \chi_m \mathbf{H}_N^T + (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{G}_N) \mathcal{Q}_N \mathbf{G}_N^T - \mathcal{H}_N \mathcal{R}_N) \quad (3.32)$$

Introduciendo multiplicadores de Lagrange podemos llegar a la función de costo (3.16) de la que derivan (3.11) y (3.12) finalizando así la demostración. El predictor  $H_2$ -OUFIR es más general, por lo tanto, este generaliza el caso del predictor OFIR que es un caso especial para un proceso gaussiano.  $\square$

### 3.5. Predictor subóptimo $H_2$ -FIR usando LMI

Para diseñar un algoritmo numérico robusto como el predictor  $H_1/H_2$  [37], se requiere usar un algoritmo  $H_2$ -FIR subóptimo empleando LMI [43, 44]. Introduzcamos la matriz auxiliar de bloques definida por:

$$\mathcal{Z} > \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T \quad (3.33)$$

Usando las definiciones para el espacio de estados extendido, expandimos la ecuación (3.33) a

$$\begin{aligned} \mathcal{Z} &> (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N) \mathcal{Q}_N (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)^T + \mathcal{H}_N \mathcal{R}_N \mathcal{H}_N^T \\ 0 &< \mathcal{Z} - \bar{\mathbf{D}}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T + \mathcal{K}^T \mathcal{H}_N^T + \mathcal{H}_N \mathcal{K} - \mathcal{H}_N \Omega_N \mathcal{H}_N^T \end{aligned} \quad (3.34)$$

Donde  $\mathcal{K}^T = \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T$ .

**Observación.** Las desigualdades pueden ser llevada a la forma de un LMI usando el complemento de Schur. La idea básica de esta relación es que una LMI es equivalente al siguiente sistema de desigualdades de matrices.

$$\begin{aligned} \mathbf{F}(x) &:= \begin{bmatrix} \mathbf{F}_{11}(X) & \mathbf{F}_{12}(X) \\ \mathbf{F}_{21}(X) & \mathbf{F}_{22}(X) \end{bmatrix} > 0 \\ &\mathbf{F}_{11}(X) > 0 \\ &\mathbf{F}_{22}(X) > 0 \\ \mathbf{F}_{11}(X) - \mathbf{F}_{12}(X) \mathbf{F}_{22}^{-1}(X) \mathbf{F}_{12}^T(X) &> 0 \\ \mathbf{F}_{22}(X) - \mathbf{F}_{21}^T(X) \mathbf{F}_{11}^{-1}(X) \mathbf{F}_{21}(X) &> 0 \end{aligned}$$

Usando el complemento de Schur [45] la ecuación (3.34) puede ser representada como la siguiente desigualdad

$$\begin{bmatrix} \mathcal{Z} - \bar{\mathbf{D}}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T + \mathcal{K}^T \mathcal{H}_N^T + \mathcal{H}_N \mathcal{K} & \mathcal{H}_N \\ \mathcal{H}_N^T & \Omega_N^{-1} \end{bmatrix} > 0 \quad (3.35)$$

La ganancia del predictor  $H_2$ -FIR restringido en sesgo puede ser obtenida numéricamente resolviendo el siguiente problema de minimización.

$$\begin{aligned} \mathcal{H}_N &= \min_{\mathcal{H}_N, \mathcal{Z}} \text{tr} \mathcal{Z} \\ \text{s.t.} & \text{ (3.35) y } \mathbf{I} = \mathcal{H}_N \mathbf{C}_N \end{aligned} \quad (3.36)$$

Es recomendable iniciar la minimización con la ganancia del filtro UFIR estándar  $\hat{\mathcal{H}}_N = (\mathbf{C}_N^T \mathbf{C}_N)^{-1} \mathbf{C}_N^T$  [46]. Una vez determinada la ganancia para el filtro subóptimo  $\mathcal{H}_N$  el estimado de la predicción puede ser calculado con (3.8) y la matriz de error con (3.9).

# Capítulo 4

## Filtrado Robusto $H_2$ -OFIR

### 4.1. Resumen

El seguimiento de objetos en movimiento utilizando el posicionamiento global (GPS) y otras señales satelitales se requiere para una variedad de propósitos, desde la seguridad hasta la confianza [47, 48]. La cantidad de información relacionada con objetivos en movimiento es a menudo representada por modelos estadísticos relativamente simples [49, 50] con unos pocos estados, ya que formular dinámicas adecuadas es a menudo una tarea difícil [51]. Los algoritmos de seguimiento [52] más comúnmente empleados utilizan el filtro de Kalman (KF) como herramienta de estimación del estado. Para mejorar el rendimiento en condiciones adversas como ambientes bajo perturbaciones, se desarrollaron varios filtros robustos limitados por normas [53], como el  $H_2$  y  $H_\infty$  [54, 55], filtro de energía a pico [56], filtro pico a pico [57], y filtro por teoría de juegos  $H_\infty$  [58]. Desde el principio, los filtros  $H_\infty$  han encontrado aplicaciones en el seguimiento [59] y han evolucionado mucho en las siguientes décadas. Algunas formas cerradas para filtros de seguimiento de objetivos de tiempo continuo de dos estados tipo Kalman,  $H_2$  y  $H_\infty$  se pueden encontrar en [60] y [61]. El problema de optimización minimax relacionado con el filtrado  $H_\infty$  fue resuelto para el seguimiento robusto de objetivos en [62]. El filtro adaptativo basado en modelo  $H_2$  fue desarrollado para posicionamiento 3D y seguimiento de sistemas de tiempo continuo en [63]. Vale la pena señalar que el rendimiento del seguimiento se puede mejorar significativamente utilizando filtros de respuesta de impulso finito (FIR), que ignoran los errores más allá del horizonte de promediado y, por lo tanto, son más robustos. A diferencia de KF, la estructura entrada acotada/salida acotada (BIBO) están basadas en convolución y, por lo tanto, los métodos robustos se pueden aplicar directamente. El filtro FIR de horizonte de retroceso temprano (RH) fue diseñado en [35] para ser varianza mínima sin sesgo (no óptima). Al poco tiempo después, varios filtros

RH  $H_2$ ,  $H_\infty$  y  $H_2/H_\infty$  con restricción de sesgo fueron propuestos en [37]. En los años siguientes, se lograron importantes progresos mediante el desarrollo y la investigación de filtros  $H_2$ -FIR en [24, 38–40], filtros  $H_2$ -FIR con restricciones de envolvente óptima en [22] y [64], el algoritmo  $H_\infty$ -FIR eficiente en frecuencia en [65], el filtro FIR de mínimos cuadrados usando el modelo de movimiento de velocidad constante en [66], y algoritmos FIR por energía a pico en [67–69]. Además, se desarrollaron el filtro FIR óptimo (OFIR) [36], [70], que generaliza el KF en horizontes infinitos, y el FIR robusto y sin sesgo (UFIR) filtro [42], que ignora por completo la media cero ruido y valores iniciales. El problema de filtrado  $H_2$  convexo tiene soluciones de forma cerrada, lo cual es una ventaja sobre los filtros robustos que usan desigualdad matricial lineal (LMI). Además, se establece en [40] que el filtro  $H_2$ -FIR puede funcionar de manera similar al  $H_\infty$  robustos y filtros de energía a pico, que es una ventaja adicional. Por otro lado, los diseños conocidos de  $H_2$ -FIR se obtuvieron como filtros sesgados (no óptimos), tomando en cuenta las perturbaciones e ignorando la medición y errores iniciales. Dado que todos estos efectos pueden ocurrir en el simultáneamente durante las tareas de seguimiento se requieren algoritmos avanzados. Para superar los inconvenientes antes mencionados, se propuso un enfoque mejorado de estimación del estado  $H_2$ -FIR en [71]. Sugiere derivar la ganancia para el estado del estimador por lotes  $H_2$ -FIR minimizando la norma cuadrada de Frobenius de la función de transferencia ponderada, donde los pesos son vinculados a cada uno de los factores de impacto. El enfoque ha sido aplicado en [71] al modelo de estado basado en el método de Euler directo (FE) que ha resultado en el predictor  $H_2$ -FIR óptimo sin sesgo ( $H_2$ -OUFIR). Poco después, el predictor  $H_2$ -OFIR basado en FE se derivó y aplicó a procesos industriales en [72]. Dado que la predicción no siempre es requerida, el modelo de espacio de estados con la forma de Euler hacia atrás (BE) se empleó en [38] para derivar el filtro  $H_2$ -OUFIR *a posteriori* y resolver el problema de seguimiento de objetos. Vale la pena señalar que los modelos en espacio de estados basados en la forma FE y BE generan cuatro estimadores de estado  $H_2$ -FIR robustos inconvertibles: predictor  $H_2$ -OUFIR, predictor  $H_2$ -OFIR, Filtro  $H_2$ -OUFIR y filtro  $H_2$ -OFIR. En ambientes hostiles, la tarea de seguimiento es desarrollada usando algoritmos robustos, pues dadas las condiciones ambientales se asume la presencia de perturbaciones, errores iniciales y de medición. En este capítulo se desarrollará un filtro FIR ( $H_2$ -OUFIR) *a posteriori* óptimo, robusto. Se implementará en un espacio de estados en tiempo discreto. La derivación del filtro será empleando el ya conocido filtro de respuesta infinita al impulso ( $H_2$ -FIR), la estimación del estado se realizara mediante la minimización de la norma de Frobenius de una función de transferencia ponderada. El filtro robusto  $H_2$ -OFIR está diseñado para bloques de matrices de error, y sus formas recursivas se muestran para matrices de errores diagonales. También se presenta el algoritmo de filtrado subóptimo  $H_2$ -FIR utilizando desigualdades de matrices lineales.

## 4.2. Modelo y Formulación del problema

Considerando un sistema lineal invariante en el tiempo (LTI) representado en un espacio de estados en la forma Backward-Euler (BE) se obtiene el modelo siguiente:

$$\begin{aligned}\mathbf{x}_k &= \mathbf{F}\mathbf{x}_{k-1} + \mathbf{E}\mathbf{u}_k + \mathbf{B}\mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{D}\mathbf{w}_k + \mathbf{v}_k\end{aligned}\quad (4.1)$$

Donde  $\mathbf{x}_k \in \mathbb{R}^K$  es el vector de estados del sistema,  $\mathbf{u}_k \in \mathbb{R}^L$  es el vector de entrada,  $\mathbf{y}_k \in \mathbb{R}^P$  es el vector de salida del sistema, y las matrices son  $\mathbf{F} \in \mathbb{R}^{K \times K}$ ,  $\mathbf{E} \in \mathbb{R}^{K \times L}$ ,  $\mathbf{B} \in \mathbb{R}^{K \times M}$ ,  $\mathbf{H} \in \mathbb{R}^{P \times K}$  y  $\mathbf{D} \in \mathbb{R}^{P \times M}$ . La perturbación  $\mathbf{w}_k \in \mathbb{R}^M$  y el error de medición  $\mathbf{v}_k \in \mathbb{R}^P$  se asumen no obligatoriamente gaussianos, de media cero, sin correlación y con matrices de error  $\mathbf{Q} = E\{w_k w_k^T\}$  y  $\mathbf{R} = E\{v_k v_k^T\}$  con una norma acotada, donde  $E\{z\}$  representa la media de  $z$ .

Para obtener un filtro robusto  $H_2$ -OFIR el sistema 4.1 se debe generar un modelo extendido en espacio de estados sobre un horizonte  $[m, k]$  de  $N$  puntos, desde  $m = k - N + 1$  a  $k$  como sigue

$$\mathbf{x}_k = \mathbf{F}^{N-1}\mathbf{x}_m + \bar{\mathbf{S}}_N \mathbf{U}_{m,k} + \bar{\mathbf{D}}_N \mathbf{W}_{m,k}, \quad (4.2)$$

$$\mathbf{Y}_{m,k} = \mathbf{H}_N \mathbf{x}_m + \mathbf{L}_N \mathbf{U}_{m,k} + \mathbf{T}_N \mathbf{W}_{m,k} + \mathbf{V}_{m,k}, \quad (4.3)$$

Los vectores aumentados se definen como:

$$\begin{aligned}\mathbf{U}_{m,k} &= [\mathbf{u}_m^T \ \mathbf{u}_{m+1}^T \ \cdots \ \mathbf{u}_k^T]^T, \\ \mathbf{W}_{m,k} &= [\mathbf{w}_m^T \ \mathbf{w}_{m+1}^T \ \cdots \ \mathbf{w}_k^T]^T, \\ \mathbf{V}_{m,k} &= [\mathbf{v}_m^T \ \mathbf{v}_{m+1}^T \ \cdots \ \mathbf{v}_k^T]^T, \\ \mathbf{Y}_{m,k} &= [\mathbf{y}_m^T \ \mathbf{y}_{m+1}^T \ \cdots \ \mathbf{y}_k^T]^T\end{aligned}$$

y las matrices aumentadas se definen como:

$$\begin{aligned}\mathbf{H}_N &= \bar{\mathbf{H}}_N \mathbf{F}_N, \\ \mathbf{L}_N &= \bar{\mathbf{H}}_N \mathbf{S}_N, \\ \mathbf{T}_N &= \mathbf{G}_N + \bar{\mathbf{T}}_N, \\ \mathbf{G}_N &= \bar{\mathbf{H}}_N \mathbf{D}_N, \\ \bar{\mathbf{H}}_N &= \text{diag}(\mathbf{H} \ \mathbf{H} \ \cdots \ \mathbf{H}), \\ \bar{\mathbf{T}}_N &= \text{diag}(\mathbf{D} \ \mathbf{D} \ \cdots \ \mathbf{D}), \\ \mathbf{F}_N &= [\mathbf{I} \ \mathbf{F}^T \ \cdots \ (\mathbf{F}^{N-2})^T \ (\mathbf{F}^{N-1})^T]^T, \\ \mathbf{S}_N &= \begin{bmatrix} \mathbf{E} & 0 & \cdots & 0 & 0 \\ \mathbf{F}\mathbf{E} & \mathbf{E} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{F}^{N-2}\mathbf{E} & \mathbf{F}^{N-3}\mathbf{E} & \cdots & \mathbf{E} & 0 \\ \mathbf{F}^{N-1}\mathbf{E} & \mathbf{F}^{N-2}\mathbf{E} & \cdots & \mathbf{F}\mathbf{E} & \mathbf{E} \end{bmatrix},\end{aligned}$$

Al reemplazar la matriz  $\mathbf{E}$  por  $\mathbf{B}$  en  $\mathbf{S}_N$  se obtiene la matriz  $\mathbf{D}_N$ . La matriz  $\bar{\mathbf{S}}_N$  es el último vector renglón de la matriz  $\mathbf{S}_N$ , las matrices  $\bar{\mathbf{D}}_N$  y  $\mathbf{D}_N$  guardan esa misma relación. Dado el sistema extendido en 4.2 y 4.3, definimos la estimación para el filtro FIR basado en convolución como:

$$\begin{aligned}\hat{\mathbf{x}}_k &= \mathcal{H}_N \mathbf{Y}_{m,k} + \bar{\mathcal{H}}_N \mathbf{U}_{m,k}, \\ &= \mathcal{H}_N \mathbf{H}_N x_m + (\mathcal{H}_N \mathbf{L}_N + \bar{\mathcal{H}}_N) \mathbf{U}_{m,k} + \mathcal{H}_N \mathbf{T}_N \mathbf{W}_{m,k} + \mathcal{H}_N \mathbf{V}_{m,k}\end{aligned}\quad (4.4)$$

Las matrices  $\mathcal{H}_N$  y  $\bar{\mathcal{H}}_N$  corresponden, respectivamente, a las ganancias homogénea y forzada del estimador. Aplicando la restricción de sesgo  $E\{\mathbf{x}_k\} = E\{\hat{\mathbf{x}}_k\}$  al espacio de estados extendido y a la ecuación 4.4 se obtienen lo siguiente:

$$\begin{aligned}\mathcal{H}_N \mathbf{H}_N x_m + (\mathcal{H}_N \mathbf{L}_N + \bar{\mathcal{H}}_N) \mathbf{U}_{m,k} &= \mathbf{F}^{N-1} x_m + \bar{\mathbf{S}}_N \mathbf{U}_{m,k} \\ (\mathbf{F}^{N-1} - \mathcal{H}_N \mathbf{H}_N) x_m + (\bar{\mathbf{S}}_N - \mathcal{H}_N \mathbf{L}_N - \bar{\mathcal{H}}_N) \mathbf{U}_{m,k} &= 0\end{aligned}$$

Similar al caso es predictor, empleando las propiedades de linealidad del sistema, la respuesta total puede ser determinada por superposición de la solución homogénea ( $\mathbf{U}_{m,k} = \mathbf{0}$ ) y la forzada ( $x_m = \mathbf{0}$ ), obteniendo:

$$\mathbf{I} = \mathcal{H}_N \mathbf{C}_N, \quad (4.5)$$

$$\bar{\mathcal{H}}_N = \bar{\mathbf{S}}_N - \mathcal{H}_N \mathbf{L}_N. \quad (4.6)$$

Donde  $\mathbf{C}_N = \mathbf{H}_N \mathbf{F}^{-(N-1)}$ , definimos el error de estimación como  $\varepsilon_k = x_k - \hat{x}_k$ , para finalmente obtener

$$\begin{aligned}\varepsilon_k &= (\mathbf{F}^{N-1} - \mathcal{H}_N \mathbf{H}_N) x_m + (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N) \mathbf{W}_{m,k} - \mathcal{H}_N \mathbf{V}_{m,k} \\ &= \mathcal{B}_N x_m + \mathcal{W}_N \mathbf{W}_{m,k} - \mathcal{V}_N \mathbf{V}_{m,k}\end{aligned}\quad (4.7)$$

Definiremos la matriz residual de sesgo  $\mathcal{B}_N$ , la matriz de error residual del sistema  $\mathcal{W}_N$ , y la matriz de error residual de observación  $\mathcal{V}_N$  como

$$\begin{aligned}\mathcal{B}_N &= \mathbf{F}^{N-1} - \mathcal{H}_N \mathbf{H}_N, \\ \mathcal{W}_N &= \bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N, \\ \mathcal{V}_N &= \mathcal{H}_N\end{aligned}\quad (4.8)$$

### 4.3. Filtrado $H_2$ -FIR estándar

Los errores de estimación son reducidos mediante la minimización de la traza del cuadrado de la norma de Frobenius  $\|\mathcal{T}(z)\|_F^2$  de la función de transferencia

perturbación-error  $\mathcal{T}(z)$  promediada en todas las frecuencias [27]. Para el filtrado FIR se sugiere obtener la ganancia  $\mathcal{H}_N$  para el filtro FIR a partir de  $\mathcal{T}(z)$  empleando  $u_k = 0$  y  $v_k = 0$  y llevando el modelo extendido a la forma:

$$\mathbf{x}_k = \mathbf{F}^{N-1} \mathbf{x}_m + \bar{\mathbf{D}}_N \mathbf{W}_{m,k}, \quad (4.9)$$

$$\mathbf{Y}_{m,k} = \mathbf{H}_N \mathbf{x}_m + \mathbf{T}_N \mathbf{W}_{m,k}, \quad (4.10)$$

La matriz  $\mathbf{W}_{m,k}$  puede ser representada de forma recursiva [35] como

$$\mathbf{W}_{m,k} = \mathbf{A}_w \mathbf{W}_{m-1,k-1} + \mathbf{B}_w \mathbf{w}_k \quad (4.11)$$

Usando las siguientes matrices dispersas auxiliares [64]

$$\mathbf{A}_w = \begin{bmatrix} 0 & \mathbf{I} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{I} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{I} \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad \mathbf{B}_w = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{I} \end{bmatrix} \quad (4.12)$$

Para derivar la matriz de función de transferencia  $\mathcal{T}(z)$  de  $\mathbf{w}$  a  $\varepsilon$  se considera que 4.11 y 4.7 forman un sistema MIMO discreto como:

$$\begin{aligned} \mathbf{W}_{m,k} &= \mathbf{A}_w \mathbf{W}_{m-1,k-1} + \mathbf{B}_w \mathbf{w}_k \\ \varepsilon_k &= \mathcal{W}_N \mathbf{W}_{m,k} + \mathcal{B}_N x_m \end{aligned} \quad (4.13)$$

Se aplica la transformada  $z$ , obteniendo

$$\begin{aligned} \mathbf{W}(z) &= \mathbf{A}_w \mathbf{W}(z) z^{-1} + \mathbf{B}_w \mathbf{w}(z) \\ \varepsilon(z) &= \mathcal{W}_N \mathbf{W}(z) + \mathcal{B}_N x_m(z) \end{aligned}$$

Al despejar  $\mathbf{W}(z)$  y reemplazar en la ecuación de error se obtiene

$$\begin{aligned} (z\mathbf{I} - \mathbf{A}_w) \mathbf{W}(z) &= z\mathbf{B}_w \mathbf{w}(z) \\ \mathbf{W}(z) &= (z\mathbf{I} - \mathbf{A}_w)^{-1} z\mathbf{B}_w \mathbf{w}(z) \\ \varepsilon(z) &= \mathcal{W}_N (z\mathbf{I} - \mathbf{A}_w)^{-1} z\mathbf{B}_w \mathbf{w}(z) + \mathcal{B}_N x_m(z) \\ \varepsilon(z) &= \mathcal{T}(z) \mathbf{w}(z) + \mathcal{B}_N x_m(z) \\ \mathcal{T}(z) &= \mathcal{W}_N (\mathbf{I}z - \mathbf{A}_w)^{-1} z\mathbf{B}_w \end{aligned} \quad (4.14)$$

Sabiendo que la norma de Frobenius se define como

$$\|\mathcal{T}(z)\|_F^2 = \frac{1}{2\pi} \int_0^{2\pi} \text{tr} [\mathcal{T}(e^{j\omega\tau}) \mathcal{T}^*(e^{j\omega\tau})] d\omega\tau \quad (4.15)$$

Donde  $\tau$  corresponde al tiempo de muestreo y  $\mathcal{J}^*(z)$  es la transpuesta conjugada de  $\mathcal{J}(z)$ , y que para la magnitud de la función de transferencia  $G(z)$  está dada por

$$\|\mathcal{J}(z)\|_2 = \sqrt{\text{tr}(\mathcal{W}_N \mathbf{L}_c \mathcal{W}_N^T)} \quad (4.16)$$

Donde  $\mathbf{L}_c$  es el Gramiano de controlabilidad dado por

$$\mathbf{L}_c = \sum_{i=0}^{\infty} \mathbf{A}_w^i \mathbf{B}_w \mathbf{B}_w^T (\mathbf{A}_w^T)^i \quad (4.17)$$

que surge de la solución a la ecuación de Lyapunov  $\mathbf{A}_w \mathbf{L}_c + \mathbf{L}_c \mathbf{A}_w^T + \mathbf{B}_w \mathbf{B}_w^T = 0$ , debido a que la  $\mathbf{A}_w$  es una matriz de desplazamiento superior (*upper shift-matrix*, USM), esta se vuelve nilpotente  $\mathbf{A}_w^k = 0, \forall k \geq N$  donde  $N$  es la dimensión de la matriz, la ecuación 4.17 se reduce a

$$\mathbf{L}_c = \sum_{i=0}^{N-1} \mathbf{A}_w^i \mathbf{B}_w \mathbf{B}_w^T (\mathbf{A}_w^T)^i \quad (4.18)$$

El producto  $\mathbf{B}_w \mathbf{B}_w^T$  da como resultado una matriz  $\mathbf{B}_c = \text{diag}\{\mathbf{0}, \dots, \mathbf{I}_{K \times K}\}$ , la operación  $\mathbf{S}^T \mathbf{A} \mathbf{S}$  donde  $\mathbf{S}$  es una matriz de desplazamiento inferior genera una permutación en la que  $\mathbf{A}$  es desplazada arriba y a la izquierda a lo largo de la diagonal principal. Debido a lo anterior podemos afirmar que la permutación  $\mathbf{A}_w^i \mathbf{B}_c (\mathbf{A}_w^T)^i$  corresponde a un desplazamiento cíclico de  $\mathbf{B}_c$ . Debido a la forma en bloques de las matrices  $\mathbf{A}_w$  y  $\mathbf{B}_w$ , cada permutación desplazará a la matriz  $\mathbf{B}_c$  en  $K$  renglones y  $K$  columna. Para  $\mathbf{B}_c \in \mathbb{R}^{NK \times NK}$  podemos concluir que la suma de todas las permutaciones sucesivas de  $\mathbf{B}_c$  en la ec. 4.18 resultan en  $\mathbf{L}_c = \mathbf{I} \in \mathbb{R}^{KN \times KN}$ . Sustituyendo 4.8 y 4.18 en 4.16 se obtiene

$$\begin{aligned} \|\mathcal{J}(z)\|_F^2 &= \text{tr}(\mathcal{W}_N \mathcal{W}_N^T) \\ &= \text{tr}[(\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)(\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)^T]. \end{aligned}$$

Es posible minimizar  $\|\mathcal{J}(z)\|_F^2$  con respecto de  $\mathcal{H}_N$  bajo la restricción de sesgo empleando multiplicadores de Lagrange construyendo el Lagrangiano siguiente

$$J = \text{tr}[(\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)(\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)^T] + \text{tr} \Delta (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) \quad (4.19)$$

$$\begin{aligned} \frac{\partial}{\partial \mathcal{H}_N} \left( \text{tr}[(\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)(\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)^T] + \text{tr} \Delta (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) \right) &= 0 \\ \frac{\partial}{\partial \mathcal{H}_N} \text{tr} \left( \bar{\mathbf{D}}_N \bar{\mathbf{D}}_N^T - \mathcal{H}_N \mathbf{T}_N \bar{\mathbf{D}}_N^T - \bar{\mathbf{D}}_N \mathbf{T}_N^T \mathcal{H}_N^T + \mathcal{H}_N \mathbf{T}_N \mathbf{T}_N^T \mathcal{H}_N^T \right) &+ \\ \frac{\partial}{\partial \mathcal{H}_N} \text{tr} \Delta (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) &= 0, \\ -2\bar{\mathbf{D}}_N \mathbf{T}_N^T + \mathcal{H}_N (\mathbf{T}_N^T \mathbf{T}_N + \mathbf{T}_N \mathbf{T}_N^T) - \Delta^T \mathbf{C}_N^T &= 0, \\ -2\bar{\mathbf{D}}_N \mathbf{G}_N^T + 2\mathcal{H}_N (\mathbf{G}_N \mathbf{G}_N^T) - \Delta^T \mathbf{C}_N^T &= 0, \\ \frac{\partial}{\partial \Delta} \left( \text{tr}[(\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)(\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)^T] + \text{tr} \Delta (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) \right) &= 0 \end{aligned}$$

Despejando  $\Delta$

$$\begin{aligned} -2\bar{\mathbf{D}}_N \mathbf{G}_N^T + 2\mathcal{H}_N (\mathbf{G}_N \mathbf{G}_N^T) - \Delta^T \mathbf{C}_N^T &= 0, \\ \Delta^T \mathbf{C}_N^T &= 2(\mathcal{H}_N \Omega_N - \bar{\mathbf{D}}_N \mathbf{G}_N^T), \\ \mathbf{C}_N \Delta &= 2(\Omega_N \mathcal{H}_N^T - \mathbf{G}_N \bar{\mathbf{D}}_N^T). \end{aligned} \quad (4.20)$$

En el que  $\Omega_N = \mathbf{G}_N \mathbf{G}_N^T$ , premultiplicando por  $\mathbf{C}_N^T \Omega_N^{-1}$

$$\begin{aligned} \mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N \Delta &= 2\mathbf{C}_N^T \Omega_N^{-1} (\Omega_N \mathcal{H}_N^T - \mathbf{G}_N \bar{\mathbf{D}}_N^T), \\ \Delta &= 2(\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Omega_N^{-1} (\Omega_N \mathcal{H}_N^T - \mathbf{G}_N \bar{\mathbf{D}}_N^T), \\ \Delta &= 2(\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} (\mathbf{C}_N^T \mathcal{H}_N^T - \mathbf{C}_N^T \Omega_N^{-1} \mathbf{G}_N \bar{\mathbf{D}}_N^T). \end{aligned} \quad (4.21)$$

La derivada parcial para  $\Delta$

$$\begin{aligned} \frac{\partial}{\partial \Delta} \left( \text{tr}[(\bar{\mathbf{D}}_N - \mathcal{H}_N T_N)(\bar{\mathbf{D}}_N - \mathcal{H}_N T_N)^T] + \text{tr} \Delta (\mathbf{I} - \mathcal{H}_N \mathbf{C}_N) \right) &= 0, \\ \mathbf{I} - \mathbf{C}_N^T \mathcal{H}_N^T &= 0 \end{aligned} \quad (4.22)$$

Sustituyendo la ec. 4.22 en la ec. 4.21

$$\Delta = 2(\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} (\mathbf{I} - \mathbf{C}_N^T \Omega_N^{-1} \mathbf{G}_N \bar{\mathbf{D}}_N^T). \quad (4.23)$$

Reemplazando la ec. 4.23 en la ec. 4.20 se obtiene

$$\begin{aligned} -\bar{\mathbf{D}}_N \mathbf{G}_N^T + \mathcal{H}_N \Omega_N - (\mathbf{I} - \mathbf{C}_N^T \Omega_N^{-1} \mathbf{G}_N \bar{\mathbf{D}}_N^T)^T (\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-T} \mathbf{C}_N^T &= 0, \\ \mathcal{H}_N &= \bar{\mathbf{D}}_N \mathbf{G}_N^T \Omega_N^{-1} + (\mathbf{I} - \bar{\mathbf{D}}_N \mathbf{G}_N^T \Omega_N^{-1} \mathbf{C}_N) (\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Omega_N^{-1}, \\ \mathcal{H}_N &= (\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Omega_N^{-1} + \\ &\quad \bar{\mathbf{D}}_N \mathbf{G}_N^T \Omega_N^{-1} (\mathbf{I} - \mathbf{C}_N (\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Omega_N^{-1}) \end{aligned}$$

Aplicando  $(\mathbf{ABC}) = \mathbf{C}^{-1} \mathbf{B}^{-1} \mathbf{A}^{-1}$  la ecuación anterior se reduce a

$$\mathcal{H}_N = (\mathbf{C}_N^T \Omega_N^{-1} \mathbf{C}_N)^{-1} \mathbf{C}_N^T \Omega_N^{-1} \quad (4.24)$$

Este resultado puede resultar impreciso, porque la matriz  $\Omega_N$  no tiene en consideración la información estadística de  $w_k$ . Es decir, usar solamente la minimización de la norma de Frobenius de  $\mathcal{T}(z)$  no garantiza la optimalidad del filtro  $H_2$ -FIR, a fin de minimizar el error se requiere incluir un factor de peso para  $\mathcal{T}(z)$  como se muestra en [71]. Ahora el problema es introducir una norma ponderada para la función de transferencia de perturbación a error  $\tilde{\mathcal{T}}(z)$  minimizando el cuadrado de la norma de Frobenius y obtener así la ganancia para el filtro *a posteriori*  $H_2$ -OFIR. Este filtro puede obtener un estimado robusto para perturbaciones y errores no gaussianos y es óptimo bajo un condiciones gaussianas. Además se presenta una forma recursiva para el ruido gaussiano desarrollando un filtro subóptimo  $H_2$ -FIR usando LMI.

## 4.4. Filtrado Robusto $H_2$ -OFIR

Para obtener un filtro *a posteriori*  $H_2$ -OFIR para sistemas con perturbaciones bajo condiciones iniciales y errores de datos primero incluimos la restricción 4.6 en 4.7 y definimos los errores como

$$\begin{aligned}\varepsilon_{xk} &= \mathbf{B}_N x_m, \\ \varepsilon_{wk} &= \mathbf{W}_N \mathbf{W}_{m,k}, \\ \varepsilon_{vk} &= -\mathcal{V}_N \mathbf{V}_{m,k}.\end{aligned}\tag{4.25}$$

Debido a que el error inicial  $\varepsilon_x$  se transfiere íntegramente al error total  $\varepsilon$ , la función de transferencia  $\mathcal{T}_x(z) = \mathbf{I}$ . La función de transferencia  $(\varepsilon_x - \varepsilon)$  de error inicial a error de salida  $\mathcal{T}_w(z)$  puede ser escrita como 4.14. De forma similar a la representación de  $\mathbf{W}_{m,k}$  en 4.13,  $\mathbf{V}_{m,k}$  se puede representar con las matrices dispersas en 4.12 como

$$\mathbf{V}_{m,k} = \mathbf{A}_w \mathbf{V}_{m-1,k-1} + \mathbf{B}_w v_k\tag{4.26}$$

y la función de transferencia  $(\varepsilon_v - \varepsilon)$  de error de medición a error total  $\mathcal{T}_v(z)$  está dada por

$$\mathcal{T}_v(z) = \mathcal{V}_N (\mathbf{I}z - \mathbf{A}_w)^{-1} z \mathbf{B}_w\tag{4.27}$$

El filtro  $H_2$ -OFIR puede diseñarse como sigue

### 4.4.1. Filtrado Robusto $H_2$ OFIR en batch

Tomando [71] como referencia, primero introduciremos la función de transferencia ponderada  $\bar{\mathcal{T}}(z) = \mathcal{T}(z)\zeta_k$ , se desea encontrar el factor de peso  $\zeta_k$  tal que las propiedades de perturbación y error de medición sean transferidas al error de salida. A continuación, definimos la norma cuadrada de Frobenius de  $\bar{\mathcal{T}}(z)$  sobre las variables  $zy$   $k$  como:

$$\begin{aligned}\|\bar{\mathcal{T}}(z)\|_F^2 &= E_z \{ E_k \{ \text{tr}[\mathcal{T}(z)\zeta_k \zeta_k^* \mathcal{T}^*(z)] \} \} \\ &= \frac{1}{2\pi} \int_0^{2\pi} \text{tr}[\mathcal{T}(e^{j\omega\tau}) \Theta \mathcal{T}^*(e^{j\omega\tau})] d\omega\tau\end{aligned}\tag{4.28}$$

Donde  $\Theta = E\{\zeta_k \zeta_k^*\}$ , la minimización de 4.28 permite calcular la ganancia  $\mathcal{H}_N$  para el filtro robusto  $H_2$ -OFIR. Los pesos para los errores  $\varepsilon_{xk}, \varepsilon_{wk}, \varepsilon_{vk}$  pueden tomarse de la

ecuación de error 4.7 y así obtener

$$\|\bar{\mathcal{J}}_x(z)\|_F^2 = \frac{1}{2\pi} \int_0^{2\pi} \text{tr} [\mathcal{B}_N \chi_m \mathcal{B}_N^T] d\omega\tau \quad (4.29)$$

$$\|\bar{\mathcal{J}}_w(z)\|_F^2 = \frac{1}{2\pi} \int_0^{2\pi} \text{tr} [\mathcal{J}_w(e^{j\omega\tau}) \mathcal{Q}_N \times \mathcal{J}_w^*(e^{j\omega\tau})] d\omega\tau \quad (4.30)$$

$$\|\bar{\mathcal{J}}_v(z)\|_F^2 = \frac{1}{2\pi} \int_0^{2\pi} \text{tr} [\mathcal{J}_v(e^{j\omega\tau}) \mathcal{R}_N \times \mathcal{J}_v^*(e^{j\omega\tau})] d\omega\tau. \quad (4.31)$$

Cabe destacar que las matrices de error  $\mathcal{Q}_N$  y  $\mathcal{R}_N$  tienen otro significado diferente a la covarianza del ruido en un modelo Gaussiano, a pesar de esto, se asume que estas son positivas definidas y simétricas. De la ecuación 4.29 obtenemos

$$\|\bar{\mathcal{J}}_x(z)\|_F^2 = \text{tr}(\mathcal{B}_N \chi_m \mathcal{B}_N^T). \quad (4.32)$$

Para la ecuación 4.30, se fija  $\mathcal{Q}_N = \mathbf{I}$  [27] y se reescribe como

$$\|\bar{\mathcal{J}}_w(z)\|_F^2 = \text{tr}(\mathcal{W}_N \mathbf{L} \mathcal{W}_N^T) \quad (4.33)$$

Donde una matriz simétrica positiva definida  $\mathbf{L}$  es una solución a la ecuación de Lyapunov en tiempo discreto

$$\mathbf{A}_w \mathbf{L} \mathbf{A}_w^T + \mathbf{M} = \mathbf{L}. \quad (4.34)$$

En la ec. 4.34,  $\mathbf{M}$  se asume como cualquier matriz positiva definida. Se elige  $\mathbf{M} = \mathbf{B}_w \mathcal{Q}_N \mathbf{B}_w^T$  para representar la solución a la ecuación anterior [45] como

$$\mathbf{L} = \sum_{i=0}^{\infty} \mathbf{A}_w^i \mathbf{B}_w \mathcal{Q}_N \mathbf{B}_w^T \mathbf{A}_w^{T i}. \quad (4.35)$$

Para  $\mathbf{A}_w$  y  $\mathbf{B}_w$  dispersas, la ecuación 4.35 se convierte en  $\mathbf{L} = \mathcal{Q}_N$  y es posible representar  $\|\bar{\mathcal{J}}_w(z)\|_F^2$  con

$$\|\bar{\mathcal{J}}_w(z)\|_F^2 = \text{tr}(\mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T). \quad (4.36)$$

Similarmente, representamos  $\|\bar{\mathcal{J}}_v(z)\|_F^2$  como

$$\|\bar{\mathcal{J}}_v(z)\|_F^2 = \text{tr}(\mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T). \quad (4.37)$$

Usando las ecuaciones (4.32),(4.36) y (4.37) finalmente podemos transformar la traza de la matriz de error  $\mathbf{P}$  como

$$\begin{aligned} \text{tr} \mathbf{P} &= E\{(\varepsilon_{xk} + \varepsilon_{wk} + \varepsilon_{vk})^T(\dots)\} \\ &= E\{\varepsilon_{xk}^T \varepsilon_{xk}\} + E\{\varepsilon_{wk}^T \varepsilon_{wk}\} + E\{\varepsilon_{vk}^T \varepsilon_{vk}\} \\ &= \|\bar{\mathcal{J}}_x(z)\|_F^2 + \|\bar{\mathcal{J}}_w(z)\|_F^2 + \|\bar{\mathcal{J}}_v(z)\|_F^2, \\ &= \text{tr}(\mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T) \end{aligned} \quad (4.38)$$

La ganancia  $\mathcal{H}_N$  para el filtro *a posteriori* robusto  $H_2$ -OFIR puede ser obtenida a través del siguiente problema de minimización

$$\mathcal{H}_N = \arg \min_{\mathcal{H}_N} \text{tr}(\mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T) \quad (4.39)$$

Donde las matrices de error residual  $\mathcal{B}_N$ ,  $\mathcal{W}_N$  y  $\mathcal{V}_N$  están previamente definidas como funciones de  $\mathcal{H}_N$ . La solución a (4.39) se puede encontrar resolviendo

$$\frac{\partial}{\partial \mathcal{H}_N} \text{tr}(\mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T) = 0$$

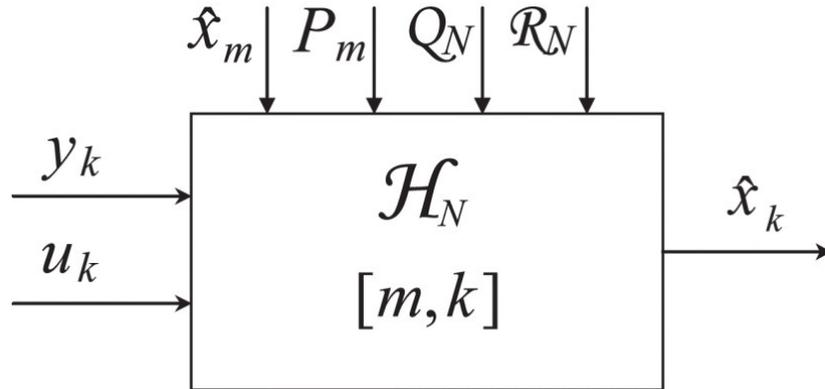
Aplicando la derivada se llega a lo siguiente

$$\mathcal{H}_N (\mathbf{H}_N \chi_m \mathbf{H}_N^T + \mathbf{T}_N \mathcal{Q}_N \mathbf{T}_N^T + \mathcal{R}_N) = \mathbf{F}^{N-1} \chi_m \mathbf{H}_N^T + \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{T}_N^T.$$

El filtro  $H_2$  fué desarrollado para sistemas estables, por lo que  $\mathbf{D} = 0$  y  $\mathbf{T}_N = \mathbf{G}_N$ . Sustituyendo esto en la ecuación anterior llegamos a

$$\mathcal{H}_N = (\mathbf{F}^{N-1} \chi_m \mathbf{H}_N^T + \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T) (\mathbf{H}_N \chi_m \mathbf{H}_N^T + \mathbf{G}_N \mathcal{Q}_N \mathbf{G}_N^T + \mathcal{R}_N)^{-1} \quad (4.40)$$

La ecuación (4.40) tiene la misma estructura que la encontrada para el filtro OFIR *a posteriori* [70]. La ganancia para este filtro será robusta si ajustamos  $\mathcal{Q}_N$  y  $\mathcal{R}_N$  para los máximos errores y perturbaciones permisibles, y será óptimo cuando éstos sean derivados de procesos blancos gaussianos.



**Figura 4.1.** Estructura general de un filtro robusto  $H_2$ -OFIR

La estructura general para el filtro *a posteriori* robusto  $H_2$  se muestra en la figura 4.1. Este filtro requiere datos de observación  $y_k$  y la señal de entrada  $u_k$ , ambos en el horizonte de estimación  $[m, k]$ . El estado inicial de la estimación  $\hat{x}_m$ , las matrices del sistema en espacio de estados, la matriz de error  $\mathbf{P}_m$  y las matrices de error en bloque  $\mathcal{Q}_N$  y  $\mathcal{R}_N$  son variables necesarias para calcular la ganancia  $\mathcal{H}_N$ . La estimación esta dada por

$$\hat{x}_k = \mathcal{H}_N \mathbf{Y}_{m,k} + (\bar{\mathbf{S}}_N - \mathcal{H}_N \mathbf{L}_N) \mathbf{U}_{m,k} \quad (4.41)$$

Y la matriz de error se calcula usando

$$\mathbf{P}_k = \mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T \quad (4.42)$$

Cabe mencionar que para matrices de error  $\mathcal{Q}_N$  y  $\mathcal{R}_N$  arbitrarias no es posible calcular (4.41) y (4.42) de forma iterativa. Pero se pueden reducir a esta forma si las matrices son diagonales.

#### 4.4.2. Forma recursiva para $\mathcal{Q}_N$ y $\mathcal{R}_N$ diagonales

Los algoritmos recursivos para los filtros  $H_2$  han sido considerados en [21, 23, 27, 61]. La forma recursiva para el filtro *a posteriori*  $H_2$ -OFIR con matrices diagonales  $\mathcal{Q}_N$  y  $\mathcal{R}_N$ , puede ser calculada como enuncia el siguiente teorema

**Teorema 4.4.1.** *Dado el modelo estable en (4.2) y (4.3) con matrices aumentadas de error en bloque  $\mathcal{Q}_N$  y  $\mathcal{R}_N$  con forma diagonal, la estimación y la covarianza del error del filtro a posteriori  $H_2$ -OFIR pueden ser calculados de forma iterativa en un horizonte  $[m, k]$  para un  $\hat{x}_m$  y  $\mathbf{P}_m$  usando las recursiones de Kalman*

$$\mathbf{P}_i^- = \mathbf{F} \mathbf{P}_{i-1}^- \mathbf{F}^T + \mathbf{B} \mathbf{Q} \mathbf{B}^T \quad (4.43)$$

$$\mathbf{S}_i = \mathbf{H} \mathbf{P}_i^- \mathbf{H}^T + \mathbf{R} \quad (4.44)$$

$$\hat{x}_i^- = \mathbf{F} \hat{x}_{i-1}^- + \mathbf{E} u_i \quad (4.45)$$

$$\mathbf{K}_i = \mathbf{P}_i^- \mathbf{H}^T \mathbf{S}_i^{-1} \quad (4.46)$$

$$\hat{x}_i = \hat{x}_i^- + \mathbf{K}_i (y_i - \mathbf{H} \hat{x}_i^-) \quad (4.47)$$

$$\mathbf{P}_i = (\mathbf{I} - \mathbf{K}_i \mathbf{H}_i) \mathbf{P}_i^- \quad (4.48)$$

Cambiando el la iteración en el tiempo  $i$  desde  $m + 1$  hasta  $k$  y tomando como salida del algoritmo la estimación en  $i = k$ .

*Demostración.* Considere los modelos (4.2) y (4.3) y reasigne  $\mathbf{W}_k \triangleq \mathbf{W}_{m,k}$ ,  $\mathbf{V}_k \triangleq \mathbf{V}_{m,k}$ ,  $\mathbf{Y}_k \triangleq \mathbf{Y}_{m,k}$ ,  $\mathbf{H}_k \triangleq \mathbf{H}_N$ ,  $\mathbf{G}_k \triangleq \mathbf{G}_N$ ,  $\mathbf{F}_k \triangleq \mathbf{F}_N$ ,  $\mathbf{D}_k \triangleq \mathbf{D}_N$ ,  $\mathbf{S}_k \triangleq \mathbf{S}_N$ ,  $\mathcal{H}_k \triangleq \mathcal{H}_N$ . Representaremos la ganancia (4.40) del filtro como  $\mathcal{H}_k = \mathcal{H}_k^a \mathcal{H}_k^b$ , donde

$$\mathcal{H}_k^a = \mathbf{F}^k \chi_0 \mathbf{H}_k^T + \bar{\mathbf{D}}_k \mathcal{Q}_k \mathbf{G}_k^T \quad (4.49)$$

$$\mathcal{H}_k^b = (\mathbf{H}_k \chi_0 \mathbf{H}_k^T + \mathbf{G}_k \mathcal{Q}_k \mathbf{G}_k^T + \mathcal{R}_k)^{-1} \quad (4.50)$$

Se asigna  $\chi_0 = E\{x_0 x_0^T\}$  y se busca una forma recursiva para (4.49) y (4.50). De acuerdo a [73], se usará la descomposición  $\mathcal{Q}_k = \text{diag}(\mathcal{Q}_{k-1}, \mathcal{Q})$

$$\begin{aligned} \bar{\mathbf{D}}_k &= [\mathbf{F} \bar{\mathbf{D}}_{k-1} \quad \mathbf{B}] \\ \mathbf{G}_k &= \begin{bmatrix} \mathbf{G}_{k-1} & 0 \\ \mathbf{H} \mathbf{F} \bar{\mathbf{D}}_{k-1} & \mathbf{H} \mathbf{B} \end{bmatrix} \\ \bar{\mathbf{D}}_k \mathcal{Q}_k \bar{\mathbf{D}}_k^T &= \mathbf{F} \bar{\mathbf{D}}_{k-1} \mathcal{Q}_{k-1} \bar{\mathbf{D}}_{k-1}^T \mathbf{F}^T + \mathbf{B} \mathbf{Q} \mathbf{B}^T \end{aligned}$$

Y se representa  $\mathcal{H}_k^a$  de forma recursiva como

$$\begin{aligned}\mathcal{H}_k^a &= \mathbf{F}^k \chi_0 \mathbf{H}_k^T + \bar{\mathbf{D}}_k \mathcal{Q}_k \mathbf{G}_k^T \\ &= [\mathbf{F} \mathcal{H}_{k-1}^a \quad \mathbf{M}_k \mathbf{H}^T]\end{aligned}\quad (4.51)$$

En la que  $\mathbf{M}_k = \mathbf{F}^k \chi_0 \mathbf{F}^{kT} + \bar{\mathbf{D}}_k \mathcal{Q}_k \bar{\mathbf{D}}_k$ . Definimos  $\mathbf{Z}_{\chi k} = \mathbf{H}_k \chi_0 \mathbf{H}_k^T$  y  $\mathbf{Z}_{2k} = \mathbf{G}_k \mathcal{Q}_k \mathbf{G}_k^T$ , la transformación  $\mathbf{Z}_{\chi k} + \mathbf{Z}_{2k}$  es

$$\mathbf{Z}_{\chi k} + \mathbf{Z}_{2k} = \begin{bmatrix} \mathcal{H}_{k-1}^{a^{-1}} - \mathcal{R}_{k-1} & \tilde{\mathcal{H}}_{k-1}^T \\ \tilde{\mathcal{H}}_{k-1} & \mathbf{H} \mathbf{M}_k \mathbf{H}^T \end{bmatrix}$$

Donde  $\tilde{\mathcal{H}}_{k-1} = \mathbf{H} \mathbf{F} \mathcal{H}_{k-1}^a$ . Para  $\mathcal{R}_k = \text{diag}(\mathcal{R}_{k-1}, \mathbf{R})$  la forma recursiva de  $\mathbf{Z}_{\chi k} + \mathbf{Z}_{2k} + \mathcal{R}_k$  resulta en

$$\mathbf{Z}_{\chi k} + \mathbf{Z}_{2k} + \mathcal{R}_k = \begin{bmatrix} \mathcal{H}_{k-1}^{b^{-1}} & \tilde{\mathcal{H}}_{k-1}^T \\ \tilde{\mathcal{H}}_{k-1} & \mathbf{H} \mathbf{M}_k \mathbf{H}^T + \mathbf{R} \end{bmatrix}$$

Separando en dos matrices de bloques  $\bar{\mathbf{Z}}_k = \text{diag}(\mathcal{H}_{k-1}^{b^{-1}}, \mathbf{R})$ , además

$$\tilde{\mathbf{Z}}_k = \begin{bmatrix} 0 & \tilde{\mathcal{H}}_{k-1}^T \\ \tilde{\mathcal{H}}_{k-1} & \mathbf{H} \mathbf{M}_k \mathbf{H}^T \end{bmatrix}$$

y descomponiendo  $\mathcal{H}_k^b$  como

$$\mathcal{H}_k^b = \bar{\mathbf{Z}}_k^{-1} \hat{\mathbf{Z}}_k^{-1} \quad (4.52)$$

En la que  $\hat{\mathbf{Z}}_k = \mathbf{I} + \tilde{\mathbf{Z}}_k \bar{\mathbf{Z}}_k^{-1}$ . Empleando el complemento de Schur, representamos  $\hat{\mathbf{Z}}_k^{-1}$  con

$$\hat{\mathbf{Z}}_{m,k}^{-1} = \begin{bmatrix} \mathcal{Z}_{11} & \mathcal{Z}_{12} \\ \mathcal{Z}_{21} & \mathcal{Z}_{22} \end{bmatrix} \quad (4.53)$$

donde  $\mathcal{Z}_{11} = \mathbf{I} + \bar{\Omega}_k$ ,  $\bar{\Omega}_k = \tilde{\mathcal{H}}_{k-1}^T \mathbf{R}^{-1} \Omega_k^{-1} \tilde{\mathcal{H}}_{k-1} \mathcal{H}_{k-1}^b$ ,  $\mathcal{Z}_{12} = \tilde{\mathcal{H}}_{k-1}^T \mathbf{R}^{-1} \Omega_k^{-1}$ ,  $\mathcal{Z}_{22} = \Omega_k^{-1}$  y

$$\Omega_k = \mathbf{I} + \mathbf{H} \mathbf{P}_k^- \mathbf{H}^T \mathbf{R}^{-1} \quad (4.54)$$

$$\mathbf{P}_k^- = \mathbf{M}_k - \mathbf{F} \mathcal{H}_{k-1} \mathcal{H}_{k-1}^{aT} \mathbf{F}^T. \quad (4.55)$$

Combinando (4.51) con (4.52) se llega a la recursión

$$\mathcal{H}_k = [(\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{F} \mathcal{H}_{k-1} \quad \mathbf{K}_k] \quad (4.56)$$

Donde

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (4.57)$$

y  $\mathbf{P}_k^-$  está dado por (4.55). Se incluye la innovación de la covarianza (4.44) para llegar de (4.57) a la ganancia de Kalman (4.46). Sustituyendo (4.56) en  $\hat{x}_k^h = \mathcal{H}_k \mathbf{Y}_k$  se llega a la recursión

$$\hat{x}_k^h = \mathbf{F}\hat{x}_{k-1}^h + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\mathbf{F}\hat{x}_{k-1}^h). \quad (4.58)$$

Considerando que  $\hat{x}_k^f = (\bar{\mathbf{S}}_k - \mathcal{H}_k \mathbf{L}_k) \mathbf{U}_k$  donde  $\bar{\mathbf{S}}_k = [\mathbf{F}\bar{\mathbf{S}}_{k-1} \quad \mathbf{E}]$  y

$$\mathbf{L}_k = \begin{bmatrix} \mathbf{L}_{k-1} & 0 \\ \mathbf{H}\mathbf{F}\bar{\mathbf{S}}_{k-1} & \mathbf{H}\mathbf{E} \end{bmatrix}$$

Usando las expresiones anteriores la respuesta forzada puede ser estimada de forma recursiva como

$$\hat{x}_k^f = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{F} \hat{x}_{k-1}^f + (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{E} u_k. \quad (4.59)$$

Combinando (4.56) con (4.59) se llega al estimado de Kalman (4.47) en el que el estimado previo  $\hat{x}_k^-$  está dado por (4.45). Para encontrar una recursión para  $\mathbf{P}_k^-$  se transforma (4.55) en una ecuación dinámica de Ricatti en tiempo discreto

$$\begin{aligned} \mathbf{P}_k^- &= \mathbf{F}\mathbf{P}_{k-1}^- \mathbf{F}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T - \mathbf{F}\mathbf{P}_{k-1}^- \mathbf{H}^T \\ &\quad \times (\mathbf{H}\mathbf{P}_{k-1}^- \mathbf{H}^T + \mathbf{R}_{k-1})^{-1} \mathbf{H}\mathbf{P}_{k-1}^- \mathbf{F}^T \end{aligned} \quad (4.60)$$

Que puede ser representada como

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}^- \mathbf{F}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T - \mathbf{F}\mathbf{K}_{k-1} \mathbf{H}\mathbf{P}_{k-1}^- \mathbf{F}^T$$

Esta puede ser fácilmente llevada a (4.43) y (4.48) completando la prueba.  $\square$

Un pseudocódigo del estimador  $H_2$ -OFIR *a posteriori* operando en un horizonte  $[m, k]$  con matrices diagonales  $\mathcal{Q}_N$  y  $\mathcal{R}_N$  se muestra en el algoritmo 1.

Dados los valores de  $\hat{x}_m$  y  $\mathbf{P}_m$ , el algoritmo actualiza de forma iterativa los estimados desde  $i = m + 1$  a  $i = k$  para producir  $\hat{x}_k$  y  $\mathbf{P}_k$  cuando  $i = k$ . El algoritmo anterior explota las recursiones de Kalman para un ruido  $\mathbf{Q}$  y  $\mathbf{R}$  no necesariamente gaussianos. Todas las variables aleatorias en (4.1) se asumen de media cero, limitadas y sin correlación mutua.

## 4.5. Filtro subóptimo $H_2$ -FIR usando LMI

Los algoritmos de filtrado  $H_2$ -FIR numéricamente subóptimos son necesarios para operar en estructuras híbridas de seguimiento robusto [55, 61]. Un ejemplo es el filtro  $H_2/H_\infty$ -FIR, cuya ganancia es determinada resolviendo el problema LMI sujeto

---

**Algoritmo 1:** Algoritmo Iterativo  $H_2$ -OFIR a posteriori para matrices de error diagonales

---

**Datos:**  $\hat{x}_m, y_k, u_k, \mathbf{P}_m, \mathbf{Q}, \mathbf{R}, \mathbf{N}$

**Resultado:**  $\hat{x}_k$

**inicio**

```

para  $k = 1, 2, \dots$  hacer
    si  $k > N - 1$ 
         $m = k - N + 1$ 
    de otra forma
         $m = 0$ 
    para  $i = m + 1, m + 2, \dots k$  hacer
         $\hat{x}_i^- = \mathbf{F}\hat{x}_{i-1} + \mathbf{E}u_i$ 
         $\mathbf{P}_i^- = \mathbf{F}\mathbf{P}_{i-1}\mathbf{F}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T$ 
         $\mathbf{S}_i = \mathbf{H}\mathbf{P}_i^-\mathbf{H}^T + \mathbf{R}$ 
         $\mathbf{K}_i = \mathbf{P}_i^-\mathbf{H}^T\mathbf{S}_i^{-1}$ 
         $\hat{x}_i = \hat{x}_i^- + \mathbf{K}_i(y_i - \mathbf{H}\hat{x}_i^-)$ 
         $\mathbf{P}_i = (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i)\mathbf{P}_i^-$ 
    
```

---

a restricciones de  $H_2$  y  $H_\infty$  [37]. Para obtener una ganancia subóptima para el filtro  $H_2$ -FIR a partir de la matriz de error (4.42) se introduce una matriz auxiliar  $\mathcal{Z}$  tal que

$$\mathcal{Z} > \mathcal{B}_N \chi_m \mathcal{B}_N^T + \mathcal{W}_N \mathcal{Q}_N \mathcal{W}_N^T + \mathcal{V}_N \mathcal{R}_N \mathcal{V}_N^T.$$

Sustituyendo las matrices de error (4.8) podemos reescribir la desigualdad como

$$\begin{aligned} \mathcal{Z} - (\mathbf{F}^{N-1} - \mathcal{H}_N \mathbf{H}_N) \chi_m (\mathbf{F}^{N-1} - \mathcal{H}_N \mathbf{H}_N)^T - (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N) \mathcal{Q}_N (\bar{\mathbf{D}}_N - \mathcal{H}_N \mathbf{T}_N)^T \\ - \mathcal{H}_N \mathcal{R}_N \mathcal{H}_N^T > 0, \\ \mathcal{Z} - \mathbf{F}^{N-1} \chi_m \mathbf{F}^{N-1T} + \mathbf{F}^{N-1} \chi_m \mathbf{H}_N^T \mathcal{H}_N^T + \mathcal{H}_N \mathbf{H}_N \chi_m \mathbf{F}^{N-1T} - \mathcal{H}_N \mathbf{H}_N \chi_m \mathbf{H}_N^T \mathcal{H}_N^T \\ - \bar{\mathbf{D}}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T + \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T \mathcal{H}_N^T + \mathcal{H}_N \mathbf{G}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T - \mathcal{H}_N \mathbf{G}_N \mathcal{Q}_N \mathbf{G}_N^T \mathcal{H}_N^T \\ - \mathcal{H}_N \mathcal{R}_N \mathcal{H}_N^T > 0, \end{aligned}$$

Agrupando términos para lograr una forma cuadrática se obtiene

$$\mathcal{Z} - \mathcal{A} + \mathcal{C}^T \mathcal{H}_N^T + \mathcal{H}_N \mathcal{C} - \mathcal{H}_N (\mathbf{H}_N \chi_m \mathbf{H}_N^T + \Omega) \mathcal{H}_N^T > 0 \quad (4.61)$$

Donde se introducen las siguientes matrices en bloques

$$\begin{aligned} \mathcal{A} &= \mathbf{F}^{N-1} \chi_m \mathbf{F}^{N-1T} + \bar{\mathbf{D}}_N \mathcal{Q}_N \bar{\mathbf{D}}_N^T \\ \mathcal{C} &= \mathbf{F}^{N-1} \chi_m \mathbf{H}_N^T + \bar{\mathbf{D}}_N \mathcal{Q}_N \mathbf{G}_N^T \\ \Omega &= \mathbf{G}_N \mathcal{Q}_N \mathbf{G}_N^T + \mathcal{R}_N. \end{aligned}$$

Empleando el complemento de Schur podemos llevar la desigualdad (4.61) a la forma de un LMI

$$\begin{bmatrix} \mathcal{Z} - \mathcal{A} + \mathcal{C}^T \mathcal{H}_N^T + \mathcal{H}_N \mathcal{C} & \mathcal{H}_N \\ \mathcal{H}_N^T & (\mathbf{H}_N \chi_m \mathbf{H}_N^T + \Omega)^{-1} \end{bmatrix} > 0. \quad (4.62)$$

La ganancia  $\mathcal{H}_N$  para el filtro subóptimo  $H_2$ -FIR puede ser calculado resolviendo el problema de minimización

$$\mathcal{H}_N = \underset{\mathcal{H}_N, \mathcal{Z}}{\text{mín}} \text{tr} \mathcal{Z} \quad (4.63)$$

subject to (4.62)

Se inicia la minimización con la ganancia del filtro UFIR  $\mathcal{H}_N = (\mathbf{C}_N^T \mathbf{C}_N)^{-1} \mathbf{C}_N^T$  [42]. Una vez obtenida la ganancia empleando (4.63) el estimado robusto  $H_2$ -FIR puede ser calculado con (4.41) y la matriz de error con (4.42).

# Capítulo 5

## Implementación de un Robot Omnidireccional de Tres Ruedas

Las aplicaciones industriales y técnicas de los robots móviles están adquiriendo cada vez más relevancia. Los robots móviles se emplean extensamente en tareas de supervisión, logística, inspección y transporte. El requisito fundamental para que los robots móviles sean autónomos radica en dos principales tareas, localización y navegación. Que implican la capacidad de un robot móvil para conocer su posición dentro del espacio de trabajo, además el robot debe ser capaz de encontrar su destino incluso en presencia de factores externos como obstáculos u otros agentes móviles. A partir de su posición dentro del espacio de trabajo, los robots puede navegar hacia su siguiente ubicación utilizando diversos sensores, referencias externas y algoritmos.

El desempeño de un robot móvil se ve significativamente influido por los algoritmos de localización y navegación, así como por sus capacidades de detección. La investigación en este campo sigue avanzando constantemente con el objetivo de mejorar la capacidad de navegación autónoma de los sistemas robóticos móviles. Para evaluar las estrategias recientemente desarrolladas en detección y localización, se presenta una plataforma robótica móvil omnidireccional, que proporciona una base flexible para poner a prueba y mejorar los algoritmos que se desarrollaran a lo largo de esta tesis.

En primer lugar, se proporcionará una breve explicación de la arquitectura omnidireccional y las especificaciones necesarias para la plataforma móvil. Se presentara la selección de la unidad de procesamiento y componentes electrónicos y electromecánicos asociados al dispositivo. Seguiremos con la presentación del software y finalmente el prototipo terminado.

## 5.1. Diseño y planificación

Se busca construir una plataforma de desarrollo para aplicaciones robóticas que tenga las siguientes características:

- Alta movilidad
- Capaz de soportar al menos 15kg de carga
- Con arquitectura de control centralizada
- Que pueda integrarse fácilmente con otros robots y softwares
- Escalable en hardware
- Diversas interfaces de comunicaciones
- Integrable con diversos OS

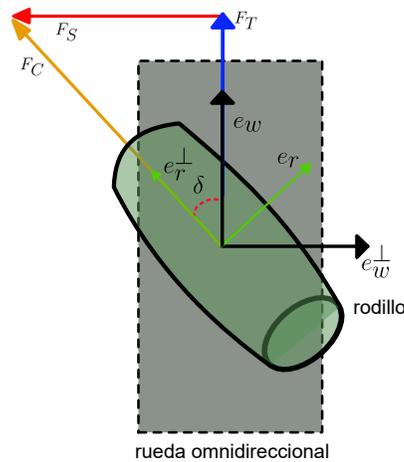
Con respecto a la movilidad y a la capacidad de carga, las mejor opción se reduce al uso de un robot omnidireccional. Una arquitectura omnidireccional puede presentarse bajo diversas configuraciones, pero en esencia todas ellas poseen la capacidad de desplazarse de un punto a otro sin modificar su orientación. Es decir pueden moverse de forma instantánea en cualquier dirección a partir de cualquier configuración. En el plano, un robot móvil tendrá tres grados de libertad. El uso de ruedas convencionales en un robot móvil causa que éste sea incapaz de moverse de forma perpendicular a la dirección de la velocidad de sus ruedas, esta restricción es llamada no-holonómica. La movilidad omnidireccional se logra mediante la incorporación de ruedas omnidireccionales en la base del robot. Los tipos de ruedas omnidireccionales mas comunes se pueden ver en la figura 5.1.



**Figura 5.1.** Ruedas omnidireccionales. a) De rodillos ortogonales, b) De rodillos angulares (Mecanum)

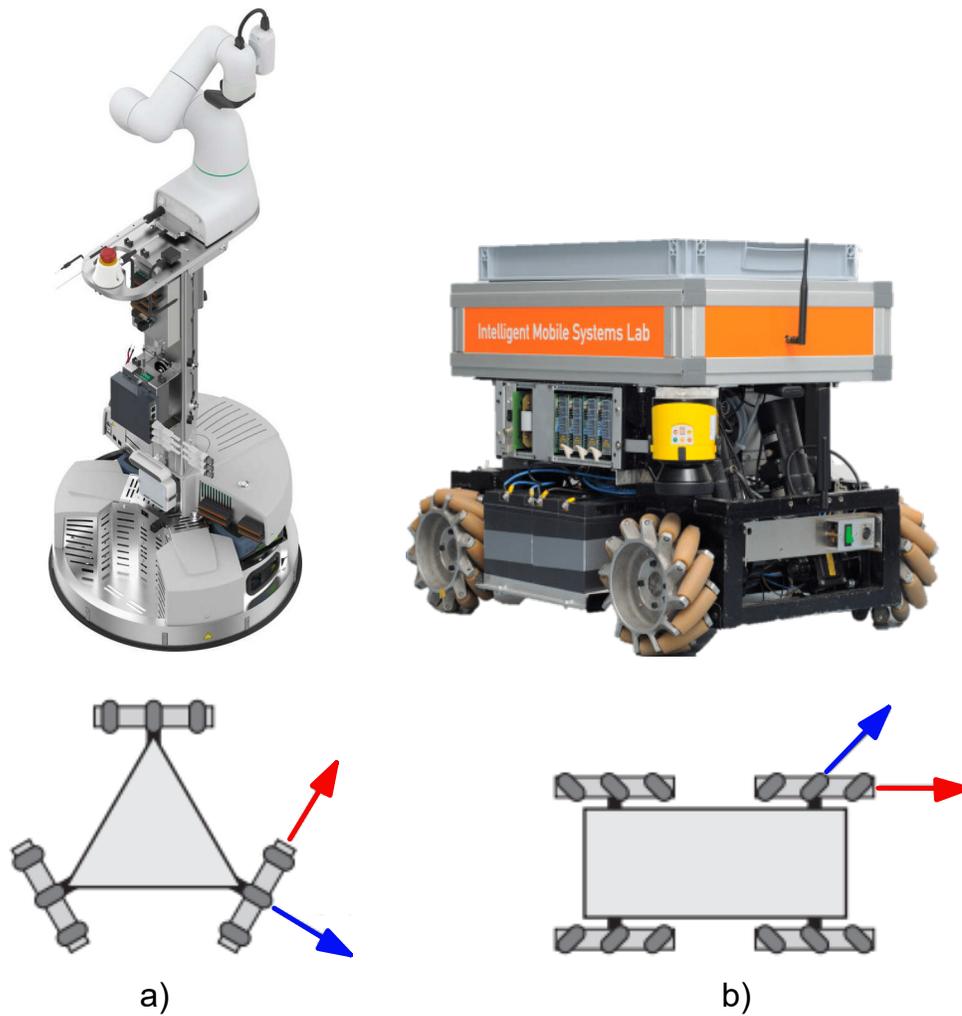
En la Fig 5.2 se observa un diagrama de la rueda omnidireccional, ésta se compone de rodillos dispuestos de manera que permiten la transmisión de una porción

de la fuerza en la dirección de rotación ( $F_T$ ) a la fuerza normal en la dirección de la rueda ( $F_S$ ), con esto se logra que el robot se desplace en todas las direcciones. La rueda tiene dos ejes, el principal  $e_w^\perp$  que se une a la flecha del motor, y una serie de ejes pasivos  $e_r^\perp$  que permiten el giro del rodillo cuando esta en contacto por la superficie. Aunque estas ruedas tienen una gran capacidad de carga, un factor de forma relativamente compacto y mejoran la maniobrabilidad del robot son altamente susceptibles al deslizamiento y el punto de contacto con el terreno es discontinuo. Un robot omnidireccional debe



**Figura 5.2.** Ejes de los rodamientos y de la rueda omnidireccional

tener al menos tres ruedas para alcanzar la tridimensionalidad en el chasis de velocidad  $\dot{q} = (\dot{x}, \dot{y}, \dot{\phi})$ , debido a que cada rueda es una variable de actuación de esta forma se satisfacen las restricciones de holonomía del sistema. Las dos configuraciones mas comunes para el chasis de un robot holonómico se muestran en la figura 5.3. Si bien existen algunas opciones open hardware para montar un robot móvil [74,75] se ha optado por usar el kit Pitsco Tetrax Max Building System, que incluye diversos elementos estructurales, motores Torquenado, servomotores, baterías, cargador, controladoras para motores PRIZM y mas de 670 elementos de construcción. Usando estos elementos y algunas piezas personalizadas impresas en PLA se construyó la plataforma para este robot. El uso de este kit nos da la posibilidad de escalar el robot para futuras mejoras sin cambios mayores. En cuanto a la configuración, se optará por el diseño de un robot omnidireccional de tres ruedas.



**Figura 5.3.** Plataformas Omnidireccionales, a) De tres ruedas, b) De cuatro ruedas. En rojo la dirección activa de la rueda, en azul la pasiva.

## 5.2. Componentes de Hardware

Dentro de las características requeridas en las especificaciones de diseño para robot destacan las relacionadas a su capacidad de comunicación. A fin de construir una plataforma de fácil integración con diversos softwares y sistemas operativos se ha optado por usar una Nvidia Jetson TX2, que cuenta con diversas interfaces tales como I2C, LAN, USB, Serial TTL, Serial CMOS. Adicionalmente la Jetson cuenta con una cámara embebida. El kit de Pitsco incluye controladoras y motores por lo que no es necesario diseñar una etapa de potencia, sin embargo, la tarjeta de control PRIZM solo se comunica a través de un cable NXT compatible con LEGO® MINDSTORMS®. Por ello, es necesario implementar en un microcontrolador un nuevo servicio, que sirva como puente entre los comandos del PRIZM. Se añadirán algunos sensores adicionales tales como una cámara T265, sensores Lidar, sensor UWB y lectora de tags RFID.

La lista de componentes empleados en la construcción del robot está en la tabla 5.1. Un diagrama esquemático de la interconexión de los componentes con la unidad de procesamiento se puede ver en la figura 5.4.

Componente	Descripción	Cantidad
JetsonTX2	Dispositivo de cómputo	1
TETRIX® Motor Controller	Controladora de 2 motores de DC	2
Cámara T265	Cámara Intel Realsense	1
DW1000	Sensor UWB	1
VL53L0X	Sensor de distancia láser	3
Batería Tetrax	Batería para motores 12v 3000 mAh	2
Batería 12V	Bateria 12V7Ah	1
Interfaz NXT-Serial	Controladora personalizada	1
Motor Torqueno	Motor 12V, 100RPM, 700 oz-in	3
Rueda Omnidireccional		3
Tetrax MAX 288mm		5
Tetrax MAX 160mm		4
Tetrax MAX 96mm		6
Tetrax MAXFlat Buiding Plate		2
Tetrax MAX Motor Mount		6
ALR-F800	Lector de Tags RFID	1
ALR-8697	Antena de polarizacion circular	2
Tornilleria 3mm		52
Tornilleria 5/16x2		16
Soporte Baterias		2
Soporte Hexagonal		6

**Tabla 5.1.** Lista de materiales

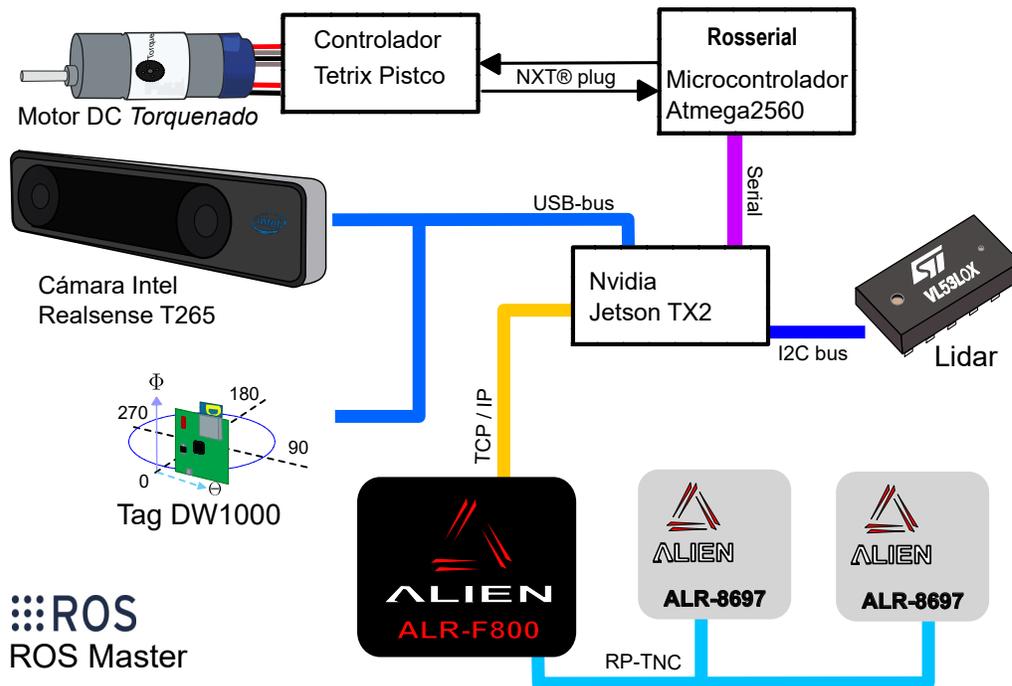


Figura 5.4

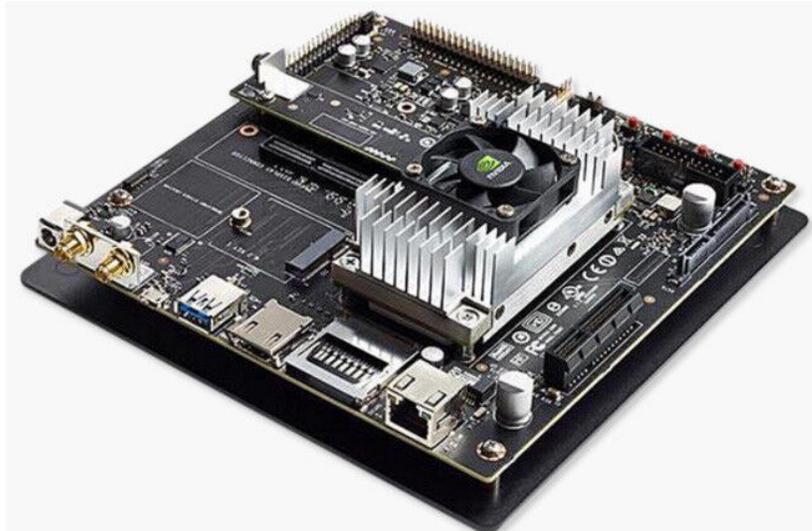
### 5.2.1. NVIDIA Jetson TX2

Las computadoras de la familia Jetson han adquirido mucha popularidad en aplicaciones móviles debido a su bajo consumo energético y al entorno NVIDIA JetPack SDK que incluye librerías optimizadas para tareas típicas de la inteligencia artificial, tales como deep learning, vision, etc. la NVIDIA Jetson TX2 es un dispositivo informático de que consume alrededor de 7,5 watts. Está construido alrededor de una GPU de la familia NVIDIA Pascal™ y cargado con 8 GB de memoria y 59,7 GB/s de ancho de banda de memoria. Cuenta con una variedad de interfaces de hardware estándar que facilitan su integración en una amplia gama de productos y factores de forma. Entre sus principales características se tienen

- GPU: GPU Pascal™ de 256 núcleos
- CPU: CPU Denver 2 de 64 bits de doble núcleo + MPCore ARM® Cortex®-A57 de cuatro núcleos
- Memoria: LPDDR4 de 8 GB de 128 bits
- Almacenamiento: 32 GB eMMC 5.1
- Codificación: 4K x 2K 60 Hz (HEVC)

- Decodificación: 4K x 2K 60 Hz (compatible con 12 bits)
- Ethernet: negociación automática 10/100/1000BASE-T
- Inalámbrico: WIFI 802.11ac + Bluetooth
- Cámara: 12 carriles MIPI CSI-2, D-PHY 1.2 (30 Gbps)
- Pantalla: HDMI 2.0 / eDP 1.4 / 2x DSI / 2x DP 1.2
- USB: USB 3.0 + USB 2.0 (MicroUSB)
- Otros: GPIO, I2C, I2S, SPI, UART
- Módulo central: 87 mm × 50 mm
- Placa base: 170,2 mm x 170,2 mm x 15,6 mm

Esta tarjeta alberga tanto la unidad de procesamiento de gráficos (GPU) como unidad central de procesamiento (CPU) en el mismo chip. La GPU integrada permite se compartan datos dinámicos en memoria de acceso aleatorio (DRAM) con la CPU. Esto permite que tanto la GPU como la CPU funcionen más eficientemente con baja potencia, entre 5 vatios con máxima eficiencia y 15 vatios con máximo rendimiento. Esta colaboración permite un procesamiento más rápido y de alta eficiencia puesto que requiere un enfriamiento mínimo y proporciona espacio adicional.



**Figura 5.5.** Tarjeta de desarrollo NVIDIA Jetson TX2

### 5.2.2. RealSense Camera



**Figura 5.6.** Intel Realsense T265

La cámara Intel® RealSense™ T265 incluye dos sensores de lente ojo de pez, una IMU y una VPU Intel® Movidius™ Myriad™ 2. Ha sido diseñada para ejecutar tareas de seguimiento de forma independiente y autónoma de la plataforma, siendo así amigables para desarrolladores y facilitando el prototipos rápido en aplicaciones de robótica, drones, realidad aumentada, visión, etc. Los algoritmos VSLAM se ejecutan directamente en la VPU, lo que permite una latencia muy baja y un consumo de energía extremadamente eficiente(1.5W) y con una desviación mejor al 1 % en lazo cerrado. También ofrece una latencia inferior a 6 ms entre el movimiento y el reflejo del movimiento en la pose Es de tamaño pequeño 108 x 25 x 13 mm y pesa 55 g. Incorpora una unidad de medición inercial (IMU) BMI055 que permite una medición precisa de la rotación y la aceleración del dispositivo para alimentar los algoritmos V-SLAM. Con el T265, los desarrolladores ahora pueden disponer de un seguimiento preciso y sólido que ha sido probado exhaustivamente en una variedad de condiciones y entornos. Este sistema de seguimiento autónomo está diseñado para una integración sencilla a través de un USB3.0.

### 5.2.3. TETRIX® MAX DC Motor Expansion Controller

Este controlador de motor se conecta al puerto de expansión PRIZM®, lo que permite controlar hasta dos motores CC de 12 voltios adicionales. Se pueden conectar hasta cuatro controladores de motor al puerto de expansión PRIZM. Tiene dos salidas de puente H para controlar la velocidad y dirección de dos motores de CC. Incluye dos puertos de entrada de codificador de cuadratura. Los puertos de alimentación y expansión adicionales admiten configuraciones de conexión en cadena. Los lenguajes de programación compatibles incluyen Arduino y LEGO® MINDSTORMS® EV3.

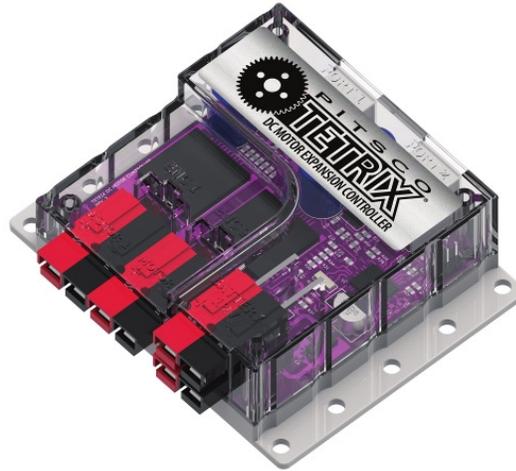


Figura 5.7

#### 5.2.4. Motores DC Torquenado

El motor Torquenado opera a 12VDC con 100 rpm y 700 oz-in. de par. El motor cuenta con un codificador de efecto Hall de alta resolución incorporado para un control preciso del motor y viene con un cable codificador. Una reducción mecánica de 60:1 viene de serie en el motor TorqueNADO. Sus características generales son:

- Peso: 324 gramos
- Voltaje de funcionamiento: 12 V
- Longitud: 134,1 mm/5,28 pulgadas.
- Relaciones de transmisión: 60:1
- Diámetro: 37mm
- Eje: eje D de 6 mm
- Par de parada: 700 oz/pulg.
- Velocidad sin carga (RPM): 100
- Motor: DC de escobillas
- Voltaje del codificador: 3,3/5 VCC
- Tipo de codificador: efecto Hall/cuadratura
- Conteos de salida por Revolución (cpr): 1.440 pulsos



Figura 5.8. Motor DC Torquenado

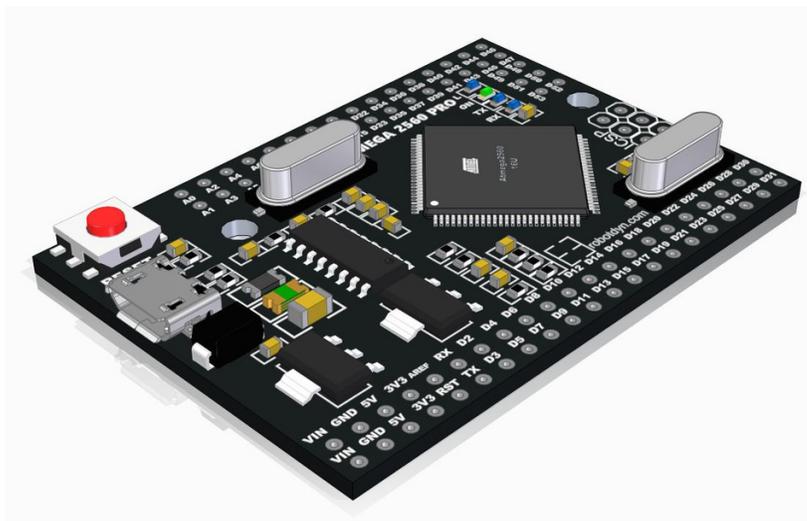
### 5.2.5. Interfaz NXT-Serial

Como se ha mencionado, esta tarjeta tendrá como finalidad lograr la interconexión del controlador TETRIX MAX y la Jetson TX2. El hardware de Pitsco originalmente debería de conectarse a una controladora PRIZM o a un Brick EVO de LEGO. Investigando las características del controlador PRIZM del fabricante encontramos lo siguiente:

- Procesador ATmega328P con gestor de arranque Arduino Optiboot instalado
- 4 puertos de sensores digitales (D2 se puede utilizar como puerto serie) y 3 puertos de sensores analógicos
- 1 puerto I2C, 1 puerto de programación USB, 1 puerto de expansión del controlador del motor y 2 puertos de entrada de codificador de cuadratura
- 2 puertos de control de motor CC de alta corriente
- 6 puertos servo de control estándar y 2 puertos servo de rotación continua (CR)
- Puerto de conexión de batería (puerto adicional utilizado para conectar en cadena la alimentación a otros dispositivos)
- Botón de inicio programable (verde) y un botón de parada/reinicio no programable (rojo)
- Memoria programable flash de 32 KB

del fabricante ofrece una librería desarrollada en Arduino, software desde el cual se pueden hacer aplicaciones y cargarlas en la controladora PRIZM. Hasta este punto podemos decir que la controladora funciona con un microcontrolador de la familia

ATMega y es compatible con el IDE de Arduino. Para crear el puente entre la controladora de los motores y la Jetson se eligió usar un ATmega2560, en su versión con breakoutboard conocida como megapro. Adicionalmente se creó un circuito impreso para conectar los dispositivos asociados a éste.



**Figura 5.9.** *Arduino Mega2560 en la breakoutboard MEGAPRO*

### 5.2.6. DWM1000

La tecnología UWB es una tecnología de comunicación de operador inalámbrico que utiliza un ancho de banda de frecuencia superior a 1 GHz. No utiliza una portadora sinusoidal, pero utiliza pulsos estrechos no sinusoidales de nivel de nanosegundos para transmitir datos. Por tanto, ocupa un amplio espectro. Aunque se utiliza comunicación inalámbrica, su velocidad de transmisión de datos puede alcanzar cientos de megabits por segundo. El uso de la tecnología UWB puede transmitir señales en un ancho de banda muy amplio. Para este robot se escogió el transceptor BU01 es un módulo que opera en base al nodeMCU y el circuito DW1000 de Decawave. Éste ya tiene integrada la antena, el circuito RF, la administración de energía y el circuito de reloj. Se puede utilizar en sistemas de posicionamiento TDOA o de alcance bidireccional, con una precisión de posicionamiento de hasta 10 cm y soporta velocidades de datos de hasta 6,8 Mbps. Este módulo se opera mediante comandos AT. Entre algunas de sus características tenemos:

- Integración sencilla, no se requiere diseño de RF
- Uso de la infraestructura RTLS para ampliar el alcance de la comunicación
- Admite alta densidad de etiquetas

- Cumple con el estándar IEEE 802.15.4-2011 UWB
- Admite 4 canales de 3,5 GHz a 6,5 GHz
- Potencia de transmisión programable
- Fuente de alimentación de 2,8 V a 3,6 V
- Consumo de energía en modo de suspensión <1 mA
- Admite alcance bidireccional y TDOA
- Admite interfaz SPI
- Velocidad de datos 110 kbps, 850 kbps, 6,8 Mbps

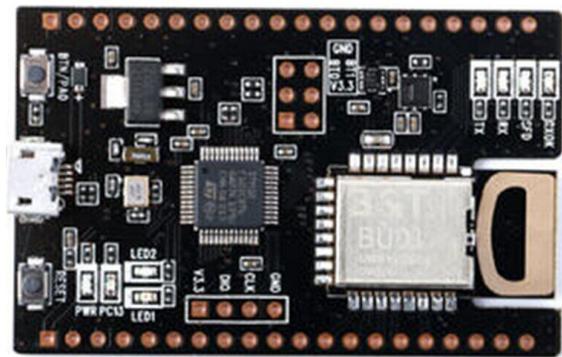
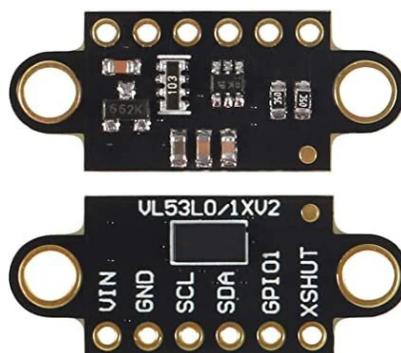


Figura 5.10. Módulo BU01-DB de AiThinker

### 5.2.7. Lidar

El VL53L0X es un sensor de distancia infrarrojo láser que podemos operar mediante una interfaz I2C para medir distancias de 50mm a 2000 mm de forma precisa. El VL53L0X es capaz de operar incluso con elevada luz ambiental infrarroja, e incorpora un sistema de compensación de la medición que lo permite hacer funcionar incluso detrás de un cristal protector. Es uno de los mejores sensores de distancia disponibles. Tiene una precisión superior que los sensores de ultrasonidos e infrarrojos, y no se ve alterado por las condiciones del ambiente como los ecos o la reflectancia de los objetos. Por otro lado el ángulo de medición es relativamente estrecho. Esto es una ventaja en la mayoría de circunstancias, donde deseamos leer la distancia justo en frente del sensor. Aunque también puede ser una desventaja, por ejemplo, a la hora de detectar obstáculos.



**Figura 5.11.** Sensor de distancia láser VL53L0X

El VL53L0X es un sensor ToF (Time of flight). Su funcionamiento consiste en enviar un pulso láser de luz infrarroja y medir el tiempo necesario en el haz en volver al sensor. El integrado incorpora un emisor láser 940nm VCSEL (Vertical Cavity Surface-Emitting Laser), un detector SPAD (Single Photon Avalanche Diodes) y la electrónica interna (denominada FlightSense™) que realiza los cálculos necesarios. El ángulo de medición o FOV (Field of View) es de 25°. Esto se traduce en un área de medición de 0.44m de diámetro a una distancia de 1m. El rango de medición depende de las condiciones del entorno (interior, o exterior), de las características del objetivo y del modo de funcionamiento. En general, tenemos dos modos. El estándar es de 50 a 1200mm, y un modo ampliado hasta 2000mm. En la Jetson, estos dispositivos se conectarán directamente a las GPIO a través del conector J21 como muestra la figura 5.12.

Jetson TX2 J21 Header					
Sysfs GPIO	Nombre	Pin	Pin	Nombre	Sysfs GPIO
	3.3 VDC Alimentación	1	2	5.0 VDC Alimentación	
	SDA1 Bus 0 I2C General	3	4	5.0 VDC Alimentación	
	SCL1 Bus 0 I2C General	5	6	GND Alimentación	
GPIO26	GPIO_GCLK Reloj maestro de Audio	7	8	TDX0 UART #0 Transmisión	
	GND Alimentación	9	10	RDX0 UART #0 Recepción	

**Figura 5.12.** Conector J21 en la Jetson TX2

### 5.3. Programación y software

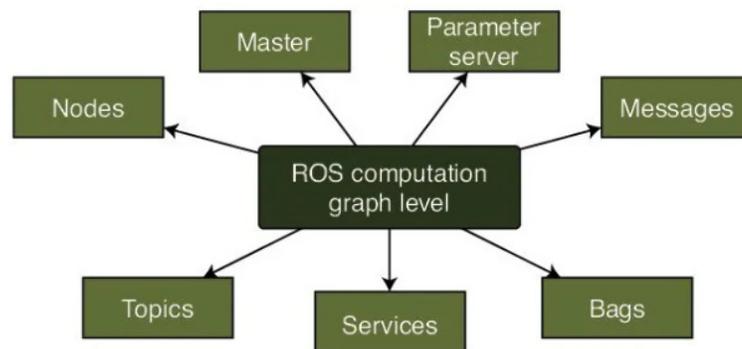
La implementación principal de el software diseñado para el control de este robot fue implementado en Robot Operative System (ROS). ROS es un manejador de paquetes de Linux diseñado para permitir la comunicación entre plataformas. ROS es un meta-sistema operativo de código abierto para robots. Este provee los mismos servicios que se pueden esperar de un sistema operativo, incluyendo abstracción de hardware, control de dispositivos de bajo nivel, implementación de funciones comúnmente empleadas, intercambio de mensajes entre procesos, y manejo de paquetes. Además, contiene herramientas y librerías para obtener, construir, escribir y ejecutar código a través de múltiples computadoras. ROS incluye una capa de abstracción de hardware similar a los sistemas operativos, sin embargo en ROS puede ser empleada para distintas implementaciones de hardware. Una descripción mas precisa es decir que ROS es un meta-sistema operativo, es decir, se ejecuta en un sistema operativo existente. Las características propias por el sistema operativo base tales como manejo del sistema, sistema de archivos, interfaces de usuario, compiladores, etc. pueden ser usadas por ROS. Además de heredar las capacidades propias el sistema base, ROS provee funciones esenciales requeridas por aplicaciones orientados a la robótica como librerías, transmisión o recepción de datos entre hardware heterogéneos, planificación y manejo de errores, utilizando capas de virtualización entre aplicaciones y distribuyendo recursos de cómputo. Este tipo de software es conocido como middleware, ROS desarrolla, maneja y provee paquetes de aplicaciones para distintos propósitos, y ha formado un ecosistema que distribuye paquetes desarrollados por sus usuarios. Como se observa en la figura, ROS es un sistema para controlar robots, sensores con una capa de abstracción de hardware y desarrollando aplicaciones de robótica basadas en sistemas operativos convencionales.

ROS es un software robusto que posee muchas características que permiten el desarrollo de robots. Tiene un sistema de comunicación a través de mensajes, herramientas para estimación, localización y navegación entre otras características que hacen de él un ecosistema que permite controlar de forma precisas proyectos de robótica. La primera etapa de esta sección consiste en instalar ROS melodic para Ubuntu 18 en la Jetson TX2, el procedimiento se encuentra descrito en [76].

#### 5.3.1. Computation Graph level

Las aplicaciones de ROS estan compuestas por procesos independientes pero acoplados que reciben el nombre de nodos, y son los encargados de realizar tareas. Los nodos se relacionan entre sí de forma similar a un grafo, donde cada nodo corresponde a un estado. ROS esta pensado para ser modular, un robot puede contener muchos nodos donde cada uno de ellos es encargado de realizar alguna tarea. En nuestro

caso, todos los nodos fueron escritos usando la librería de clientes de ROS *rospy*. Los nodos se comunican entre ellos intercambiando mensajes. Estos son una estructura de datos, ROS ya posee una serie de mensajes predefinidos, sin embargo es posible crear mensajes personalizados. Los mensajes se intercambian usando una semántica de *publish/subscribe*. Un nodo envía un mensaje a otro publicándolo en un tópico. El nodo que requiera información de ese tópico deberá suscribirse a él. Además se cuenta con *Bags*, que es un formato que permite guardar todos los mensajes que fueron enviados a través de ROS. En ROS existe un *maestro*, que es el encargado de registrar y observar todo el grafo. Sin el nodo maestro los demás nodos no podrían observarse entre sí, intercambiar mensajes o ejecutar servicios. Todos los nodos se reportan al maestro para registrar su información, además pueden recibir información del resto de los nodos. Los nodos se conectan entre sí de forma directa, sin embargo el maestro indica la ruta para hacerlo, de forma similar a un servidor DNS.



**Figura 5.13.** Representación de los niveles de cómputo del grafo

### 5.3.2. Tópicos

Para intercambiar información con el robot móvil se generaron diversos tópicos; Se abre una terminal y se ejecuta *roscore* para iniciar el nodo maestro. Estos pasos pueden ser ejecutados tras cada inicio del OS del Robot, una buena opción para realizarlo de forma automática es usar el Startup Applications menu.

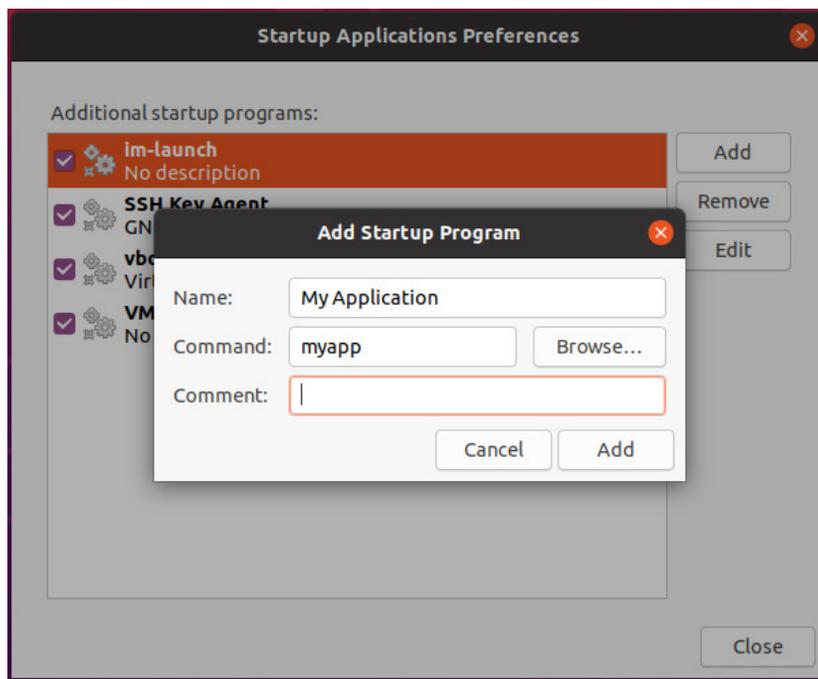


Figura 5.14. Ventana de aplicaciones de inicio

Una vez colocadas las tareas a iniciar estas se ejecutarán al arrancar la Jetson en el orden en el que aparecen en el menú. Por default, al iniciar el *roscore*, algunos nodos se ejecutarán, para poder observar los nodos que se encuentran activos se puede emplear el comando *rostopic list* en una nueva terminal. Algunos de los nodos que podremos ver serán los siguientes.

- /rosout
- /rosout\_agg
- /tf
- /tf\_static

## Cámara T265

En el caso de este dispositivo, el fabricante ya provee una wrapper para ROS, los pasos para su instalación se encuentran en [77]. A continuación en una nueva terminal se ejecutará `roslaunch realsense2_camera rs_t265.launch` para iniciar todos los sensores de la cámara y publicarlos en los tópicos apropiados. Este proceso también debe ser agregado al inicio. En la figura 5.15 se puede verificar la salida en la terminal al ejecutar el launchfile de la T265.

```

[*] /opt/ros/melodic/share/realsense2_camera/launch/rs_t265.launch http://192.168.101.5:11311 80x24
[ INFO] [1695077665.420623808]: Sync Mode: Off
[ INFO] [1695077665.420715106]: Device Sensors:
[ INFO] [1695077665.420980934]: Tracking Module was found.
[ INFO] [1695077665.421126633]: (Depth, 0) sensor isn't supported by current device! -- Skipping...
[ INFO] [1695077665.421198314]: (Color, 0) sensor isn't supported by current device! -- Skipping...
[ INFO] [1695077665.421265675]: (Confidence, 0) sensor isn't supported by current device! -- Skipping...
[ INFO] [1695077665.421358925]: num_filters: 0
[ INFO] [1695077665.421433230]: Setting Dynamic reconfig parameters.
[ WARN] [1695077665.445137632]: Param '/camera/tracking_module/frames_queue_size' has value 256 that is not in range [0, 32]. Removing this parameter from dynamic reconfigure options.
[ INFO] [1695077665.485396139]: Done Setting Dynamic reconfig parameters.
[ INFO] [1695077665.485633840]: gyro stream is enabled - fps: 200
[ INFO] [1695077665.485704049]: accel stream is enabled - fps: 62
[ INFO] [1695077665.485770482]: pose stream is enabled - fps: 200
[ INFO] [1695077665.485825715]: setupPublishers...
[ INFO] [1695077665.505524865]: setupStreams...
[ INFO] [1695077665.532793456]: SELECTED BASE:Pose, 0
[ INFO] [1695077665.542683352]: RealSense Node Is Up!

```

Figura 5.15. Salida en terminal del launchfile para la T265

Los tópicos empleados por la cámara son:

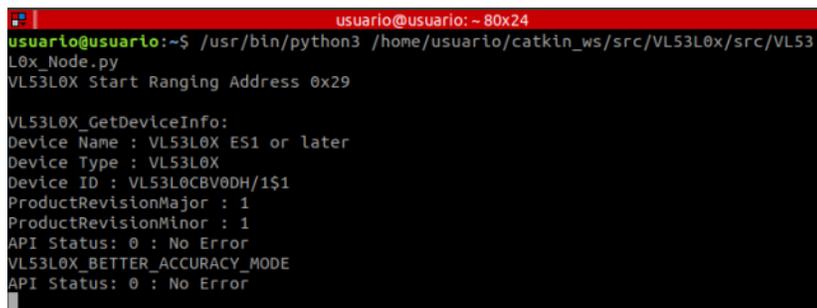
- /camera/accel/imu\_info
- /camera/accel/metadata
- /camera/accel/sample
- /camera/gyro/imu\_info
- /camera/gyro/metadata
- /camera/gyro/sample
- /camera/odom/metadata
- /camera/odom/sample
- /camera/realsense2\_camera\_manager/bond
- /camera/tracking\_module/parameter\_descriptions
- /camera/tracking\_module/parameter\_updates

## Lidar

Para poder comunicar el sensor VL53L0x con la Jetson es necesario instalar un componente de Python que permita la comunicación a través del puerto I2C. Esta requiera dos cables de señal, el de transferencia de datos(SDA), y una señal de reloj que indique la latencia con la que se intercambia la información(SCL). Existen dos puertos I2C (Bus0 y Bus1) en el conector J21 GPIO de la NVIDIA Jetson TX2. Para instalar la librería se ejecuta `sudo apt-get install libi2c-dev i2c-tools`, es posible verificar cuantos dispositivos estan asociados al bus I2C ejecutando `sudo i2cdetect -y -r 1`, donde el "1" al final es el número de bus I2C por probar. El nodo utilizará un mensaje estándar para llevar la lectura del sensor al respectivo tópico de ROS. También recibirá como argumento opcional la dirección del esclavo y la velocidad con la que se refrescarán los datos. Cada mensaje tendrá una estructura de datos como la siguiente:

```
1   range_msg.radiation_type=range_msg.INFRARED
2   range_msg.field_of_view=25
3   range_msg.min_range=50
4   range_msg.max_range=1200
5   rate = rospy.Rate(rate_)
6   range_msg.header.frame_id=str(address_)
7   % actualizacion de las mediciones
8   range_msg.header.stamp=rospy.get_rostime()
9   range_msg.range=distance
```

Para poder identificar las mediciones que provienen de distinto sensor en la cabecera (frame) del mensaje publicado se usa como identificador (frame\_id) la dirección (address) del VL53L0x. El tópico donde se encontrarán estas mediciones tiene el nombre de "/Lidar".



```
usuario@usuario: ~ 80x24
usuario@usuario:~$ /usr/bin/python3 /home/usuario/catkin_ws/src/VL53L0x/src/VL53
L0x_Node.py
VL53L0X Start Ranging Address 0x29

VL53L0X_GetDeviceInfo:
Device Name : VL53L0X ES1 or later
Device Type : VL53L0X
Device ID : VL53L0CBV0DH/1$1
ProductRevisionMajor : 1
ProductRevisionMinor : 1
API Status: 0 : No Error
VL53L0X_BETTER_ACCURACY_MODE
API Status: 0 : No Error
```

Figura 5.16. Salida en terminal del nodo para el sensor VL53L0x

## Interfaz NXT-Serial

Este dispositivo se encargará de llevar los valores publicados en el tópico de ROS *drive* a la controladora TETRIX MAX y también leer los valores de los sensores acoplados a ella y publicarlos en los tópicos *current\_m*, *battery\_m* y *encoder\_m*. Leyendo el código que proporciona el fabricante se pudo saber que la interfaz empleada por la controladora es I2C, los cables que requiere la controladora fueron adaptados para poder interconectarlos con un microcontrolador más potente que el empleado por la controladora PRIZM del fabricante, en esta aplicación usamos el Atmega2560 y para comunicarse con la Jetson se emplea la librería *rosserial*, el microcontrolador es el encargado de gestionar todos los nodos y empleando *rosserial* los mensajes de los pueden ser transportados a la capa TCP de ROS. Para publicar mensajes de prueba para la controladora se puede emplear *rostopic pub -1 /drive\_cmd geometry\_msgs/Vector3 -0.0 0.0 0.0* donde los números corresponde a la velocidad de cada rueda.

```

1      % Librerias
2      std_msgs::UInt32 bt_msg;
3      geometry_msgs::Vector3Stamped currents;
4      geometry_msgs::Vector3Stamped encoders;
5      ros::NodeHandle nh;
6
7      void MotorSpeeds( const geometry_msgs::Vector3& vel){
8          % procesos . . .
9      }
10     % Topicos
11     ros::Subscriber<geometry_msgs::Vector3>
12     drive_sub("drive_cmd",&MotorSpeeds);
13
14     ros::Publisher im_pub("current_m",&currents);
15     ros::Publisher de_pub("encoder_m",&encoders);
16     ros::Publisher bt_pub("battery_m",&bt_msg);
17
18     void setup(){
19         % Inicializar
20     }
21
22     void loop(){
23         % Actualizar mediciones
24         nh.spinOnce();
25         delay(10);
26     }

```

```
usuario@usuario: ~ - 80x24
usuario@usuario:~$ rosrn rosserial_python serial_node.py /dev/ttyUSB0
[INFO] [1695077794.273046]: ROS Serial Python Node
[INFO] [1695077794.309694]: Connecting to /dev/ttyUSB0 at 57600 baud
[INFO] [1695077796.427839]: Requesting topics...
[INFO] [1695077799.663106]: Note: publish buffer size is 512 bytes
[INFO] [1695077799.669977]: Setup publisher on /tf [tf/tfMessage]
[INFO] [1695077799.684199]: Setup publisher on current_m [geometry_msgs/Vector3Stamped]
[INFO] [1695077799.704104]: Setup publisher on encoder_m [geometry_msgs/Vector3Stamped]
[INFO] [1695077799.718375]: Setup publisher on battery_m [std_msgs/UInt32]
[INFO] [1695077799.740291]: Note: subscribe buffer size is 512 bytes
[INFO] [1695077799.746749]: Setup subscriber on drive_cmd [geometry_msgs/Vector3]
```

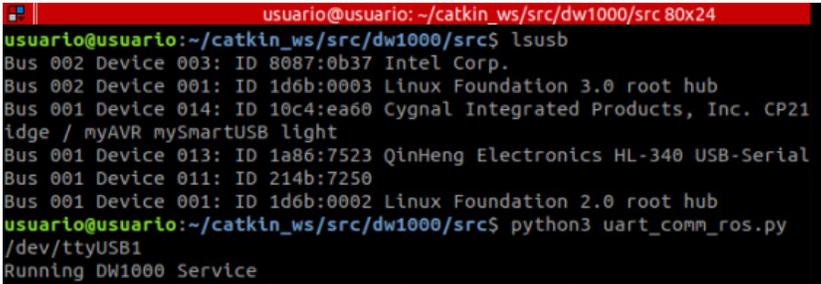
Figura 5.17. Salida en terminal de rosserial

## Tag UWB

Dentro del microcontrolador STM32F01 embebido en el transceptor BU01 se encuentra un firmware que ejecuta el algoritmo TWR (two-way ranging) basado en TDOA (Time difference of arrival) diseñado por el fabricante que permite encontrar la distancia entre el tag y varias anclas. Previo al uso es necesario asegurarse de que el dispositivo esté configurado en el modo adecuado a través de comandos AT. Para que un transceptor se encuentre en modo tag se debe iniciar una conexión con algún software tipo terminal con una velocidad de 115200 bps. Se enviará la instrucción  $AT+anchor\_tag=0$ , poniendo al dispositivo en modo Tag, este dispositivo es único e irá montado sobre el robot móvil. Los demás serán configurados como ancla empleando  $AT+anchor\_tag=1,noanchor$  donde *noanchor* será un número de 0-127 indicando el número de la ancla. Para iniciar la medición se enviará al tag el comando  $AT+switchdis=1$ , las anclas no necesitan recibir comandos adicionales una vez configuradas. La salida del Tag está codificada en texto y puede ser leída a través de la interfaz serie del microcontrolador. Al conectarse este dispositivo al puerto USB de la Jetson es reconocido por su  $VENDOR\_ID = 0x10c4$  y su  $PRODUCT\_ID = 0xea60$ , el nodo de ROS asociado al sensor UWB automáticamente encuentra el número de puerto serie al cual está asociado este dispositivo. Para diferenciar las distancias relativas a cada ancla en el tópico *uwb* se agrega como cabecera de em mensaje publicado el identificador del ancla. Una vez que se termina la conexión del nodo con ROS se envía un comando AT de paro al BU01.

```
1  # Librerías
2  def publisher():
3  # Dispositivos USB
4  dev = usb.core.find(idVendor=VENDOR_ID, idProduct=PRODUCT_ID)
5
6  rospy.init_node('dw1000', anonymous=False)
7  pub=rospy.Publisher("/uwb", Range, queue_size=5)
8
9  # Configurar mensaje
```

```
10
11 while not rospy.is_shutdown():
12     if sp.inWaiting() > 0:
13         data = sp.readline()
14         n=len(data)
15         if n>4:
16             anchor_id=data[2]
17             meas=float(data[4:n-3])
18             range_msg.header.frame_id=str(anchor_id-48)
19             range_msg.header.stamp=rospy.get_rostime()
20             range_msg.range=meas
21             #print(meas)
22             pub.publish(range_msg)
23             rate.sleep()
24 sp.write(b'AT+switchdis=0\r\n')
25 sp.close()
```



```
usuario@usuario:~/catkin_ws/src/dw1000/src 80x24
usuario@usuario:~/catkin_ws/src/dw1000/src$ lsusb
Bus 002 Device 003: ID 8087:0b37 Intel Corp.
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 014: ID 10c4:ea60 Cygnal Integrated Products, Inc. CP21
idge / myAVR mySmartUSB light
Bus 001 Device 013: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial
Bus 001 Device 011: ID 214b:7250
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
usuario@usuario:~/catkin_ws/src/dw1000/src$ python3 uart_comm_ros.py
/dev/ttyUSB1
Running DW1000 Service
```

Figura 5.18. Salida en terminal del nodo para el transceptor BU01

# Capítulo 6

## Localización del robot omnidireccional en una red RFID

Los robots móviles con ruedas (WMR) se conocen como sistemas robóticos omnidireccionales cuando son capaces de alcanzar cualquier posición en su entorno. El primer robot omnidireccional llamado Urano fue diseñado y construido en la Universidad Carnegie Melon [78]. Desde entonces, diferentes tipos de este tipo de robots han sido explotados en la industria [79], la medicina, los almacenes, la logística, etc. Uno de los tipos más eficientes de WMR es el robot omnidireccional de tres ruedas (TWOR) [80], que funciona eficazmente en ambientes interiores y exteriores. El TWOR tiene muchas ventajas prácticas, como la flexibilidad para viajar a lo largo de trayectorias industriales complejas y la capacidad de cambiar la posición y la orientación rápidamente.

Para representar el modelo cinemático-dinámico estándar de un TWOR, los autores utilizan principalmente dos metodologías basadas en el enfoque vectorial [81] y el enfoque de transformación [82]. Los primeros estudios extensos en esta área se realizaron en [83], donde se creó el modelo cinemático de un robot móvil utilizando la transformación de la matriz de coordenadas. Posteriormente, la formulación de Lagrange se usó en [84] para representar el modelo con ecuaciones de orden reducido utilizando una matriz de holonomía del sistema. Más recientemente, en [85] se analizó una estructura en cascada de TWOR con un bloque cinemático inverso para generar las referencias de velocidad requeridas por el controlador predictivo. Aumenta la robustez a las perturbaciones y también es más adecuado para problemas multivariantes con fenómenos de fricción.

En entornos interiores, la auto-localización de TWOR se puede organizar utilizando tecnología de banda ultra-ancha (UWB) [86], sistema de navegación inercial (INS), esquemas híbridos [87] y etiquetas de identificación por radiofrecuencia (RFID)

---

[88]. Este último enfoque ha llamado la atención debido a su bajo costo, bajo (o nulo) consumo de energía y un amplio rango de distancia [89]. También se puede combinar con otros métodos. Por ejemplo, un nuevo método de localización propuesto en [90] combina los datos de la etiqueta RFID con mediciones basadas en láser. En [91], las capacidades de una etiqueta RFID de potencia variable y una etiqueta RFID pasiva de ultra alta frecuencia (UHF) se emplean para crear redes en entornos complejos. Un sistema de localización diseñado en [92] combina dos tipos de señales generadas por etiquetas RFID con una estrategia de clasificación lógica. En [93], los autores resuelven el problema de localización de vehículos utilizando un esquema de rango multi-frecuencia de una sola antena y calculan la posición del vehículo a través de métodos de optimización no lineal de parámetros múltiples.

Independientemente del método utilizado, la mayor precisión de navegación se logra utilizando estimadores de estado. El estimador bayesiano desarrollado en [92] emplea distribuciones de proceso y estado sin usar un modelo, pero es computacionalmente ineficiente. En entornos gaussianos lineales, el problema de estimación de estado se resuelve de manera óptima mediante el filtro de Kalman (KF). Para los modelos no lineales, a menudo se utilizan el KF extendido (EKF), el KF sin perfume (UKF) y el filtro de partículas (PF). Una buena robustez la ofrece un filtro de respuesta de impulso finito no sesgado (UFIR) [94] y su versión extendida (EFIR) [95], que no requieren información de ruido medio cero y valores iniciales.

Vale la pena señalar que la navegación basada en etiquetas RFID está acompañada de ruido de medición coloreado (CMN) [96,97]. Para tratar con el CMN de Gauss-Markov se suele utilizar el algoritmo de Bryson [98] y el algoritmo de Petovello [99]. En el algoritmo de Bryson, el CMN se filtra en dos fases: suavizado y filtrado. En el algoritmo de Petovello, solo se utiliza la etapa de filtrado. Recientemente se diseñaron varios algoritmos de para Gauss-Markov CMN en [100] utilizando el método de Euler hacia atrás. Además, el problema con CMN fue abordado por Zhou et al. en [101] usando el segundo momento de información y por Ding et al. en [102] usando el modelo de promedio móvil auto-regresivo.

En esta sección, proponemos un modelo no lineal TWOR usando: 1) interpretación geométrica de perfiles de velocidad que varían suavemente y 2) representación de tiempo incremental en tiempo discreto. También desarrollamos los algoritmos de filtrado EKF y EFIR y los modificamos como cEKF y cEFIR para CMN. Mostramos que en entornos donde la señal de observación es obtenida desde etiquetas RFID, el filtro cEFIR supera significativamente al filtro EFIR, cEKF y EKF bajo CMN. El filtro cEFIR es casi tan robusto como el PF. Sin embargo, este último es menos práctico en tiempo real. Los principales resultados de esta sección son los siguientes:

- Un modelo simplificado para el TWOR que sea de fácil implementación en hardware/software considerando una computadora a bordo del robot móvil.



**Figura 6.1.** Plataforma de desarrollo para robótica móvil Festo Robotino [2].

- Algoritmos de filtrado cEFIR y cEKF para la localización de TWOR en CMN.
- Evidencia experimental de un mejor rendimiento del filtro cEFIR en comparación con el filtro EKF, cEKF, EFIR y PF.

## 6.1. Modelado cinemático del robot

El TWOR fabricado como plataforma de desarrollo robótico móvil de Festo Robotino [2] se muestra en la figura 6.1. Está diseñado para que el movimiento plano tenga tres grados de libertad (DOF). Dependiendo de los tipos de restricciones dinámicas y de la configuración espacial actual, un robot puede o no alcanzar otra configuración en el espacio. Para el robot móvil usado en este trabajo es posible controlar cada uno de los DOF de forma independiente. Si este es el caso, este tipo de configuración se conoce como *holonómico*, y puede viajar en cualquier dirección sin restricciones de orientación.

Considerando los resultados obtenidos en [82, 83], modificamos el diagrama cinemático del TWOR considerando que las ruedas omnidireccionales tienen rodillos perpendiculares como se muestra en la Fig. 6.2. Para determinar el modelo cinemático del robot consideraremos que es impulsado por los desplazamientos de sus ruedas. Suponiendo una condición sin deslizamiento, el desplazamiento  $\Delta u_j$ ,  $j \in [1, 3]$ , en la  $j$ -ésima rueda se puede encontrar proyectando el vector de desplazamiento  $u_j$  en el sistema de coordenadas local  $(x_L, y_L)$  y utilizando la contribución del segmento de arco causado por la rotación  $\Delta\varphi$  del robot. Lo anterior permite plantear la ecuación

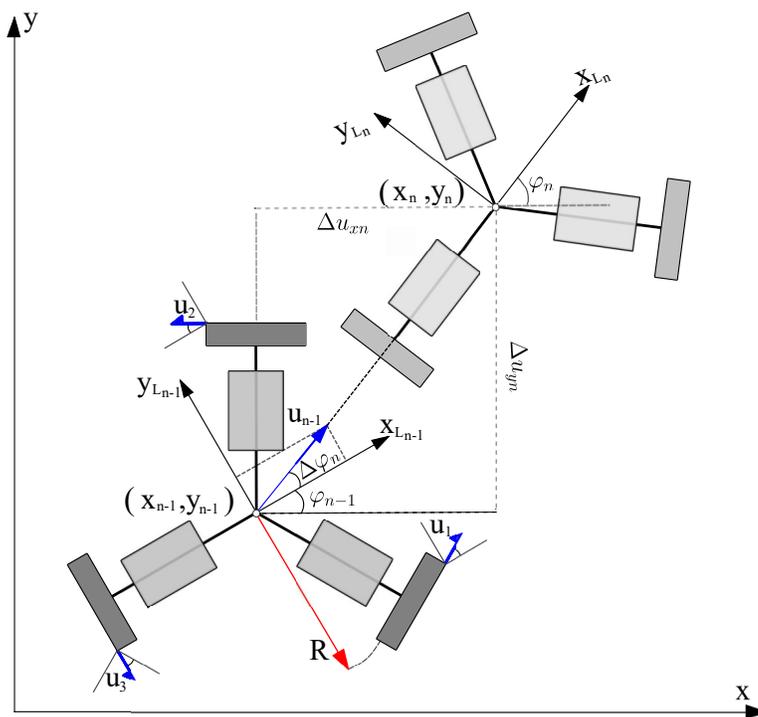


Figura 6.2. Diagrama cinemático del TWOR mostrado en la Fig. 6.1.

siguiente:

$$\Delta u_j = \left\| \text{proj}_{\vec{u}_j} \vec{u}_{x_L} \right\| + \left\| \text{proj}_{\vec{u}_j} \vec{u}_{y_L} \right\| + \Delta \varphi R, \quad (6.1)$$

donde  $R$  es el radio de la plataforma del robot. Además, usaremos (6.1) para representar la evolución del TWOR en el espacio de estados.

## 6.2. Representación en espacio de estados

Observando el diagrama cinemático (Fig. 6.2) y considerando el TWOR en dos puntos sucesivos en tiempo discreto, ahora podemos representar la dinámica no lineal del robot móvil en un espacio de estados.

### 6.2.1. Ecuación de Transición de Estado

Las variables de estado razonables para el TWOR son  $u_{x_L}$ ,  $u_{y_L}$  y  $\varphi$ . Usando estas variables y (6.1), reescribimos  $\Delta u_j$ ,  $j \in [1, 3]$ , en términos de los incrementos

$\Delta u_{x_L}$ ,  $\Delta u_{y_L}$ , y  $\Delta\varphi$  como

$$\begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{bmatrix} = \begin{bmatrix} \cos \delta & \sin \delta & R \\ -\cos \delta & \sin \delta & R \\ 0 & -1 & R \end{bmatrix} \begin{bmatrix} \Delta u_{x_L} \\ \Delta u_{y_L} \\ \Delta\varphi \end{bmatrix}, \quad (6.2a)$$

$$\Delta \mathbf{U}_i = \mathbf{A} \Delta \mathbf{U}_L, \quad (6.2b)$$

donde el vector  $\Delta \mathbf{U}_i$  de desplazamientos y el vector  $\Delta \mathbf{U}_L$  de variables de estado se definen comparando (6.2a) y (6.2b), aquí,  $\mathbf{A}$  es una matriz de mapeo definida por (6.2a). Tenga en cuenta que el vector  $\Delta \mathbf{U}_L$ , que contiene toda la información sobre la ubicación del robot en coordenadas locales, se puede determinar en términos de  $\Delta \mathbf{U}_i$  invirtiendo (6.2b). Dado que la cinemática del TWOR (Fig. 6.2) sugiere que  $\delta = \pi/6$ , tenemos

$$\begin{aligned} \Delta \mathbf{U}_L &= \mathbf{A}^{-1} \Delta \mathbf{U}_i, \\ \mathbf{A}^{-1} &= \begin{bmatrix} \sqrt{3}/3 & -\sqrt{3}/3 & 0 \\ 1/3 & 1/3 & -2/3 \\ 1/3R & 1/3R & 1/3R \end{bmatrix}. \end{aligned}$$

Hay dos formas de representar la dinámica del robot en el espacio de estados. Se pueden asignar los estados del robot y aproximar la trayectoria usando un modelo de estado polinómico [87] o proyectar la posición actual del robot un paso adelante usando las ecuaciones (6.2a) y (6.2b) [88, 103]. En nuestro caso, para encontrar la posición en el sistema de coordenadas global, introducimos el índice de tiempo discreto  $n$ , aplicamos el operador de rotación matricial  $\mathbf{R}(\varphi_n)$ , obtenemos  $\Delta \mathbf{U} = R(\varphi_n) \mathbf{U}_L$  y  $\Delta \mathbf{U} = R(\varphi_n) \mathbf{A}^{-1} \mathbf{U}_i$ , y proporciona algunas transformaciones.

$$\begin{aligned} \Delta u_{xn} &= \frac{\sqrt{3}}{3} \Delta u_{12n} \cos(\varphi_{n-1} + \Delta\varphi_n) \\ &\quad - \frac{1}{3} \Delta u_{123n} \sin(\varphi_{n-1} + \Delta\varphi_n), \end{aligned} \quad (6.3)$$

$$\begin{aligned} \Delta u_{yn} &= \frac{\sqrt{3}}{3} \Delta u_{12n} \sin(\varphi_{n-1} + \Delta\varphi_n) \\ &\quad + \frac{1}{3} \Delta u_{123n} \cos(\varphi_{n-1} + \Delta\varphi_n), \end{aligned} \quad (6.4)$$

$$\Delta\varphi_n = \frac{1}{3R} (\Delta u_{1n} + \Delta u_{2n} + \Delta u_{3n}), \quad (6.5)$$

Donde  $\Delta u_{12n} = \Delta u_{1n} - \Delta u_{2n}$  y  $\Delta u_{123n} = \Delta u_{1n} + \Delta u_{2n} - 2\Delta u_{3n}$ ,  $\Delta u_{xn}$ ,  $\Delta u_{yn}$  y  $\Delta\varphi_n$  son incrementos en las coordenadas del robot,  $x$ ,  $y$  y el ángulo  $\varphi$  son las variables de estado.

Las coordenadas  $x_{1n} = x_{xn}$ , y  $x_{2n} = x_{yn}$ , y  $x_{3n} = \varphi_n$  en  $n$  ahora se puede

representar con funciones no lineales

$$\begin{aligned} f_{1n} &= x_n = x_{1n} = x_{1(n-1)} + \Delta u_{xn}, \\ f_{2n} &= y_n = x_{2n} = x_{2(n-1)} + \Delta u_{yn}, \\ f_{3n} &= \varphi_n = x_{3n} = x_{3(n-1)} + \Delta \varphi_n, \end{aligned}$$

Donde los incrementos  $\Delta u_{xn}$ ,  $\Delta u_{yn}$  y  $\Delta \varphi_n$  están dados por (6.3)–(6.5).

Dado que  $\varphi_{n-1} = x_{3(n-1)}$ , tenemos las siguientes funciones no lineales con respecto a los estados del robot,

$$\begin{aligned} f_{1n} &= x_{1(n-1)} + \frac{\sqrt{3}}{3} \Delta u_{12n} \cos(x_{3(n-1)} + \Delta \varphi_n) \\ &\quad - \frac{1}{3} \Delta u_{123n} \sin(x_{3(n-1)} + \Delta \varphi_n), \end{aligned} \quad (6.6)$$

$$\begin{aligned} f_{2n} &= x_{2(n-1)} + \frac{\sqrt{3}}{3} \Delta u_{12n} \sin(x_{3(n-1)} + \Delta \varphi_n) \\ &\quad + \frac{1}{3} \Delta u_{123n} \cos(x_{3(n-1)} + \Delta \varphi_n) \end{aligned} \quad (6.7)$$

$$f_{3n} = x_{3(n-1)} + \Delta \varphi_n. \quad (6.8)$$

A continuación, introducimos el vector de estado  $\mathbf{x}_n = [x_{1n} \ x_{2n} \ x_{3n}]^T$  y el vector de control  $\mathbf{u}_n = [\Delta u_{1n} \ \Delta u_{2n} \ \Delta u_{3n}]^T$ , cuyas componentes son variables de incrementos  $\Delta u_{xn}$ ,  $\Delta u_{yn}$ , y  $\Delta \varphi_n$  dado por (6.6)–(6.8). También suponemos que el ruido TWOR es blanco, gaussiano, de media cero, y aditivo e introducimos el vector de ruido  $\mathbf{w}_n = [w_{1n} \ w_{2n} \ w_{3n}]^T \sim \mathcal{N}(0, \mathbf{Q})$  con la covarianza  $\mathbf{Q} = E\{w_n w_n^T\}$ . En este punto, la dinámica del TWOR se puede representar en un espacio de estados de tiempo discreto con la siguiente ecuación de estado no lineal

$$\mathbf{x}_n = f(\mathbf{x}_{n-1}, \mathbf{u}_n) + \mathbf{w}_n, \quad (6.9)$$

donde los componentes de la función vectorial no lineal  $f = [f_{1n} \ f_{2n} \ f_{3n}]^T$  se especifican por (6.6)–(6.8).

### 6.2.2. Ecuación de Observación

Un sistema de observación para localización en interiores puede crearse empleando etiquetas RFID, se puede organizar con  $k$ -etiquetas  $T_j$ ,  $j \in [1, k]$ , con coordenadas conocidas  $(\chi_j, \mu_j, \nu_j)$  de tal manera que en existan al menos tres etiquetas observables en cada  $n$ . La posición del TWOR se puede determinar mediante las distancias  $d_{jn}$  medidas entre las etiquetas y el lector en presencia de ruido aditivo  $v_{jn}$ . El ángulo de orientación del TWOR  $\varphi_n = x_{3n}$  se mide en coordenadas globales

directamente mediante un giroscopio digital, su valor  $\phi_n$  se asume con presencia de ruido aditivo  $v_{\varphi n}$ , y las ecuaciones de observación pueden así ser escrito como

$$\begin{aligned} d_{1n} &= \sqrt{(x_{1n} - \chi_1)^2 + (x_{2n} - \mu_1)^2 + \nu_1^2} + v_{1n}, \\ d_{2n} &= \sqrt{(x_{1n} - \chi_2)^2 + (x_{2n} - \mu_2)^2 + \nu_2^2} + v_{2n}, \\ &\vdots \\ d_{kn} &= \sqrt{(x_{1n} - \chi_k)^2 + (x_{2n} - \mu_k)^2 + \nu_k^2} + v_{kn}, \\ \phi_n &= x_{3n} + v_{\varphi n}. \end{aligned}$$

Dado que el ruido de medición basado en etiquetas RFID es coloreado [96, 97], representamos  $v_{jn}$  mediante el modelo de Gauss-Markov.

$$v_{jn} = \psi_j v_{j(n-1)} + \xi_{jn}, \quad (6.10)$$

donde  $\xi_{jn} \sim \mathcal{N}(0, \mathbf{R}_{jn})$  es ruido blanco gaussiano con media cero y covarianza  $\mathbf{R}_{jn}$ , el factor de color  $0 < \psi_j < 1$  para  $\psi_{jn} = 0$  hace que  $v_{jn}$  sea blanco. Para satisfacer la estabilidad  $\psi_j$  no debe exceder la unidad.

La ecuación de observación no lineal ahora se puede escribir como

$$y_n = h(\mathbf{x}_n) + \mathbf{v}_n, \quad (6.11)$$

donde  $y_n = [y_{1n} \ y_{2n} \ \dots \ y_{kn} \ y_{(k+1)n}]^T \in \mathbb{R}^{k+1}$  es el vector de observación, los componentes de la función no lineal  $h(\mathbf{x}_n) = [h_1(\mathbf{x}_n) \ h_2(\mathbf{x}_n) \ \dots \ h_k(\mathbf{x}_n) \ h_{k+1}(\mathbf{x}_n)]^T$  están dados por

$$\begin{aligned} h_1 &= \sqrt{(x_{1n} - \chi_1)^2 + (x_{2n} - \mu_1)^2 + \nu_1^2}, \\ h_2 &= \sqrt{(x_{1n} - \chi_2)^2 + (x_{2n} - \mu_2)^2 + \nu_2^2}, \\ &\vdots \\ h_k &= \sqrt{(x_{1n} - \chi_k)^2 + (x_{2n} - \mu_k)^2 + \nu_k^2}, \\ h_{k+1} &= x_{3n} \end{aligned}$$

y el CMN  $\mathbf{v}_n = [v_{1n} \ v_{2n} \ \dots \ v_{kn} \ v_{\varphi n}]^T \in \mathbb{R}^{k+1}$  se representa con

$$\mathbf{v}_n = \Psi_n \mathbf{v}_{n-1} + \boldsymbol{\eta}_n, \quad (6.12)$$

donde  $\boldsymbol{\eta}_n \sim \mathcal{N}(0, \mathbf{R}) \in \mathbb{R}^{k+1}$  es ruido blanco gaussiano con media cero de covarianza  $\mathbf{R}$  y la matriz del factor de coloración  $\Psi_n = \text{diag}(\psi_1 \ \psi_2 \ \dots \ \psi_k \ 0)$  es diagonal.

El modelo de espacio de estados es representado por las ecuaciones no lineales (6.9), (6.11) y (6.12) ahora se puede ampliar para utilizar estimadores lineales.

## 6.3. Filtros extendidos para ruido coloreado

### 6.3.1. Modelo Extendido del espacio de estados

El enfoque estándar para aplicar estimadores de estado lineales a modelos no lineales es expandir las funciones no lineales  $f(\mathbf{x}_{n-1}, \mathbf{u}_n)$  y  $h(\mathbf{x}_n)$  usando la serie de Taylor de primer orden. La función  $f(\mathbf{x}_{n-1}, \mathbf{u}_n)$  se puede expandir alrededor de  $\mathbf{x}_{n-1}$  como [95]

$$f(\mathbf{x}_{n-1}, \mathbf{u}_n) = f(\hat{\mathbf{x}}_{n-1}, \mathbf{u}_n) + \mathbf{F}_n(\mathbf{x}_{n-1} - \hat{\mathbf{x}}_{n-1}) + \mathbf{w}_n,$$

donde  $\hat{\mathbf{x}}_{n-1}$  es la estimación pasada disponible y  $\mathbf{F}_n = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{n-1}}$  es la matriz Jacobiana. La ecuación de estado extendida se puede entonces formalizar con

$$\mathbf{x}_n = \mathbf{F}_n \mathbf{x}_{n-1} + \tilde{\mathbf{u}}_n + \mathbf{w}_n, \quad (6.13)$$

donde  $\tilde{\mathbf{u}}_n = f(\hat{\mathbf{x}}_{n-1}, \mathbf{u}_n) - \mathbf{F}_n \hat{\mathbf{x}}_{n-1}$  es un vector conocido. De manera similar, la función  $h(\mathbf{x}_n)$  se puede expandir alrededor de  $\mathbf{x}_n^-$  como

$$h(\mathbf{x}_n) = h(\hat{\mathbf{x}}_n^-) + \mathbf{H}_n(\mathbf{x}_n - \hat{\mathbf{x}}_n^-) + \mathbf{v}_n,$$

donde  $\hat{\mathbf{x}}_n^- \triangleq \hat{\mathbf{x}}_{n|n-1} = \mathbf{F}_n \hat{\mathbf{x}}_{n-1|n-1}$  es la estimación previa disponible en  $n$  y  $\mathbf{H}_n = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_n^-}$  es la matriz Jacobiana.

Introduciendo un nuevo vector de observación  $\zeta_n = y_n - \tilde{s}_n$ , donde  $\tilde{s}_n = h(\hat{\mathbf{x}}_n^-) - \mathbf{H}_n \hat{\mathbf{x}}_n^-$  es una función conocida, la ecuación de observación extendida (6.11) adquiere la forma

$$\zeta_n = \mathbf{H}_n \mathbf{x}_n + \mathbf{v}_n, \quad (6.14)$$

donde  $\mathbf{v}_n$  es un ruido que sigue a un proceso de Gauss-Markov (6.12). Las ecuaciones extendidas (6.13) y (6.14) se pueden modificar aún más para CMN usando diferenciación de medidas como se muestra a continuación.

### 6.3.2. Modificación del sistema para Ruido Coloreado

Introduciendo una nueva observación  $z_n$ , usando diferenciación de mediciones y siguiendo [100], obtenemos

$$\begin{aligned} z_n &= \zeta_n - \Psi_n \zeta_{n-1} \\ &= \mathbf{H}_n \mathbf{x}_n + \mathbf{v}_n - \Psi_n \mathbf{H}_{n-1} \mathbf{x}_{n-1} - \Psi_n \mathbf{v}_{n-1} \\ &= \mathbf{H}_n \mathbf{x}_n + \mathbf{v}_n - \Psi_n \mathbf{v}_{n-1} \\ &\quad - \Psi_n \mathbf{H}_{n-1} (\mathbf{F}_n^{-1} \mathbf{x}_n - \mathbf{F}_n^{-1} \mathbf{w}_n - \mathbf{F}_n^{-1} \tilde{\mathbf{u}}_n) \\ &= \mathbf{D}_n \mathbf{x}_n + \Gamma_n \tilde{\mathbf{u}}_n + \bar{\mathbf{v}}_n, \end{aligned} \quad (6.15)$$

Donde

$$\mathbf{D}_n = \mathbf{H}_n - \Gamma_n \quad (6.16)$$

$$\Gamma_n = \Psi_n \mathbf{H}_{n-1} \mathbf{F}_n^{-1}, \quad (6.17)$$

$$\bar{v}_n = \Gamma_n w_n + \eta_n. \quad (6.18)$$

A continuación, eliminamos el error de sesgo de (6.15) como  $\bar{z}_n = z_n - \Gamma_n \tilde{u}_n$ , donde se conoce  $\Gamma_n \tilde{u}_n$ . Para así obtener la nueva ecuación de observación.

$$\bar{z}_n = \mathbf{D}_n x_n + \bar{v}_n, \quad (6.19)$$

en el que, según sea necesario,  $\bar{v}_n$  es blanco gaussiano con las propiedades

$$E\{\bar{v}_n \bar{v}_n^T\} = \Gamma_n \mathbf{Q}_n \Gamma_n^T + \mathbf{R}_n, \quad (6.20)$$

$$E\{\bar{v}_n w_n^T\} = \Gamma_n \mathbf{Q}_n, \quad (6.21)$$

$$E\{\bar{w}_n v_n^T\} = \mathbf{Q}_n \Gamma_n^T. \quad (6.22)$$

Como puede verse,  $\bar{v}_n$  está correlacionado con  $w_n$ . Para aplicar el EKF, se requiere una nueva ganancia de Kalman para  $\bar{v}_n$  y  $w_n$  correlacionados en el tiempo.

### 6.3.3. Filtro de Kalman Extendido para ruido coloreado

Para derivar la ganancia de Kalman para  $w_n$  y  $\bar{v}_n$  correlacionados en el tiempo, definimos el error de estimación como  $\epsilon_n = x_n - \hat{x}_n$  y el error de estimación anterior como  $\epsilon_n^- = x_n - \hat{x}_n^-$ , donde  $\hat{x}_n^-$  es la estimación previa de  $x_n$ . A continuación, seguimos [100] y consideramos las siguientes funciones: la covarianza del error previo  $\mathbf{P}_n^- = E\{\epsilon_n^- \epsilon_n^{-T}\}$ ,

$$\mathbf{P}_n^- = \mathbf{F}_n \mathbf{P}_{n-1} \mathbf{F}_n^T + \mathbf{Q}_n, \quad (6.23)$$

con un residuo de medición  $\bar{s}_n = \bar{z}_n - \mathbf{D}_n \hat{x}_n^-$ ,

$$s_n = \mathbf{D}_n \mathbf{F}_n \epsilon_{n-1} + \mathbf{D}_n w_n + \bar{v}_n, \quad (6.24)$$

La covarianza de innovación  $\bar{\mathbf{S}}_n = E\{\bar{s}_n \bar{s}_n^T\}$  se obtiene usando (6.24),

$$\mathbf{S}_n = \mathbf{D}_n \mathbf{P}_n^- \mathbf{D}_n^T + \mathbf{H}_n \mathbf{Q}_n \Gamma_n^T + \Gamma_n \mathbf{Q}_n \mathbf{D}_n^T + \mathbf{R}_n, \quad (6.25)$$

donde  $\Gamma_n = \mathbf{H}_n - \mathbf{D}_n$ , el error de estimación será

$$\begin{aligned} \epsilon_n &= (\mathbf{I} - \mathbf{K}_n \mathbf{D}_n) \mathbf{F}_n \epsilon_{n-1} \\ &\quad + (\mathbf{I} - \mathbf{K}_n \mathbf{D}_n) w_n - \mathbf{K}_n \bar{v}_n, \end{aligned} \quad (6.26)$$

Donde  $\mathbf{I}$  denota a la matriz identidad con dimensiones apropiadas. La covarianza del error se obtiene a través de (6.26) como

$$\begin{aligned}
 \mathbf{P}_n &= (\mathbf{I} - \mathbf{K}_n \mathbf{D}_n) \mathbf{P}_n^- (\mathbf{I} - \mathbf{K}_n \mathbf{D}_n)^T + \mathbf{K}_n \mathbf{R}_n \mathbf{K}_n^T \\
 &\quad - (\mathbf{I} - \mathbf{K}_n \mathbf{D}_n) \mathbf{Q}_n \Gamma_n^T \mathbf{K}_n^T + \mathbf{K}_n \Gamma_n \mathbf{Q}_n \Gamma_n^T \mathbf{K}_n^T \\
 &\quad - \mathbf{K}_n \Gamma_n \mathbf{Q}_n (\mathbf{I} - \mathbf{K}_n \mathbf{D}_n)^T \\
 &= \mathbf{P}_n^- + \mathbf{K}_n \mathbf{S}_n \mathbf{K}_n^T - (\mathbf{P}_n^- \mathbf{D}_n^T + \mathbf{Q}_n \Gamma_n^T) \mathbf{K}_n^T \\
 &\quad - \mathbf{K}_n (\mathbf{P}_n^- \mathbf{D}_n^T + \mathbf{Q}_n \Gamma_n^T)^T.
 \end{aligned} \tag{6.27}$$

Debido a que la traza de  $\mathbf{P}_n$  es convexa, su mínimo puede ser obtenido a través de la primer derivada de (6.27) con respecto a la ganancia de corrección de sesgo  $\mathbf{K}_n$  como

$$\frac{\partial \text{tr}(\mathbf{P}_n)}{\partial \mathbf{K}_n} = -2(\mathbf{P}_n^- \mathbf{D}_n^T + \mathbf{Q}_n \Gamma_n^T) + 2\mathbf{K}_n \mathbf{S}_n = 0, \tag{6.28}$$

La solución a la ecuación anterior produce la ganancia óptima para la corrección de sesgo considerando ruido correlacionado.

$$\mathbf{K}_n = (\mathbf{P}_n^- \mathbf{D}_n^T + \mathbf{Q}_n \Gamma_n^T) \mathbf{S}_n^{-1}. \tag{6.29}$$

Finalmente, sustituyendo  $\mathbf{K}_n$  de la ec. (6.29) en (6.27) se obtiene

$$\mathbf{P}_n = (\mathbf{I} - \mathbf{K}_n \mathbf{D}_n) \mathbf{P}_n^- - \mathbf{K}_n \Gamma_n \mathbf{Q}_n. \tag{6.30}$$

Un pseudocódigo para el cEKF considerando  $w_n$  y  $\bar{v}_n$  correlacionadas esta dado en Alg. 2. Este algoritmo requiere las funciones no lineales para actualizar las estimaciones y las matrices de las ecuaciones linealizadas.

### 6.3.4. Filtro FIR Extendido para ruido coloreado

La ventaja del filtro UFIR robusto derivado originalmente en [104] es que no requiere ninguna información sobre las estadísticas de los ruidos de proceso y medición. El único factor de ajuste en este filtro es el horizonte promedio, que debe ser óptimo,  $N_{\text{opt}}$  a fin minimizar el error cuadrático medio (MSE). Para desarrollar el filtro UFIR para el modelo no lineal (6.9) y (6.10) usando series de Taylor, seguimos [94] y usamos el modelo lineal (6.13) y (6.20). En el horizonte  $[m, n]$  de  $N$  puntos, de  $m = n - N + 1$ , la ecuación de estado (6.13) se puede extender como se indica en [95] obteniendo

$$\mathbf{X}_{m,n} = \mathbf{F}_{m,n} x_m + \mathbf{B}_{m,n} (\mathbf{U}_{m,n} + \mathbf{W}_{m,n}), \tag{6.31}$$

**Algoritmo 2:** Algoritmo cEKF para  $w_n$  y  $\bar{v}_n$  correlacionadas

---

**Datos:**  $\varsigma_n, \hat{x}_0, u_k, \Psi_n, P_0, Q_n, R_n$

**Resultado:**  $\hat{x}_n, P_n$

**inicio**

**para**  $n = 1, 2, \dots$  **hacer**

$$\begin{aligned}
 & z_n = \varsigma_n - \Psi_n \varsigma_{n-1}; \\
 & \hat{x}_n^- = f(\hat{x}_{n-1}, u_n); \\
 & P_n^- = F_n P_{n-1} F_n^T + Q_n; \\
 & S_n = H_n P_n^- H_n^T - \Psi_n H_{n-1} F_n^{-1} (P_n^- - Q_n) H_n^T \\
 & \quad + \Psi_n H_{n-1} F_n^{-1} (P_n^- - Q_n) F_n^{-T} H_{n-1}^T \Psi_n \\
 & \quad - H_n (P_n^- - Q_n) F_n^{-T} H_{n-1}^T \Psi_n + R_n; \\
 & K_n = [P_n^- H_n^T - (P_n^- - Q_n) F_n^{-T} H_{n-1}^T \Psi_n] S_n^{-1}; \\
 & \hat{x}_n = \hat{x}_n^- + K_n [z_n - h(\hat{x}_n^-)]; \\
 & P_n = (I - K_n H_n) P_n^-; \\
 & \quad + K_n \Psi_n H_{n-1} F_n^{-1} (P_n^- - Q_n);
 \end{aligned}$$


---

donde  $\mathbf{X}_{m,n} = [x_m^T \ x_{m+1}^T \ \dots \ x_n^T]^T$ ,  $\mathbf{U}_{m,n} = [\tilde{u}_m^T \ \tilde{u}_{m+1}^T \ \dots \ \tilde{u}_n^T]^T$ , y  $\mathbf{W}_{m,n} = [w_m^T \ w_{m+1}^T \ \dots \ w_n^T]^T$  son los vectores extendidos, las matrices extendidas en bloque son:

$$\mathbf{F}_{m,n} = [\mathbf{I} \ \mathbf{F}_{m+1}^T \ \dots \ (\mathcal{F}_{n-1}^{m+1})^T \ (\mathcal{F}_n^{m+1})^T]^T, \quad (6.32)$$

$$\mathbf{B}_{m,n} = \begin{bmatrix} \mathbf{I} & 0 & \dots & 0 & 0 \\ \mathbf{F}_{m+1} & \mathbf{I} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathcal{F}_{n-1}^{m+1} & \mathcal{F}_{n-1}^{m+2} & \dots & \mathbf{I} & 0 \\ \mathcal{F}_n^{m+1} & \mathcal{F}_n^{m+2} & \dots & \mathbf{F}_n & \mathbf{I} \end{bmatrix}, \quad (6.33)$$

y el producto de matrices  $\mathcal{F}_r^g$  esta dado

$$\mathcal{F}_r^g = \begin{cases} \mathbf{F}_r \mathbf{F}_{r-1} \dots \mathbf{F}_g, & g < r+1, \\ \mathbf{I}, & g = r+1 \\ \mathbf{0}, & g > r+1 \end{cases}. \quad (6.34)$$

Asumimos un estado inicial conocido  $x_m$ , por ello  $\tilde{u}_m = 0$  y  $w_m = 0$ .

De forma similar, asignamos los vectores extendidos  $\mathbf{Y}_{m,n} = [z_m^T \ z_{m+1}^T \ \dots \ z_n^T]^T$  y  $\mathbf{V}_{m,n} = [v_m^T \ v_{m+1}^T \ \dots \ v_n^T]^T$ , las ecuaciones del sistema de observación extendido sera

$$\mathbf{Y}_{m,n} = \mathbf{H}_{m,n} x_m + \mathbf{L}_{m,n} \mathbf{U}_{m,n} + \mathbf{G}_{m,n} \mathbf{W}_{m,n} + \mathbf{V}_{m,n}, \quad (6.35)$$

Con la matriz de observación dada por

$$\mathbf{H}_{m,n} = \begin{bmatrix} \mathbf{D}_m (\mathcal{F}_k^{m+1})^{-1} \\ \mathbf{D}_{m+1} (\mathcal{F}_k^{m+2})^{-1} \\ \vdots \\ \mathbf{D}_{n-1} \mathbf{F}_n^{-1} \\ \mathbf{D}_n \end{bmatrix}, \quad (6.36)$$

Las demás matrices en bloque están definidas de la siguiente forma  $\mathbf{L}_{m,n} = \mathbf{D}_{m,n}\mathbf{E}_{m,n}$ ,  $\mathbf{G}_{m,n} = \mathbf{D}_{m,n}\mathbf{B}_{m,n}$ , y  $\mathbf{D}_{m,n} = \text{diag}(\mathbf{D}_m \mathbf{D}_{m+1} \cdots \mathbf{D}_n)$ . El algoritmo requiere ser iniciado en un horizonte corto sobre un vector de datos  $\mathbf{Y}_{m,s}$ , donde  $s = m + K - 1$  y  $K$  es el número de estados. La estimación no tendrá sesgo si se cumple la condición  $E\{\mathbf{x}_n\} = E\{\hat{\mathbf{x}}_n\}$ . Para especificar el estado  $\bar{\mathbf{x}}_s$ , lo proyectamos secuencialmente desde  $m$  a  $s$  a través de la función no lineal  $f(\mathbf{x}_n)$ . Para obtener el estado no disponible  $\bar{\mathbf{x}}_m$  durante los primeros  $N$  puntos, empleamos un KF de soporte. También nos referimos a [103] y asignamos  $\mathbf{G}_s = \mathbf{I}$ .

El pseudocódigo de los algoritmos de filtrado cEFIR desarrollados para CMN aparece como Algoritmo 3. Vale la pena señalar que, dado que el filtro cEFIR ignora

---

**Algoritmo 3:** Algoritmo cEFIR para CMN

---

**Datos:**  $\varsigma_n$

**Resultado:**  $\hat{\mathbf{x}}_k$

**inicio**

```

para  $k = N - 1, N, \dots$  hacer
     $m = k - N + 1, s = m + K - 1$ 
     $\mathbf{G}_s = \mathbf{I}$ 
    para  $j = 1 : K - 1$  hacer
         $\bar{\mathbf{x}}_{m+j} = f(\bar{\mathbf{x}}_m)$ 
    para  $l = s + 1 : k$  hacer
         $z_l = \varsigma_l - \Psi_l \varsigma_{l-1}$ 
         $\hat{\mathbf{x}}_l^- = f(\bar{\mathbf{x}}_{l-1})$ 
         $\mathbf{G}_l = [D_l^T D_l + (F_l G_{l-1} F_l^T)^{-1}]^{-1}$ 
         $\mathbf{K}_l = \mathbf{G}_l D_l^T$ 
         $\bar{\mathbf{x}}_l = F_l \bar{\mathbf{x}}_{l-1} + \mathbf{K}_l [z_l - h(\hat{\mathbf{x}}_l^-) + \Psi_l h(\bar{\mathbf{x}}_{l-1})]$ 
     $\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k$ 

```

---

el ruido medio cero, la correlación temporal entre el  $w_n$  blanco y el  $\bar{v}_n$  se ignora, a diferencia de cEKF. Para determinar  $N_{\text{opt}}$  en el sentido MSE [105], primero calculamos la covarianza de error del filtro cEFIR mediante (6.27), donde establecemos  $\mathbf{K}_n = \mathbf{G}_n \mathbf{D}_n^T$ ,

$$\mathbf{P}_n = \mathbf{P}_n^- - (2\mathbf{P}_n^- \mathbf{D}_n^T + 2\mathbf{Q}_n \Gamma_n^T + \mathbf{G}_n \mathbf{D}_n^T \mathbf{S}_n) \mathbf{D}_n \mathbf{G}_n. \quad (6.37)$$

Antes de la estimación es necesario encontrar numéricamente  $N_{\text{opt}}$  mediante la minimización de la traza de (6.36) como se muestra en [105].

# Capítulo 7

## Localización del TWOR en una red de sensores UWB

En este capítulo se presenta el desarrollo de un sistema de seguimiento para un robot móvil omnidireccional de tres ruedas empleando como sistema de observación una red de sensores de banda ultraancha (UWB). En realidad, los robots móviles con ruedas (WMR) tienen aplicaciones en muchos campos como el cuidado personal [106], la salud [107] y los procesos de fabricación [108]. Los robots móviles omnidireccionales (OMR) son los vehículos que cada vez llaman más la atención por su capacidad de adoptar cualquier postura independientemente de su orientación. Hoy en día, los robots móviles parecen ser una nueva tendencia en la Industria 4.0 [109], el control [110], el análisis de movimiento [111] y la planificación de rutas [112] son desafíos comunes en el diseño de una robot, pero con la necesidad de robots cooperativos la dificultad de las tareas de localización está aumentando [113]. El robot necesita datos para realizar una tarea. Utilizando varios sensores de alcance de banda ultraancha (UWB), estimamos la localización del OMR. La tecnología UWB puede lograr una localización precisa [114] y es una excelente opción para implementar redes de sensores debido a su eficiencia en el consumo de energía y su robustez en entornos hostiles. Aquí, discutimos el caso del robot omnidireccional de tres ruedas (TWOR) que se mueve a lo largo de una red de sensores UWB [115]. En [116, 117] los autores presentan un modelo para la medición del alcance UWB. Los métodos clásicos de estimación del estado se encuentran en [118–121]. Algunos problemas en el alcance UWB se tratan en [122, 123] donde los autores discutieron la mitigación del error cuando los sensores están fuera de la vista.

## 7.1. Descripción del robot móvil

Equipamos el TWOR con tres ruedas omnidireccionales (OMW) [124]. Constan de un conjunto de rodillos inferiores dispuestos alrededor de un eje de rotación. En nuestro caso, los ejes de los rodillos y la rueda son perpendiculares. Los ejes de rotación del eje omnidireccional son independientes entre sí. Consideramos que la señal de actuación  $u_n$  representa el desplazamiento de la rueda. Un sistema de transmisión omnidireccional requiere al menos tres puntos de intersección entre los ejes de los rodillos inferiores para satisfacer la condición de rango completo para el jacobiano de la matriz de orientación de las ruedas [112]. Para la topología del robot, utilizamos la configuración de simetría rotacional [125]. Para equilibrar la reacción vertical en las ruedas y, en consecuencia, conseguir fuerzas de fricción longitudinal similares. Los TWOR utilizados en nuestro experimento tienen un diámetro de 60 cm y utilizan ruedas omnidireccionales con un radio de 5 cm. Dependiendo de las restricciones dinámicas, un robot móvil con ruedas puede controlar cada DOF de forma independiente. En la Fig. 6.2 se muestra un diagrama cinemático de un robot omnidireccional de tres ruedas (TWOR) con rodillos perpendiculares. Usamos las siguientes notaciones y símbolos para describir el modelo del sistema.

- Coordenadas del sistema global
  - $x_n, y_n$  Posición del robot
  - $x, y$  Sistema coordenado global
  - $\Phi_n$  Orientación del sistema coordenado local con respecto al global
- Coordenadas del sistema local
  - $p_n$  Desplazamiento total del robot
  - $\varphi_n$  Ángulo de desplazamiento con respecto del sistema local
  - $u_i$  Segmento de arco por paso
  - $x_L, y_L$  Sistema coordenado local
- Constantes del modelo
  - $R$  Radio del Robot

El robot es impulsado por el desplazamiento de sus ruedas. Suponiendo una condición de balanceo puro, el desplazamiento  $\Delta u_j$  se puede encontrar mediante:

$$\Delta u_j = \left\| \text{proj}_{\vec{u}_j} \vec{u}_{x_L} \right\| + \left\| \text{proj}_{\vec{u}_j} \vec{u}_{y_L} \right\| + \Delta \varphi R, \quad (7.1)$$

De (7.1), la cinemática del robot en la representación del espacio de estados es la siguiente:

$$\begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{bmatrix} = \begin{bmatrix} \cos \pi/6 & \sin \pi/6 & R \\ -\cos \pi/6 & \sin \pi/6 & R \\ 0 & -1 & R \end{bmatrix} \begin{bmatrix} \Delta u_{x_L} \\ \Delta u_{y_L} \\ \Delta \varphi \end{bmatrix}, \quad (7.2)$$

$$\Delta U_i = \mathbf{A} \Delta U_L$$

Donde  $\Delta U_i$  es el vector de desplazamientos,  $\Delta U_L$  es el vector de posición del robot en el marco local y  $\mathbf{A}$  es la matriz del sistema. Al invertir (7.2), tenemos

$$\Delta U_L = A^{-1} \Delta U_i$$

$$A^{-1} = \begin{bmatrix} \sqrt{3}/3 & -\sqrt{3}/3 & 0 \\ 1/3 & 1/3 & -2/3 \\ 1/3R & 1/3R & 1/3R \end{bmatrix}. \quad (7.3)$$

Y aplicando una rotación para transformarnos en el marco global, obtenemos

$$\begin{aligned} \Delta U &= R(\Phi) U_L \\ \Delta U &= R(\Phi) A^{-1} U_i \end{aligned} \quad (7.4)$$

Entonces, cada coordenada se puede escribir como:

$$\Delta u_x = \frac{\sqrt{3}}{3} \Delta u_{12} \cos(\Phi_{n-1}) - \frac{1}{3} \Delta u_{123} \sin(\Phi_{n-1}), \quad (7.5)$$

$$\Delta u_y = \frac{\sqrt{3}}{3} \Delta u_{12} \sin(\Phi_{n-1}) + \frac{1}{3} \Delta u_{123} \cos(\Phi_{n-1}), \quad (7.6)$$

$$\Delta \varphi = \frac{1}{3R} (\Delta u_1 + \Delta u_2 + \Delta u_3), \quad (7.7)$$

Donde  $\Delta u_{12} = \Delta u_1 - \Delta u_2$  y  $\Delta u_{123} = \Delta u_1 + \Delta u_2 - 2\Delta u_3$ . Podemos medir el desplazamiento real de la rueda usando encoders incrementales, por lo tanto se conocen las cantidades  $\Delta u_1$ ,  $\Delta u_2$  y  $\Delta u_3$ . El modelo se puede simplificar considerando  $p_n$  y  $\Delta \varphi$  como variables exógenas. Haciendo algunas transformaciones trigonométricas en (7.5-6) podemos demostrar que:

$$p_n = \frac{1}{3} \sqrt{3\Delta u_{12}^2 + \Delta u_{123}^2}, \quad (7.8)$$

$$\varphi_n = \Delta \varphi_n + \Phi_{n-1}, \quad (7.9)$$

Introduciendo el vector de estado  $\mathbf{x}_n = [x_{1n} \ x_{2n} \ x_{3n}]^T$  de las coordenadas globales, con los componentes

$$\begin{aligned} x_{1n} &= x_{1n-1} + p_n \cos(x_{3n-1} + \Delta \varphi_n), \\ x_{2n} &= x_{2n-1} + p_n \sin(x_{3n-1} + \Delta \varphi_n), \\ x_{3n} &= x_{3n-1} + \Delta \varphi_n, \end{aligned} \quad (7.10)$$

Podemos escribir la ecuación de estado no lineal como:

$$\mathbf{x}_n = \mathbf{f}_n(\mathbf{x}_{n-1}, \mathbf{u}_n, \mathbf{w}_n), \quad (7.11)$$

donde  $w_n \sim \mathcal{N}(0, \mathbf{Q}_n)$  es cero ruido blanco gaussiano medio con la covarianza  $\mathbf{Q}_n$ .

### 7.1.1. Modelo de Observación

Implementamos nuestra configuración utilizando el transceptor Decawave DW1000 con un ancho de banda de 900 MHz y compatible con IEEE 802.15.4a. La distancia entre el ancla y las etiquetas se calculará utilizando el alcance bidireccional asimétrico de doble cara (AltDS-TWR) [126, 127]. Básicamente, AltDS-TWR deriva la estimación del alcance basándose en el retraso de ida y vuelta entre dos ráfagas de datos. Este algoritmo tiene el error más pequeño debido a la deriva del reloj [1] incluso si el error aumenta con la distancia, el orden es inferior a 30 cm para una deriva del reloj  $xi < 3ppm$  y una  $t_{respuesta} < 650\mu s$ . Tomamos los datos de distancia de múltiples módulos DWM1000 configurados como ancla. También asumimos que la posición de cada ancla es fija y conocida. Se adjunta al TWOR un módulo DWM1000 adicional configurado como etiqueta. La figura 7.1 muestra el entorno de prueba utilizado en la evaluación experimental. Podemos escribir la distancia entre el ancla k y la etiqueta como:

$$r_k = \sqrt{(x_k - x_{1n})^2 + (y_k - x_{2n})^2} \quad (7.12)$$

Donde  $x_k, y_k$  son la posición del ancla k. Suponemos que no hay cambios en el nivel z y que los módulos transceptores (etiquetas de anclaje) están alineados en z. El patrón de radiación (Fig. 7.1) en el plano horizontal es omnidireccional [115] si los transceptores están orientados verticalmente.

### Fuentes de error de medición

El caso ideal aparece cuando los transceptores están en línea de visión (LOS), aquí la única fuente de error es sistemática. Varios factores como la reflexión y la difracción afectarán las mediciones [128]. Cuando el robot navega en entornos complejos, puede estar en condiciones sin línea de visión (NLOS) o fenómenos de trayectorias múltiples, lo que puede afectar la precisión de las mediciones. Un modelo determinista para el modo de canal UWB debe ser capaz de describir condiciones LOS o NLOS al mismo tiempo sin información a priori. Debido a las dificultades presentadas en escenarios reales, proponemos una prueba experimental previa para encontrar un modelo para la atenuación de la señal en condiciones LOS.

$$d_k = \sqrt{(\mathbf{a}_k - \bar{\mathbf{x}}_n)^T (\mathbf{a}_k - \bar{\mathbf{x}}_n)} + \text{bias}(r_k) + v_k \quad (7.13)$$

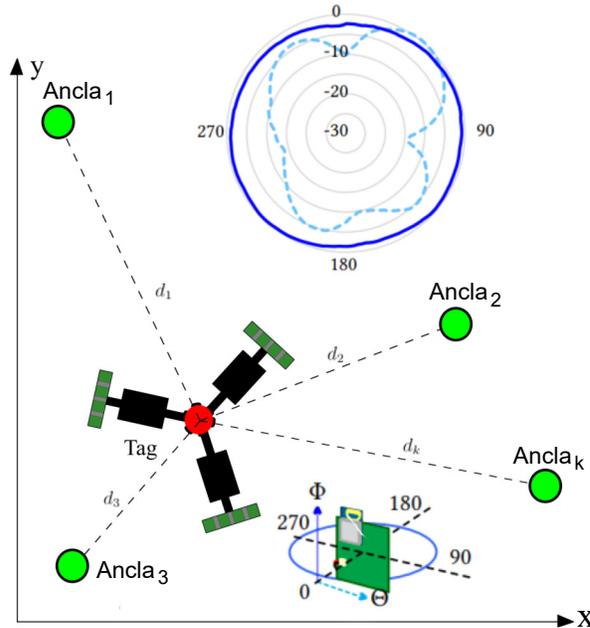


Figura 7.1. Esquema general de la localización en una red de sensores UWB

Donde  $a_k$  es el vector de coordenadas X-Y de la etiqueta, y  $\bar{x}_n$  son los dos primeros elementos del vector de posición  $x_n$ . El vector  $v_n$  es el ruido de observación modelado como una variable aleatoria con distribución gaussiana blanca, media cero y covarianza  $Q_n$ . Podemos escribir el modelo de observación no lineal como:

$$d_n = g_n(x_n, v_n) \tag{7.14}$$

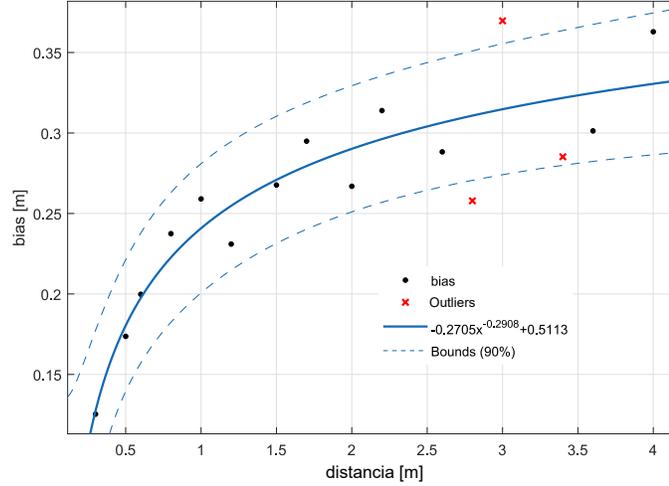
## 7.2. Seguimiento del TWOR

En esta sección, vamos a comparar estimadores de posición clásicos, como mínimos cuadrados y Kalman extendido, con estimadores FIR, utilizando un modelo de observación simple y el modelo con corrección de sesgo. Proponemos un modelo de covarianza para mitigar el error debido a NLOS mediante discriminación binaria.

### 7.2.1. Caracterización del sensor UWB en LOS

Nuestro entorno de caracterización era un escenario ideal, con conexión LOS entre la etiqueta y el ancla. En esta condición, la señal medida corresponde a las distancias entre transeptores con algo de ruido aditivo. Las mediciones se realizan

en un área de 5m x 8m sin obstáculos entre los transeptores. Las paredes no pueden inducir un error de trayectoria múltiple, porque la distancia entre los transeptores es pequeña, en comparación con la distancia de la trayectoria de trayectoria múltiple. Con los módulos DWM1000 alineados verticalmente y separados del suelo, a una distancia de 1,5m, estamos realizando mediciones de alcance. Cuando la distancia entre transeptores aumenta, obtenemos un sesgo mayor, como se muestra en la figura 7.2 .



**Figura 7.2.** Incremento del bias como una función de la distancia

En condiciones LOS, la interferencia destructiva con la señal reflejada en el suelo puede causar atenuación, lo que lleva a fenómenos de polarización. Para obtener un modelo aproximado, ajustamos el sesgo a una función polinómica, la ecuación 13 ahora es:

$$r_k = \sqrt{(a_k - \bar{x}_n)^T (a_k - \bar{x}_n)} \quad (7.15)$$

$$d_k = r_k - 0.2705r_k^{-0.2908} + 0.5113 + v_k$$

Establecemos la covarianza de  $v_k$  como la media de los valores experimentales, entonces  $\sigma_v^2 = 0.0105$  Finalmente, el modelo de observación debe ser capaz de minimizar la influencia de las medidas NLOS en la estimación, para ello definimos una regla binaria:

$$\sigma_{v,k}^2 = \begin{cases} 0.0105 & |r_k - \mathbf{H}_n \hat{x}_n| < \epsilon \\ 100 & otherwise \end{cases} \quad (7.16)$$

Con la restricción en la Ec.16 el valor de  $\epsilon$  actúa como un umbral, cuando la medición  $k$  está lejos de la distancia predicha, debemos suponer un valor atípico. Y luego, establecemos su covarianza como un valor "grande". Esto provocará que el estimador tenga menos confianza en la medición. Y luego minimizar su impacto en la estimación estatal.

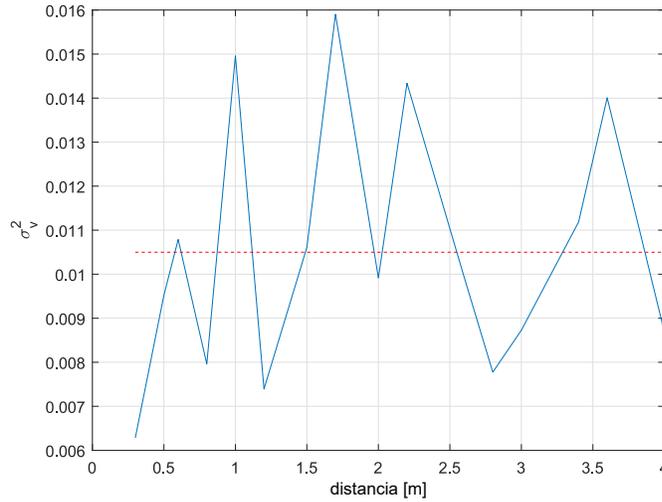


Figura 7.3. Covarianza de las mediciones

### 7.2.2. Localización del Tag en la red UWB

Cuando el robot navega hacia un escenario real, sin conocimiento previo de la configuración del espacio, necesitará muchos nodos de anclaje estáticos con una posición conocida. Vamos a comparar con diferentes enfoques.

#### Estimador por mínimos cuadrados

Se supone que un conjunto de medidas discretas  $y_n$  es una función lineal de parámetros desconocidos y algo de ruido aditivo. Entonces, para el modelo lineal  $y_n = \mathbf{H}_n \mathbf{x}_n + v_n$  la solución  $\hat{\mathbf{x}}_n$  que minimiza el error al cuadrado viene dado por:

$$\hat{\mathbf{x}}_n = (\mathbf{H}_n^T \mathbf{H}_n)^{-1} \mathbf{H}_n^T y_n \quad (7.17)$$

Para obtener un modelo lineal a partir de la medición, reescribimos la distancia euclidiana entre el nodo y el ancla, luego obtenemos:

$$\begin{aligned}
 R &= x^2 + y^2 \\
 \hat{\mathbf{x}}_n &= [x \quad y \quad R]^T \\
 \mathbf{H}_n &= \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_1 & -2y_1 & 1 \\ \vdots & \vdots & \vdots \\ -2x_n & -2y_n & 1 \end{bmatrix} \\
 \mathbf{y}_n &= \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 \\ d_2^2 - x_2^2 - y_2^2 \\ \vdots \\ d_n^2 - x_n^2 - y_n^2 \end{bmatrix}
 \end{aligned}$$

### Filtro de Kalman Extendido

Considere el sistema no lineal definido por las ecuaciones 11,14. Supongamos que tenemos un estado inicial  $x_0$  con media y covarianza conocidas. Los ruidos  $w_n$  y  $v_n$  no están correlacionados temporalmente. En el paso inicial, hicimos una predicción de estado usando:

$$\begin{aligned}
 \hat{\mathbf{x}}_n^- &= \mathbf{f}_n(\hat{\mathbf{x}}_{n-1}, \mathbf{u}_n, 0) \\
 \mathbf{P}_n^- &= \mathbf{F}_n \mathbf{P}_{n-1} \mathbf{F}_n^T + \mathbf{W}_n \mathbf{Q}_n \mathbf{W}_n^T
 \end{aligned}$$

El paso de corrección está dado por

$$\begin{aligned}
 \mathbf{K}_n &= \mathbf{P}_n^- \mathbf{H}_n^T (\mathbf{H}_n \mathbf{P}_n^- \mathbf{H}_n^T + \mathbf{V}_n \mathbf{R}_n \mathbf{V}_n^T)^{-1} \\
 \hat{\mathbf{x}}_n &= \hat{\mathbf{x}}_n^- + \mathbf{K}_n (y_n - h(\hat{\mathbf{x}}_n^-, 0)) \\
 \mathbf{P}_n &= (\mathbf{I} - \mathbf{K}_n \mathbf{H}_n) \mathbf{P}_n^-
 \end{aligned}$$

Donde  $\mathbf{F}_n$  es el jacobiano del modelo de sistema evaluado en el punto de estimación anterior  $\hat{\mathbf{x}}_{n-1}$  y  $\mathbf{H}_n$  es el jacobiano del modelo de observación evaluado en el estado predicho  $\hat{\mathbf{x}}_n^-$ .  $\mathbf{Q}_n$  es la covarianza del modelo de ruido y  $\mathbf{R}_n$  es la covarianza del ruido de medición. En nuestro caso,  $\mathbf{W}_n$  y  $\mathbf{V}_n$  son la matriz identidad  $\mathbf{I}$ .

### Filtro FIR Extendido Óptimo

El filtro EOFIR es el estimador FIR óptimo más general (Algoritmo 1). Calculamos estimaciones óptimas utilizando recursiones de Kalman. La estimación será

limitada y el filtro EOFIR es BIBO estable para sistemas estables. El filtro EOFIR tiene mejor respuesta en presencia de perturbaciones debido a la longitud del horizonte  $[m, k]$ . El filtro EOFIR requiere una longitud de horizonte de  $N$ , esto se puede determinar

---

**Algoritmo 4:** Algoritmo EOFIR

---

**Entrada:**  $y_k, \hat{x}_m, P_m, Q_k, R_k, N$

**Salida:**  $\hat{x}_k$

**para**  $k = 1, 2 \dots$  **hacer**

**si**  $k > N - 1$  **entonces**

$m = k - N + 1$

**en otro caso**

$m = 0$

**para**  $l = m + 1 : k$  **hacer**

$\bar{x}_l^- = f_l(\bar{x}_{l-1})$

$P_l^- = F_l P_{l-1} F_l^T + Q_l$

$S_l = H_l P_l^- H_l^T + R_l$

$K_l = P_l^- H_l^T S_l^{-1}$

$\bar{x}_l = \bar{x}_l^- + K_l (y_l - h_l(\bar{x}_l^-))$

$P_l = (I - K_l H_l) P_l^-$

$\hat{x}_k = \bar{x}_k$

---

experimentalmente. Cuanto mayor sea el valor de  $N$ , más lento será el cálculo de la estimación. El filtro EOFIR tiene un formato por lotes [129], que se ejecuta más rápidamente en computación paralela. El valor de  $N$  que minimiza el RMSE se conoce como  $N_{opt}$ .

# Capítulo 8

## Resultados

### 8.1. Predictor H2-FIR

En esta sección describiremos los resultados del predictor  $H_2$ -OUFIR usando el siguiente modelo de seguimiento de dos estados.

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{B}\mathbf{w}_k, \quad (8.1)$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (8.2)$$

Donde las matrices se asignan como sigue

$$\mathbf{F} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \tau/2 \\ 1 \end{bmatrix}, \quad \mathbf{H} = [1 \ 0], \quad (8.3)$$

Para las matrices anteriores,  $\tau$  es un incremento de tiempo,  $w_k$  es una perturbación escalar en la velocidad y  $v_k$  es un error escalar de medición.

1) Simulaciones: La perturbación  $w_k$  es simulada para  $\tau = 0.5s$ , como un ruido de medición estacionario de Markov-Gauss se usó  $w_k = \phi w_{k-1} + \zeta_k$ , donde el factor de coloración  $\phi$  se escogió en el rango de  $0 \leq \phi < 1$  y  $\zeta_k \approx \mathcal{N}(0, \sigma_\zeta^2)$  es un ruido blanco gaussiano con desviación estándar  $\sigma_\zeta = 0.3 m/s$ . El ruido de medición escalar  $v_k \approx \mathcal{N}(0, \sigma_v^2)$  en una desviación estándar de 10m. Se generaron los datos para el proceso de seguimiento y con ellos se alimentaron los predictores  $H_2$ -OUFIR, UFIR y Kalman. Para realizar una comparativa se tomaron todos los estimadores bajo las condiciones de mejor sintonizado con valores iniciales conocidos y una estimación de  $N_{opt}$  y  $\mathcal{Q}_N$ . Los errores de estimación producidos por los algoritmos se encuentran en la tabla, los menores errores están remarcados en negritas.

Es posible notar que el predictor de Kalman produce el menor error para  $\phi = 0$ , pero la diferencia con el resto de los algoritmos es insignificante  $\mathcal{Q}_N$ . Por el

**Tabla 8.1.** *RMSEs bajo perturbaciones de Markov-Gauss.*

$N_{opt}, \phi$	RMSE [m]		
	$H_2$ -OUFIR	UFIR	KF
26,0	4.0992	4.1802	<b>4.0482</b>
10,0.95	<b>6.4410</b>	6.5657	9.1782

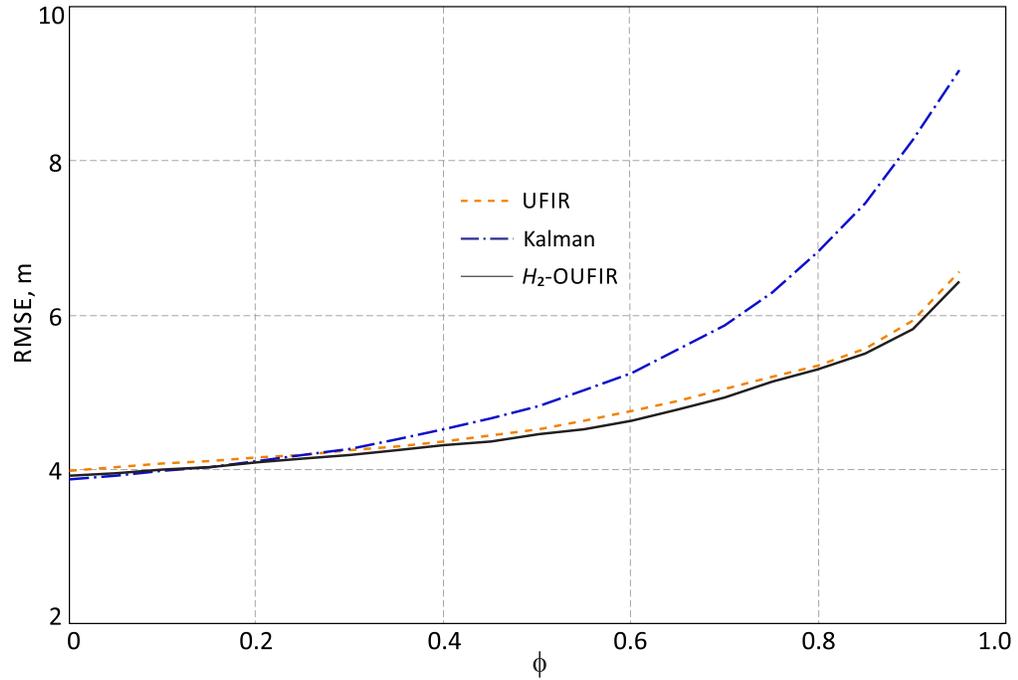
contrario, cuando  $\phi = 0.95$ , las componentes cruzadas de  $\mathcal{Q}_N$  no pueden ser descartadas, el algoritmo de Kalman produce un error al menos 30 % mas grande que el predictor  $H_2$ -OUFIR predictor. Para soportar este resultado, a continuación se presenta un ejemplo de la matriz  $\mathcal{Q}_N$  medida para  $N = 5$  y  $\phi = 0.95$ ,

$$\mathcal{Q}_N = \begin{bmatrix} 1.0882 & 1.0425 & 0.9990 & 0.9575 & 0.9169 \\ 1.0425 & 1.0895 & 1.0436 & 1.0001 & 0.9584 \\ 0.9995 & 1.0436 & 1.0904 & 1.0446 & 1.0009 \\ 0.9575 & 1.0001 & 1.0446 & 1.0914 & 1.0453 \\ 0.9169 & 0.9584 & 1.0009 & 1.0453 & 0.9584 \end{bmatrix} \quad (8.4)$$

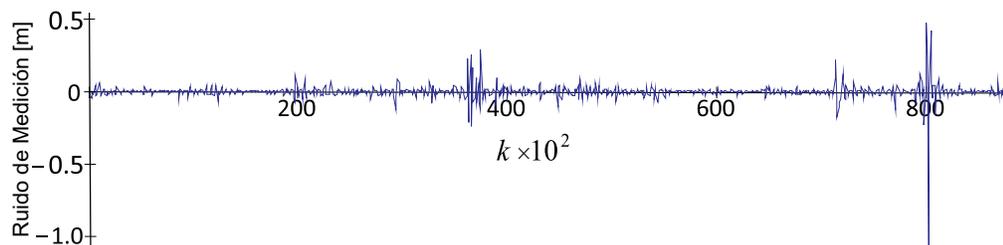
Como puede notar, la magnitud de todas las componentes es de orden similar y por tanto las covarianza entre las componentes del estimador no puede ignorarse sin introducir un error. El incremento en el error de predicción producido por los estimadores al aumentar  $\psi$  se muestra en la figura. Note que un efecto similar es observado cuando se usa un ruido de proceso ( $\zeta_k$ ) con distribución de Cauchy y una cola que no está exponencialmente limitada. Como puede notarse el predictor UFIR tiene un rendimiento similar al  $H_2$ -OUFIR. Esto es debido a que la estructura UFIR no requiere ninguna información estadística del proceso siendo así mas robusto por diseño.

2) Resultados Experimentales: Para validar el rendimiento del predictor  $H_2$ -OUFIR se presenta el análisis realizado a una serie de datos obtenidos con un sistema de observación de Banda Ultraancha (UWB) [130], usando cuatro nodos como referencia y un periodo de muestreo de  $\tau = 0.5s$ . Para minimizar el horizonte de observación se empleó un adelgazamiento de datos en un factor de  $r = 10$ . Se obtuvo una pseudo-referencia usando un suavizador [131] FIR de primer orden con una ventana de 6 puntos. Posteriormente se estimo el ruido de medición como la diferencia entre el valor original y la pseudo-referencia, como se muestra en la figura 8.2 y calculamos la matriz de error de observación  $\mathcal{R}_N$ .

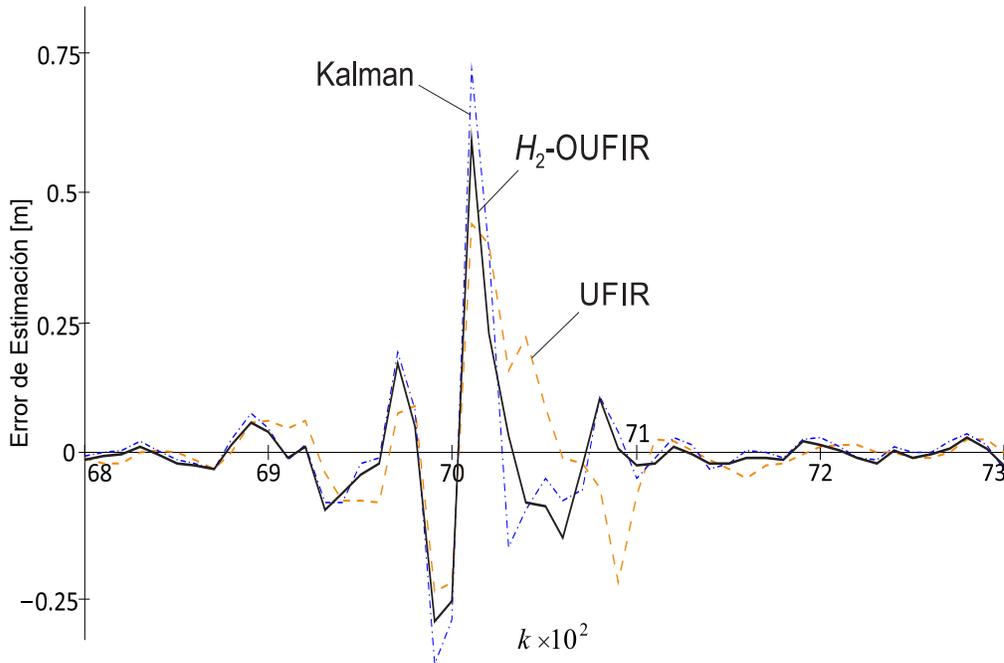
$$\mathcal{R}_N \times 10^3 = \begin{bmatrix} 3.295 & -0.406 & \dots & 0.005 & -0.318 \\ -0.406 & 3.294 & \dots & 0.216 & 0.004 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0.005 & -0.216 & \dots & 3.287 & -0.406 \\ -0.318 & -0.004 & \dots & -0.404 & 3.285 \end{bmatrix} \quad (8.5)$$



**Figura 8.1.** RMSEs producidos por los predictores  $H_2$ -OUFIR, UFIR y Kalman como función de  $\phi$  bajo perturbaciones de Markov-Gauss.



**Figura 8.2.** Ruido de medición no-Gaussiano en la localización por UWB para un robot móvil.



**Figura 8.3.** Errores de estimación producidos por los algoritmos  $H_2$ -OUFIR, UFIR y Kalman

El robot se mueve con una velocidad de  $0.4m/s$ , asumimos  $\sigma_w = 0.1 m/s$  y  $Q_N = \text{diag}(\sigma_w^2 \ \sigma_w^2 \ \dots \ \sigma_w^2)$ . Usamos estos datos con los predictores y se determina un  $N_{opt} = 7$  para el predictor UFIR, fijamos  $N = 10$  para el predictor  $H_2$ -OUFIR. A partir de la matriz de error de observación podemos inferir que los elementos en la diagonal son esencialmente predominantes y por tanto no encontraremos diferencias significativas contra al filtro de Kalman. Los errores de estimación se muestran en la figura 8.3, podemos concluir que, en efecto las predicciones de los estimadores Kalman y  $H_2$ -OUFIR son muy cercanas. Los menores RMSE son  $0.0439m$  para el Kalman,  $0.0468$  para  $H_2$ -OUFIR y  $0.0474m$  para UFIR.

## 8.2. Estimador H2-OFIR

En esta sección se probará el desempeño del estimador  $H_2$ -OFIR a través del seguimiento de un vehículo empleando GPS, se comparará el resultado contra el producido por los filtros de Kalman y UFIR.

### 8.2.1. Seguimiento entre vehículos

Usaremos un ejemplo práctico de seguimiento [52, 132], en este escenario dos vehículos equipados con navegadores GPS se mueven juntos como se indica en la figura 8.4. Adicionalmente, el frente del vehículo esta equipado con un radar de 24

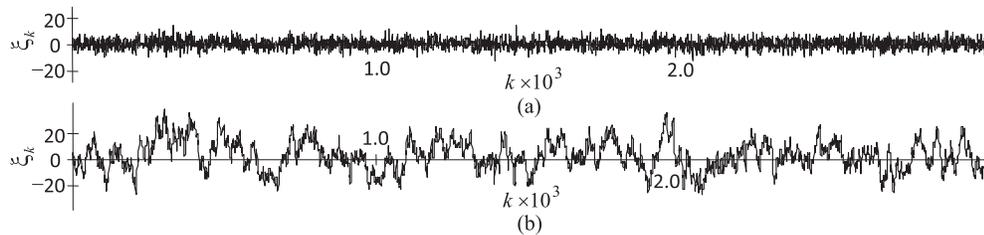


**Figura 8.4.** Seguimiento de un vehículo frontal por un vehículo trasero

GHz de corto alcance (1 a 50m) SRR-20X y que puede ser empleado para aplicaciones industriales, científicas y médicas. Para el vehículo en la delantera el navegador GPS produce una señal de referencia (Ground Truth) de la localización en coordenadas planas  $x, y$ . El radar mide el ángulo y la distancia con el vehículo al frente por motivos de seguridad. Ambas mediciones son combinadas con los datos del GPS para calcular la posición del vehículo al frente, estas serán empleadas como los datos de medición. Debido a que el ruido en la señal del radar no es blanco [51] y que los datos son afectados por el movimiento de ambos vehículos para probar el filtro  $H_2$ -OFIR los datos provenientes del GPS fueron contaminados usando ruido de Markov-Gauss  $\xi_k = \varphi \xi_{k-1} + v_k$  con un factor de coloración  $0 < \varphi < 1$  y un ruido de proceso blanco gaussiano (WGN)  $v_k$ . Este ruido es desconocido por el estimador, ejemplos de un ruido casi blanco  $\varphi = 0.05$  y un ruido muy coloreado  $\varphi = 0.95$  se muestran en la figura 8.5.

Se comprobaron los siguientes filtros

- Filtro  $H_2$ -OFIR por lotes operando en  $[m, k]$  con matrices de error completas  $\mathcal{Q}_N$  y  $\mathcal{R}_N$ .
- Filtro  $H_2$ -OFIR recursivo acorde al algoritmo 1 desarrollado para matrices de



**Figura 8.5.** Ruido de medición de radar  $\xi_k$ ,  $m$ . (a) Casi WGN con  $\varphi = 0.05$ . (b) Fuertemente coloreado con  $\varphi = 0.95$ .

error diagonales  $\mathcal{Q}_N$  y  $\mathcal{R}_N$ .

- Filtro UFIR iterativo operando en un horizonte óptimo  $[m, k]$  de  $N_{opt}$  puntos.
- Filtro Kalman Estándar

Para investigar el tradeoff en la precisión calcularemos la respuesta del filtro al impulso y escalón unitarios, así como a los errores producidos al maniobrar los vehículos de forma lenta y rápida.

### Ajuste del filtro

Para calcular del rendimiento del filtro  $H_2$ -OFIR primero representamos la dinámica del vehículo en las direcciones  $x$  y  $y$  con un modelo de seguimiento de dos coordenadas.

$$\begin{aligned} x_k &= \mathbf{F}x_{k-1} + \mathbf{B}w_k \\ y_k &= \mathbf{H}x_k + \xi_k \\ \xi_k &= \varphi\xi_{k-1} + v_k \end{aligned}$$

Donde  $\xi_k$  es un ruido desconocido que es modelado a través de un proceso de Gauss-Markov. Las mediciones son obtenidas a intervalos fijos de tiempo de forma que  $\tau = t_k - t_{k-1}$ , las matrices de transición de estado, de entrada y de medición están dadas por:

$$F = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \frac{\tau}{2} \\ 1 \end{bmatrix}, H = [1 \quad 0] .$$

La velocidad promedio del vehículo ronda los  $36 \text{ km/h}$  ó  $10 \text{ m/s}$ . Para validar la robustez del filtro maximizamos la desviación estándar del error de perturbación como  $\sigma_w = 4 \text{ m/s}$  lo cual produce una  $Q = 16 (\text{m/s})^2$ , este valor se empleará en todos los filtros. La matriz de error  $\mathcal{R}_N$  es maximizada de forma experimental. Debido a que esta matriz afecta significativamente la salida del filtro, presentaremos dos ejemplos medidos para  $\varphi = 0.05$  y  $\varphi = 0.95$  con  $N = 5$

$$\mathcal{R}_N^{(0.05)} = \begin{bmatrix} 18.23 & 0.526 & 0.118 & 0.297 & -0.072 \\ 0.526 & 18.24 & 0.529 & 0.122 & 0.302 \\ 0.118 & 0.529 & 18.24 & 0.532 & 0.129 \\ 0.297 & 0.122 & 0.532 & 18.24 & 0.540 \\ -0.072 & 0.302 & 0.129 & 0.540 & 18.25 \end{bmatrix}$$

$$\mathcal{R}_N^{(0.95)} = \begin{bmatrix} 178.2 & 168.8 & 160.3 & 152.2 & 144.2 \\ 168.8 & 178.2 & 168.9 & 160.4 & 152.3 \\ 160.3 & 168.9 & 178.2 & 168.9 & 160.5 \\ 152.2 & 160.4 & 168.9 & 178.3 & 169.0 \\ 144.2 & 152.2 & 160.5 & 169.0 & 178.4 \end{bmatrix}.$$

Como se puede apreciar, despreciar las componentes fuera de la diagonal principal puede conducir a errores de estimación muy grandes cuando  $\varphi = 0.95$ , que es el caso del algoritmo 1 y el filtro de Kalman, para el cual fijaremos  $R$  como el primer elemento de la matriz de error  $\mathcal{R}_N$ .

### Respuesta al Impulso y al Escalón

Las respuestas al impulso y al escalón son características generales del filtro que determinan la precisión y la robustez. Para obtener esas funciones utilizaremos los cambios abruptos de velocidad, es decir los puntos en los que la velocidad del vehículo aumenta y posteriormente regresa a un valor en un tiempo anterior. Tomaremos ese comportamiento como un impulso unitario, experimentalmente se determinó un  $N_{opt} = 5$ . La respuesta al escalón y al impulso se muestran en las figuras 8.6 y 8.7. Analizando ambas figuras podemos concluir que el filtro  $H_2$ -OFIR en lotes tiene un rendimiento superior a los filtros de Kalman y al UFIR. Entre las ventajas del filtro  $H_2$ -OFIR podemos destacar las siguientes. 1) Cuando el ruido es casi blanco,  $\varphi = 0.05$ , el filtro responde al escalón unitario con el menor sobre-pico, tiene la misma longitud del estado transitorio del filtro UFIR [vea Fig. 8.6(a)]. 2) Ofrece el mejor seguimiento para ruidos fuertemente coloreados,  $\varphi = 0.95$  [vea Fig. 8.6(b)].

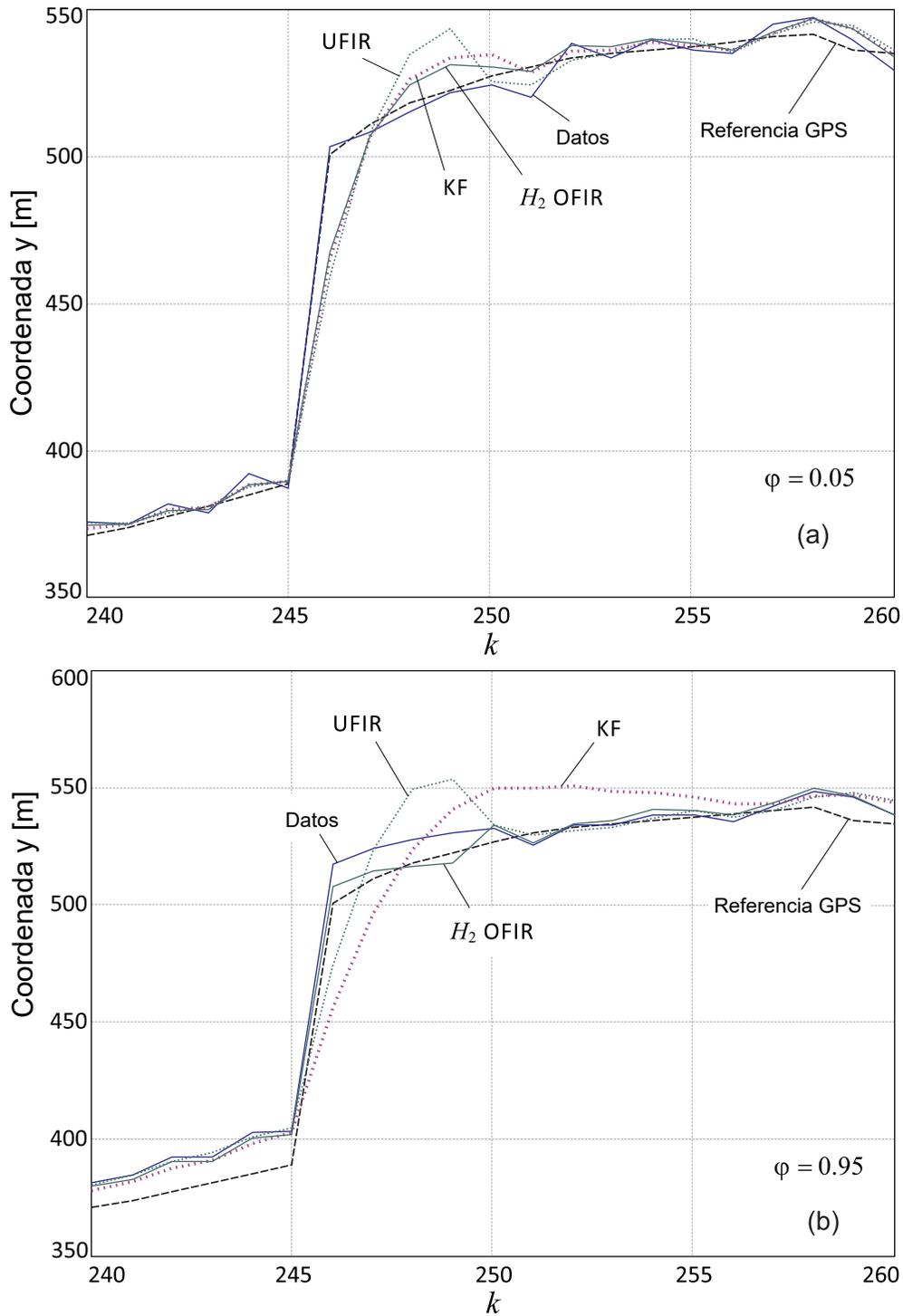
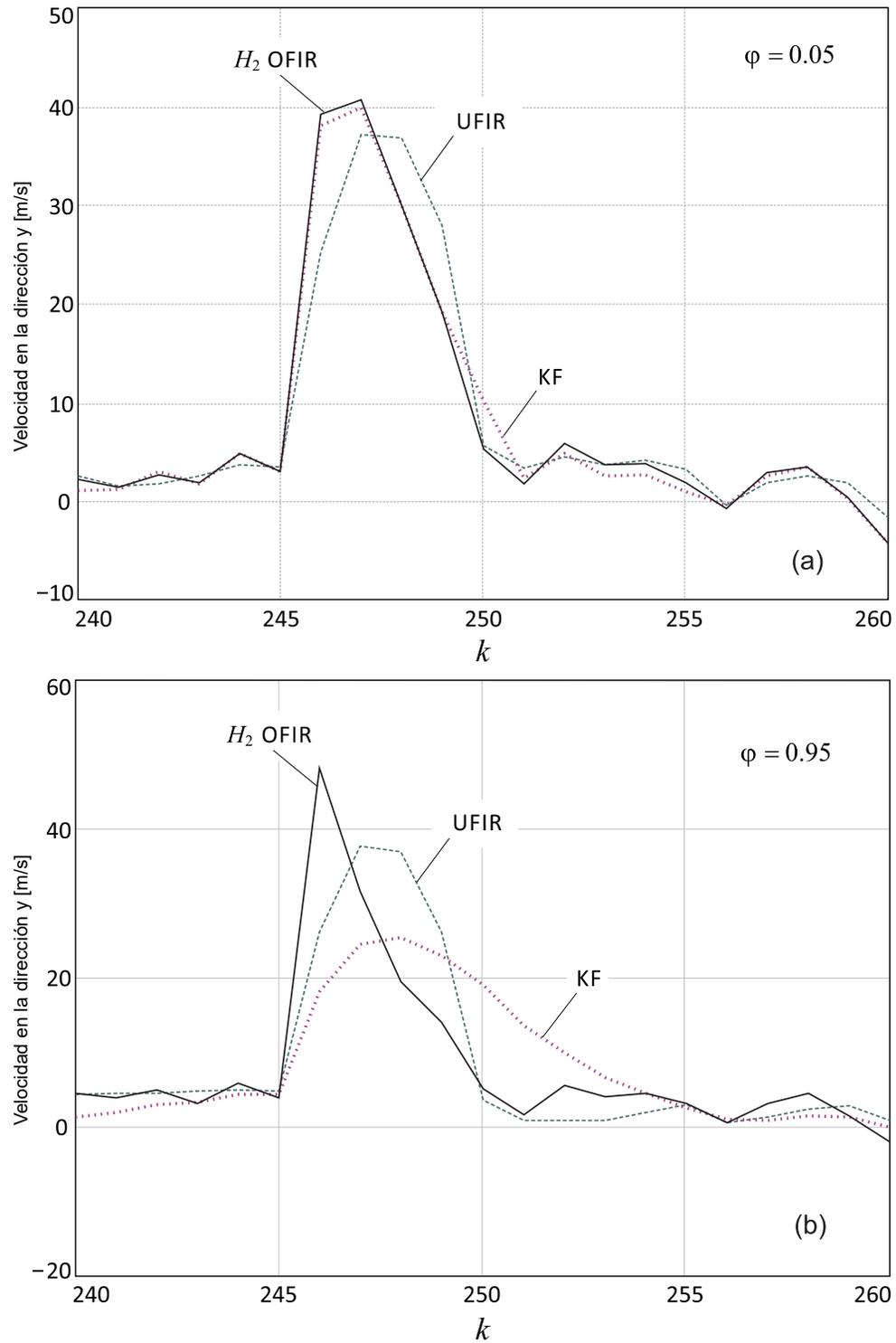


Figura 8.6. Respuesta al escalón de los filtros en la coordenada y para dos valores extremos de factor de color (a)  $\varphi = 0.05$ . (b)  $\varphi = 0.95$ .

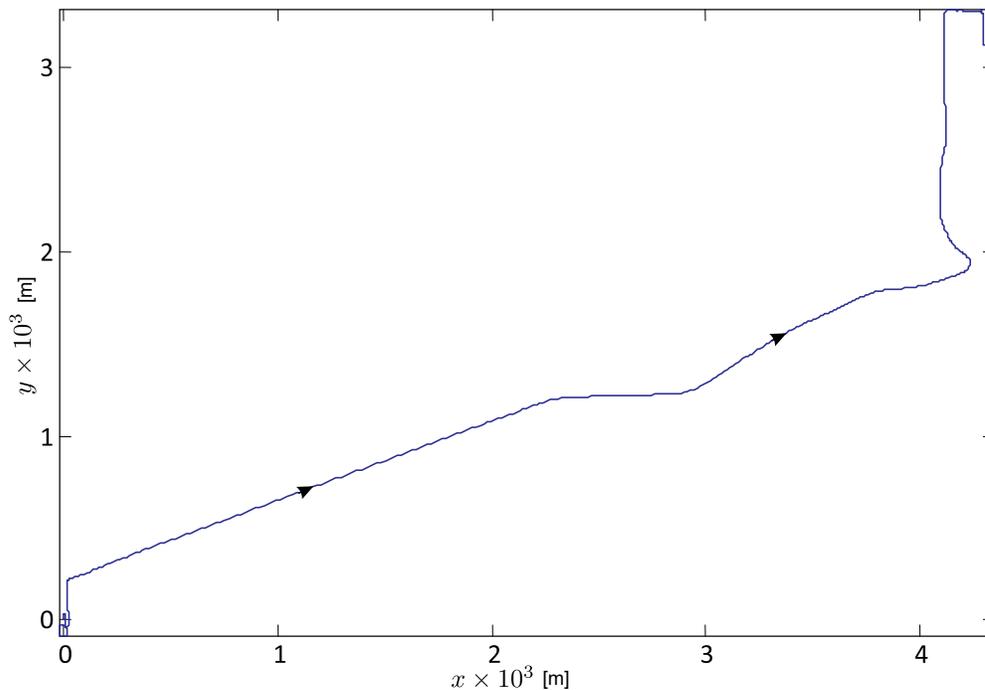


**Figura 8.7.** Respuesta al impulso de los filtros en la coordenada y para dos valores extremos de factor de color (a)  $\varphi = 0.05$ . (b)  $\varphi = 0.95$ .

Se puede concluir que el filtro  $H_2$ -OFIR es el mejor estimador, su rendimiento es mejor que el filtro UFIR que ignora el ruido y las condiciones iniciales y que el filtro de Kalman que requiere la matriz diagonal  $R_N$ .

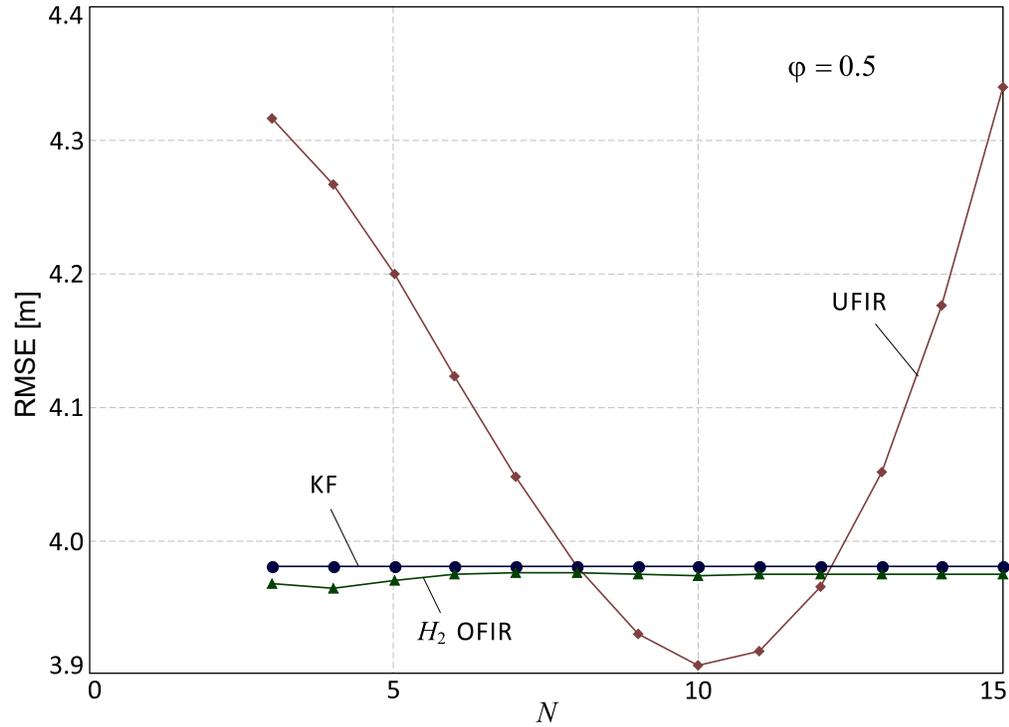
### Seguimiento de un vehículo con maniobra suave

Consideremos ahora el caso de una maniobra suave, en la figura 8.8 podemos ver la posición estimada mediante los datos del GPS. Examinaremos los errores a lo largo de la dirección x. El objetivo es aprender el efecto de la longitud del horizonte  $N$  en el error del estimador FIR.



**Figura 8.8.** Trayectoria medida por el GPS para un vehículo realizando una maniobra suave

El RMSE producido por los filtros como función de  $N$  bajo ruido medianamente coloreado  $\varphi = 0.5$  está mostrado en la figura 8.9. Este resultado fue promediado sobre 50 corridas de una simulación de Monte Carlo (MC).

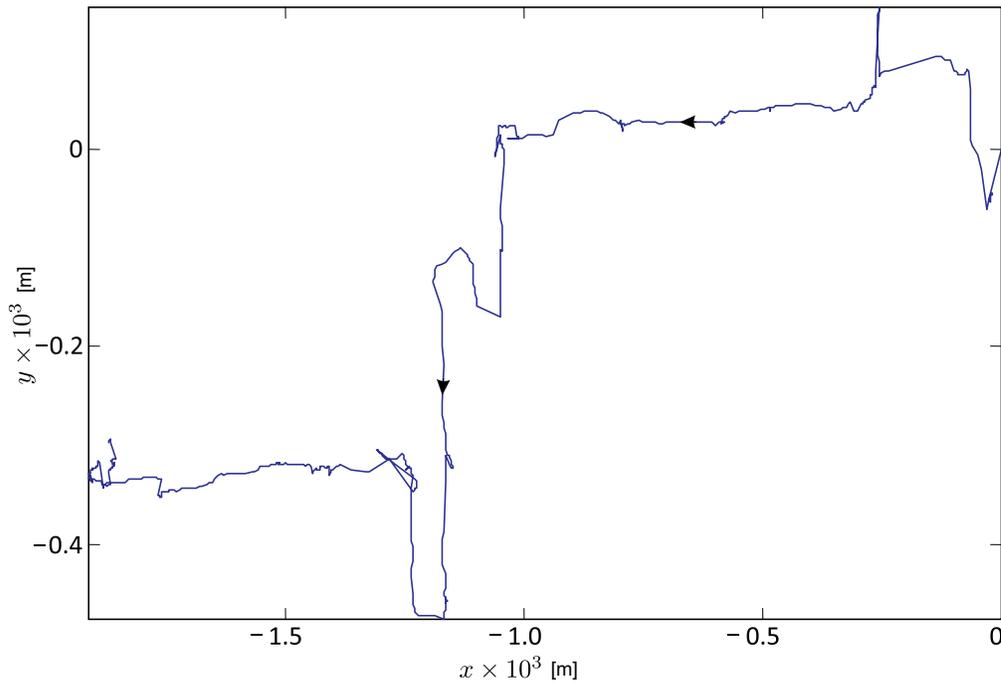


**Figura 8.9.** Valores típicos de RMSE producidos por los filtros  $H_2$ -OFIR por lotes, Kalman, y UFIR a lo largo de la dirección en  $x$  como función de  $N$  para maniobras suaves considerando un vehículo bajo ruido moderadamente coloreado  $\varphi = 0.5$ .

Como podemos ver, el filtro UFIR resulta óptimo para  $N_{opt} = 10$ . El KF tiene un RMSE invariante al valor de  $N$ , para valores  $8 \leq N \leq 12$  el filtro UFIR supera en rendimiento al KF. El filtro  $H_2$ -OFIR por lotes, que opera con la matriz completa  $\mathcal{R}_N$ , supera al KF para todos los valores de  $N$  y produce el menor RMSE para  $N_{opt} = 4$ ; en este valor, el algoritmo resulta más eficiente en tiempos de cómputo. Esto significa que despreciar las componentes fuera de la diagonal principal de  $\mathcal{R}_N$  incrementa los errores de seguimiento. Además, el valor de  $N$  no afecta al filtro  $H_2$ -OFIR de la misma forma que al UFIR, por ello, el filtro  $H_2$ -OFIR puede operar en horizontes cortos haciendo más eficiente computacionalmente.

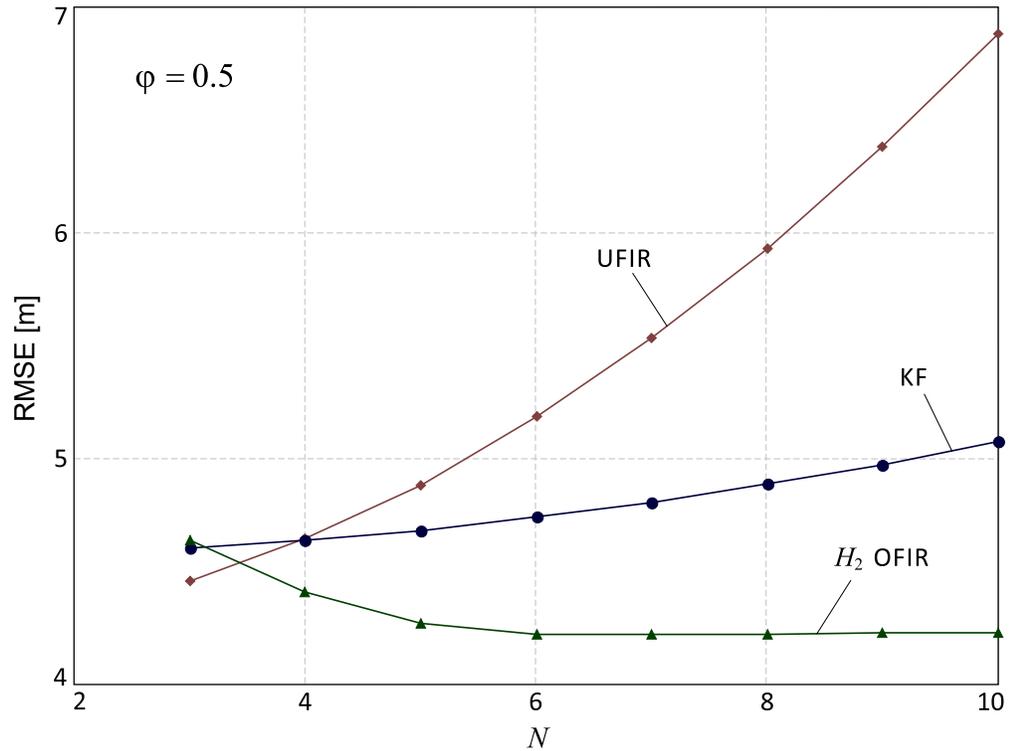
### Seguimiento de un vehículo con maniobra rápida

Finalmente, se considerará una trayectoria con maniobras rápidas, la ruta medida por el GPS puede ser vista en la figura 8.10.



**Figura 8.10.** *Trajectory measured by GPS for a vehicle performing rapid maneuvers*

El RMSE promediado sobre 50 corridas de MC esta mostrado en la figura 8.11 como una función de  $N$ .



**Figura 8.11.** Valores típicos de RMSEs producidos por los filtros  $H_2$ -OFIR por lotes, Kalman, y UFIR a lo largo de la dirección en  $x$  como función de  $N$  para maniobras rápidas considerando un vehículo bajo ruido moderadamente coloreado  $\varphi = 0.5$ .

El filtro UFIR requiere una longitud del horizonte menor  $N = 3$ , en este caso tiene un mejor rendimiento que el KF y el  $H_2$ -OFIR. Sin embargo, un incremento en  $N$  debido a errores de sintonizado puede causar que el filtro UFIR tienda a diverger. Además es posible ver que el filtro  $H_2$ -OFIR tiene un mejor desempeño que el resto para  $N > 3$  lo que habla en favor de sus capacidades de seguimiento.

### Complejidad computacional

El filtro  $H_2$ -OFIR por lotes tiene una complejidad  $O(N^2)$  que puede resultar en computacionalmente ineficiente cuando  $N \gg 1$ . El filtro UFIR iterativo tiene una menor complejidad  $O(N)$ , y el filtro Kalman tiene la menor complejidad resultando el más rápido. En base a los experimentos de seguimiento podemos observar que el filtro  $H_2$ -OFIR opera en horizontes cortos con mayor precisión (Fig. 8.9 y Fig. 8.11). Esto implica que, a diferencia de los tiempos en que Kalman era el mejor estimador, incluso con computadoras lentas y con poca memoria, las arquitecturas FIR por lotes tienen gran potencial para incrementar la precisión en tareas de seguimiento.

### 8.3. AutoLocalización en un entorno con RFID-tags

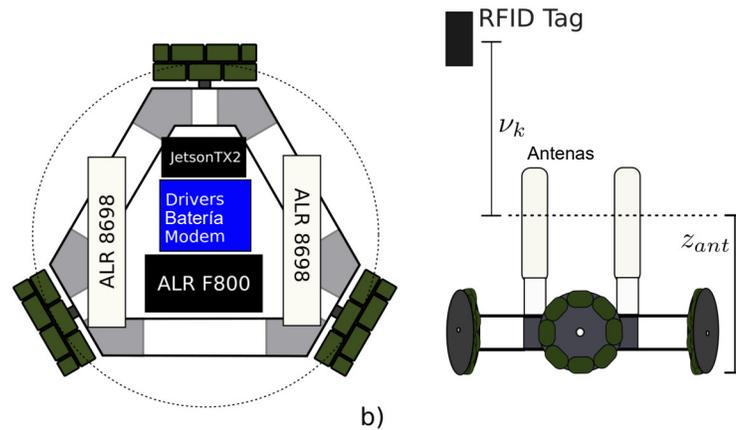
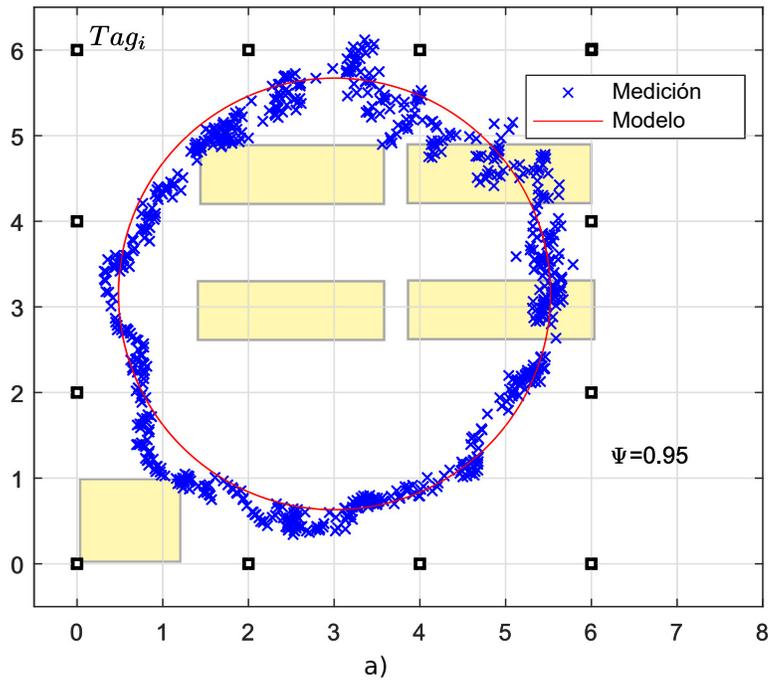
Ahora consideramos dos escenarios de autolocalización TWOR en entornos de etiquetas RFID utilizando el filtro cEKF y cEFIR junto con los filtros EKF y EFIR como puntos de referencia. Cuando los datos de una etiqueta se pierden durante la transmisión, estimamos la observación perdida a partir del modelo de observación utilizando la estimación anterior y establece su covarianza como un valor "grande". En consecuencia, el filtro "ignora" la medida. De esta forma siempre tenemos la misma longitud en el vector de observación. Para probar la robustez de los estimadores frente a CMN, establecemos dos condiciones de optimización  $\Psi_l = 0.05$  y  $\Psi_h = 0.95$ . En el caso del algoritmo cEKF, fijamos el parámetro  $\Psi$  en el algoritmo. Para los filtros FIR debemos elegir una longitud de filtro  $N$  para cada condición de optimización  $N_{opt}|\Psi=0.05$  y  $N_{opt}|\Psi=0.95$ . Para optimizar el tamaño del filtro, calculamos el RMSE para diferentes valores de  $N$ , luego seleccionamos el valor pequeño como  $N_{opt}$ . Finalmente, probamos filtros FIR para valores fijos de  $(N_{opt}, \Psi)$ . Probaremos todos los estimadores para CMN en el rango de  $\Psi \in [0.05, 0.95]$ . Nuestro punto de referencia será el RMSE de los algoritmos estándar EKF y EFIR y el filtro de partículas (PF), que pueden proporcionar una buena estimación del problema del filtrado no lineal.

Los efectos de trayectorias múltiples pueden afectar una medición de RF cuando no hay una trayectoria directa entre la etiqueta y la antena; esta condición se conoce como "sin línea de visión"(NLOS). NLOS reduce la potencia efectiva de la señal transmitida, lo que lleva a un valor sesgado de la intensidad de la señal recibida (RSSI). Estos fenómenos son la principal fuente de error de seguimiento en nuestro experimento.

Se presentarán dos experimentos. En el primer caso, el TWOR recorrerá de una trayectoria circular en un espacio interior con un desplazamiento del motor constante y ruido de medición. En el segundo caso, probamos todos los filtros con los datos experimentales disponibles en [133].

#### 8.3.1. Trayectoria circular en un ambiente interior

La red de etiquetas RFID y la trayectoria circular TWOR planificada en un espacio de navegación interior se muestran en la Fig. 8.12 junto con datos corrompidos por CMN con  $\Psi = 0, 95$ .



**Figura 8.12.** a) Ambiente interior con etiquetas RFID y trayectoria de prueba del TWOR. b) TWOR y los dispositivos asociados

Se utilizaron 12 etiquetas RFID UHF de largo alcance (Chip Alien H3, Protocolos ISO18000-6B/C, Dimensiones: 78 x 30 x 8 mm). Todas las etiquetas están ubicadas a una distancia de 2 m entre sí en una habitación cuadrada de (8 x 8)m. Cada etiqueta tiene un número de identificación único y coordenadas de ubicación conocidas exactamente. El interrogador ALR-F800 (Alien Technology) interactúa con las etiquetas dentro del alcance del lector, recopila datos y los transmite a una computadora host. La velocidad de interrogación del lector es de 1000 etiquetas por segundo, pero empleamos sólo 12 etiquetas. Para mitigar el efecto de la antena RFID en el rendimiento del sistema, utilizamos una antena con polarización circular Alien ALR-8698. Creamos

**Tabla 8.2.** *RMSEs (m) Producido por los filtros FIR para  $N_{\text{opt}} = 152; 400$*

Filter	$\Psi = 0.05$		$\Psi = 0.5$		$\Psi = 0.95$	
	152	400	152	400	152	400
EFIR	0.805	1.048	1.222	1.327	8.739	5.985
cEFIR	0.805	<b>1.047</b>	<b>1.170</b>	<b>1.323</b>	<b>5.504</b>	<b>4.941</b>

el vector de observación utilizando la información de intensidad de la señal recibida (RSSI) del lector. Durante una etapa fuera de línea, medimos algunos valores RSSI para estimar los parámetros constantes del modelo de propagación log-normal. Establecimos el coeficiente de pérdida de energía para oficinas en  $n = 33$ . (según lo recomendado por el Sector de Radiocomunicaciones de la Unión Internacional de Telecomunicaciones UIT-R P.1238-7)

Las ecuaciones en espacio de estados son (6.9) y (6.11), donde el número de etiquetas es  $n = 12$  y el CMN se modela con las ecuaciones (6.15) y (6.19). Para aplicar filtros, utilizamos las ecuaciones extendidas, (6.13) y (6.14). El CMN se representa con un modelo de Gauss-Markov (6.12), donde el factor de coloración  $\Psi_n$  tiene media cero y la matriz de covarianza diagonal  $R_n = \text{diag}(\sigma_{v_1}^2 \dots \sigma_{v_n}^2 \sigma_{\Phi}^2)$ , en el cual  $\sigma_{v_i} = 20$  cm,  $i \in [1, n]$  y  $\sigma_{\Phi} = \pi/360$  rad. La matriz de covarianza diagonal del ruido del sistema  $Q_n = \text{diag}(\sigma_{w_1}^2 \sigma_{w_2}^2 \sigma_{w_3}^2)$  tiene  $\sigma_{w_i} = 10$  cm,  $\sigma_{v_i} = 20$  cm y  $\sigma_{\Phi} = \pi/360$  rad.

Para reducir los errores de autolocalización bajo CMN y probar la robustez de los estimadores, a continuación realizamos investigaciones estableciendo diferentes valores del factor  $\Psi_n$ . Para cubrir posibles casos observados en datos reales, cambiamos  $\Psi$  de 0,05 a 0,95 con un paso de 0,05 y consideramos las siguientes opciones.

### Sintonizado del Filtro para $\Psi_n = 0.05$

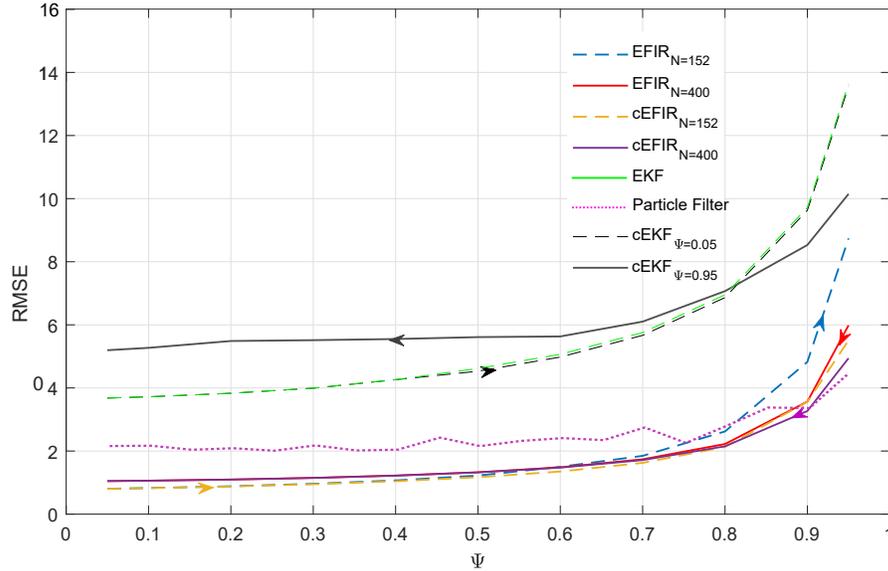
Para el filtro cEFIR, medimos  $N_{\text{opt}} = 152$  y aplicamos este valor a todos los demás valores de  $\Psi_n$ . Luego ejecutamos la simulación Monte Carlo (MC) 100 veces y mostramos los errores cuadráticos medios producidos por los filtros en la Fig. 8.13. Lo que podemos ver es que los filtros EFIR producen las estimaciones más pequeñas y consistentes cuando  $\Psi < 0,5$  y que el cEKF ajustado a  $\Psi = 0,05$  tiene un error ligeramente mayor. Además, no se observa ninguna diferencia significativa entre las estimaciones de EFIR cuando  $\Psi < 0,5$ . Por el contrario, bajo ruido fuertemente coloreado con  $\Psi = 0,95$ , todos los filtros producen errores de estimación grandes, sin embargo, los filtros FIR tienen mayor precisión. Para los filtros FIR, los RMSE producidos para  $\Psi = 0.05$ ,  $\Psi = 0.5$  y  $\Psi = 0.95$  se enumeran en la Tabla 8.2 y, para el filtro cEKF, en la Tabla 8.3, donde los errores más pequeños producidos por el filtro cEFIR están resaltados.

**Tabla 8.3.** RMSEs (m) Producido por el algoritmo cEKF para  $\Psi_{opt} = 0.05; 0.95$

Filter	$\Psi = 0.05$	$\Psi = 0.5$	$\Psi = 0.95$
cEKF( $\Psi_{opt} = 0.05$ )	3.679	4.529	13.551
cEKF( $\Psi_{opt} = 0.95$ )	5.195	5.610	10.148

### Sintonizado del Filtro para $\Psi_n = 0.95$

En el modo robusto, ajustamos todos los filtros al factor máximo  $\Psi = 0,95$ , lo que da  $N_{opt} = 400$  para el filtro UFIR. Aunque no hay diferencias significativas entre los RMSE cuando  $\Psi < 0,5$ , el filtro cEFIR es ligeramente más preciso. Bajo el ruido ligeramente coloreado ( $\Psi = 0.05, N_{opt} = 152$ ), los filtros FIR (EFIR y cEFIR) funcionan mejor que los filtros Kalman, produciendo menor error. Como era de esperar, el filtro cEFIR calcula el mejor resultado. Vale la pena señalar que los errores de filtrado crecen con un aumento de  $\Psi$ . Aun así, el filtro cEFIR tiene la tasa más baja y produce el RMSE más pequeño en el modo extremo de  $\Psi = 0.95$  y  $N_{opt} = 400$ .

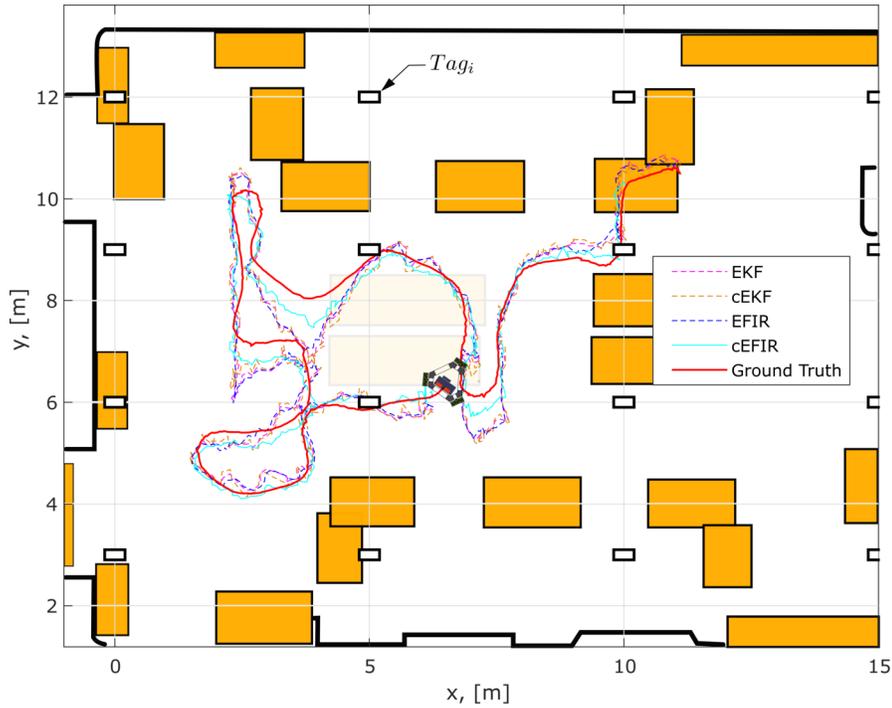


**Figura 8.13.** RMSE producido por los algoritmos cEKF, EFIR y cEFIR

### 8.3.2. Simulación con datos experimentales

Para validar los resultados mediante simulaciones, ahora consideramos la trayectoria experimental del robot móvil disponible en [133]. El vector de medición se crea para una malla de 16 etiquetas en una cuadrícula cuadrada de 5 m, del conjunto de datos tomamos los campos *RFID\_global\_robot* para calcular la distancia a cada

etiqueta. Usamos el mismo modelo de espacio de estados. Para probar la robustez de los filtros, consideramos diferentes valores de  $\Psi$  en dos opciones de ajuste.



**Figura 8.14.** Trayectoria estimada del robot y medición de referencia.

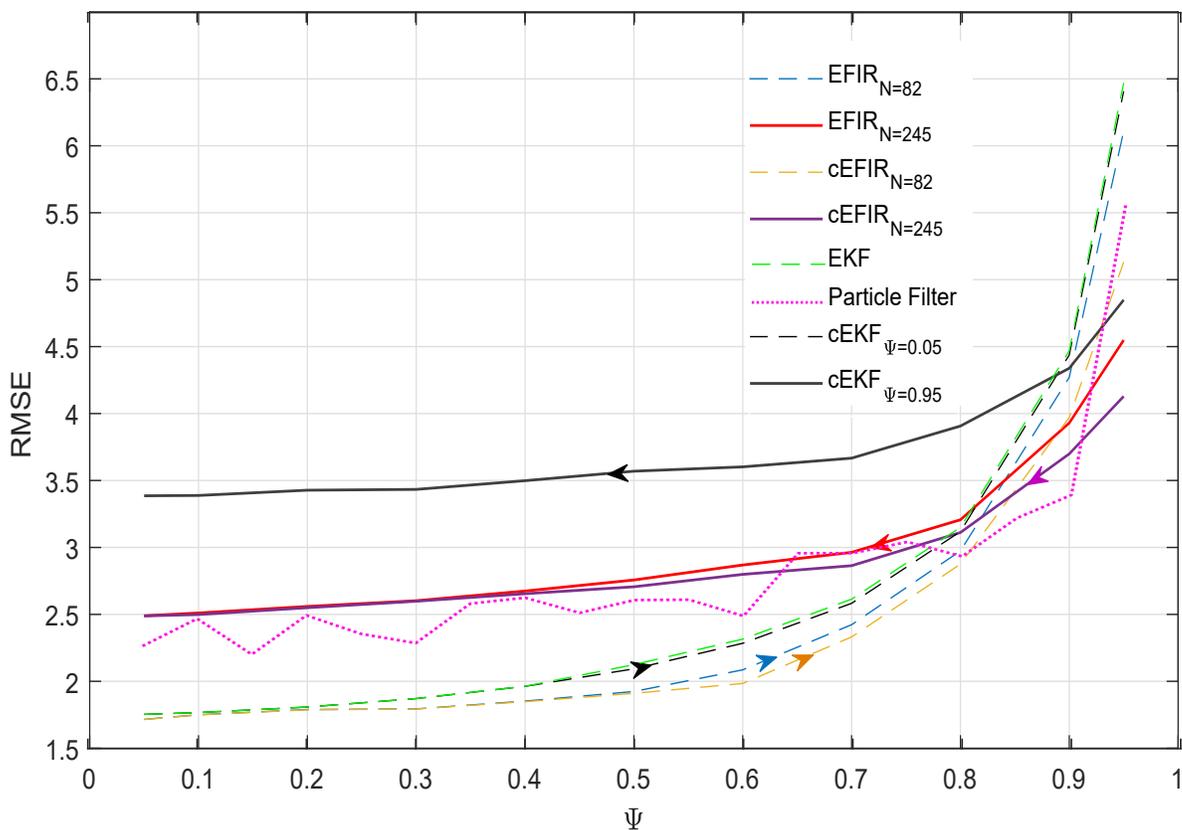
El espacio de trabajo simulado para TWOR se encuentra en la Fig.8.14, se muestran la ubicación, las posiciones de la etiqueta y la medición de referencia (Ground Truth). Además, la figura muestra los resultados del estimador para la peor condición de ruido  $\Psi_n = 0.95$ . Los RMSE se enumeran en la Tabla 8.4 para los filtros FIR y en la Tabla 8.5 para el filtro cEKF. Los RMSE producidos por los filtros en la segunda prueba se muestran en la figura 8.15.

**Tabla 8.4.** RMSEs (m) Producido por los filtros FIR para  $N_{\text{opt}} = 82; 245$

Filter	$\Psi = 0.05$		$\Psi = 0.5$		$\Psi = 0.95$	
	82	245	82	245	82	245
EFIR	1.715	2.489	1.923	2.756	6.110	4.548
cEFIR	<b>1.712</b>	<b>2.488</b>	<b>1.910</b>	<b>2.706</b>	<b>5.131</b>	<b>4.128</b>

**Tabla 8.5.** RMSEs (m) Producido por el algoritmo cEKF para  $\Psi_{opt} = 0.05; 0.95$

Filter	$\Psi = 0.05$	$\Psi = 0.5$	$\Psi = 0.95$
cEKF( $\Psi_{opt} = 0.05$ )	1.753	2.093	6.407
cEKF( $\Psi_{opt} = 0.95$ )	3.385	3.569	4.848



**Figura 8.15.** RMSE producido por los algoritmos cEKF, EFIR y cEFIR

En general, llegamos a la siguiente observación crítica: *ajustado a  $\Psi_n = 0.05$  con  $N_{opt} = 82$  o a  $\Psi_n = 0.95$  con  $N_{opt} = 245$ , el filtro cEFIR proporciona los errores mínimos que se ajustan a las simulaciones.*

### 8.3.3. Consumo de tiempo y Complejidad Computacional

Aquí medimos el consumo de tiempo y la complejidad computacional de los algoritmos de estimación propuestos y también presentamos una comparación con EKF, EFIR y el filtro de partículas. En nuestros algoritmos, el cuello de botella radica en el cálculo de la covarianza residual  $S_n$  (cEKF), y la matriz de ganancia  $G_n$  (cEFIR), la

complejidad es  $O(n^3)$  para ambos casos, en términos de Punto de operación flotante (FLOP) la complejidad. Mostramos la eficiencia en términos de FLOPS en la Tabla 8.6. Donde  $K$  es el número de estados del sistema,  $M$  es el número de mediciones y  $N_{opt}$  es el número de iteraciones del filtro FIR, hemos omitido los términos de orden inferior.

**Tabla 8.6.** Número de FLOPs para los algoritmos *cEKF* y *cEFIR*

Algoritmo	Función de transición de estado	Función de Medición	Algoritmo iterativo
<b>cEKF</b>	1	1	$9K^3 + 10K^2M + 5MK^2 + M^2 + 4M^3$
<b>cEFIR</b>	$K - 1 + N_{opt}$	$N_{opt}$	$N_{opt}(4K^3 + 4M^2K + 2K^2 + 2M^3)$

Para probar el consumo de tiempo, ejecutamos los algoritmos con un entorno simulado, con 10 etiquetas y movimiento en línea recta. Las etiquetas se distribuyeron uniformemente a 0,5 m a lo largo del camino y se separaron del suelo a 1 m. Luego, calculamos una trayectoria con 500 puntos y alimentamos los algoritmos con los datos. Los algoritmos FIR y el algoritmo de Filtro de Partículas (PFA) tienen un parámetro extra, que aumenta el tiempo de cálculo, para hacer una buena comparación establecemos los siguientes criterios:

- Correr los algoritmos para  $\Psi_n = 0$ , debido a que bajo esta condición el RMSE del estimador propuesto y de los métodos estándar son iguales.
- Para el filtro FIR, escoger como longitud del filtro  $N_{opt}$  aquella que genere un RMSE similar al del filtro EKF.
- Correr el filtro de partículas con  $N_{opt}$  como el número de partículas.

De la figura 3 se desprende claramente que los algoritmos propuestos necesitan más potencia computacional. Para el filtro Kalman, el tiempo de cálculo aumentó de

0,108 s (EKF) a 0,2848 s (cEKF), y para el filtro FIR, el tiempo fue de 3,0521 s (EFIR) a 5,1906 s (cEFIR). El tiempo medio del algoritmo del Filtro de Partículas fue de 6,7301s. La siguiente sección presenta el error de seguimiento de los algoritmos propuestos, mostramos el error del algoritmo del filtro de partículas como referencia. No fue posible comparar el RMSE entre algoritmos en condiciones similares porque, para obtener un valor uniforme para el RMSE, el PFA debe ejecutarse en una simulación de Monte Carlo con 620 rondas con 500 partículas. PFA tiene un pequeño error de seguimiento en nuestros experimentos, lo que muestra una gran solidez frente a CMN pero la peor eficiencia temporal.

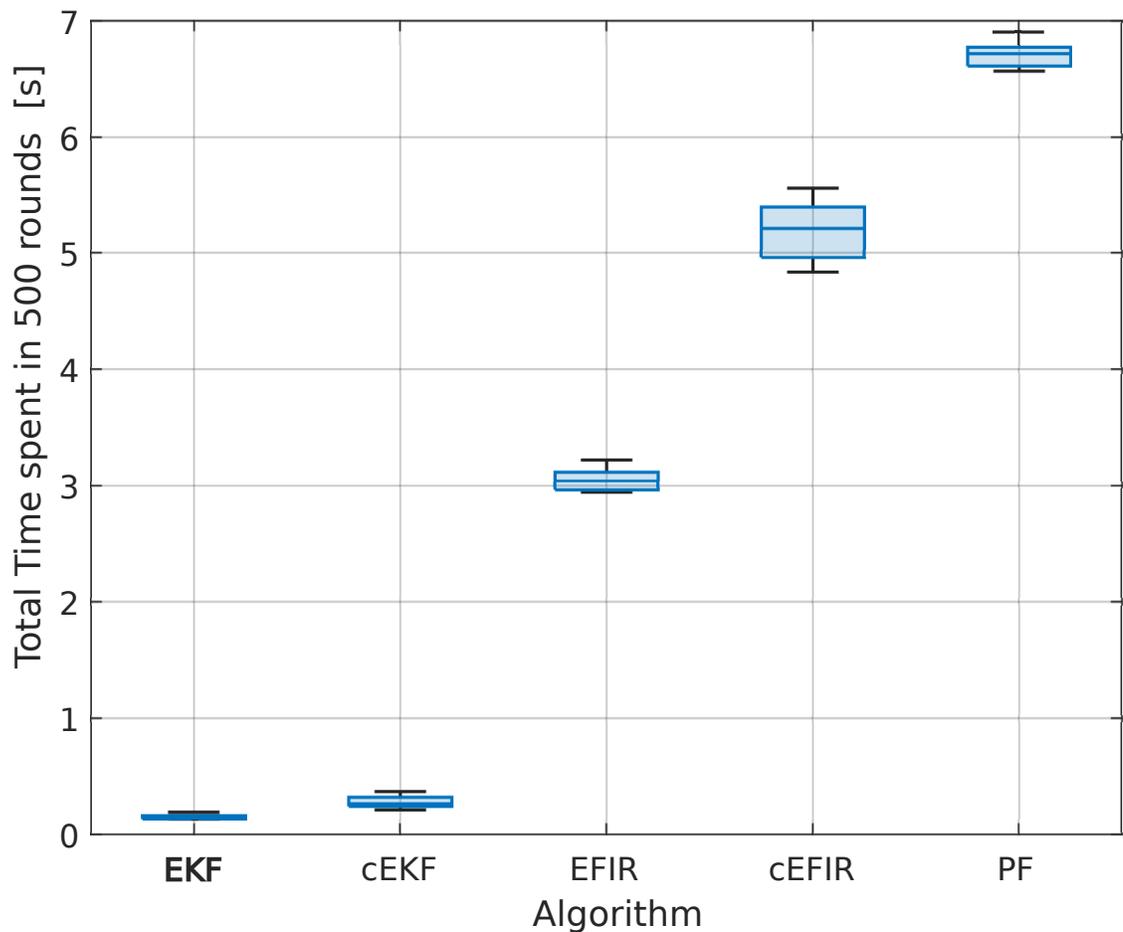


Figura 8.16. Consumo en tiempo de los algoritmos

## 8.4. Localización del TWOR en una red UWB

Para el entorno de prueba, desplegamos 7 anclajes en posiciones fijas en una sala de 6x5m aproximadamente. No utilizamos ninguna disposición particular para las etiquetas. Fueron colocados arbitrariamente. El robot dispone de una brújula digital para orientarse. En nuestro experimento, el robot siempre sigue una orientación tangente a la trayectoria. La precisión de la brújula es  $\pm 4$  grados. El camino descrito por el TWOR es circular. La estimación de localización se encuentra en la figura 8.17.

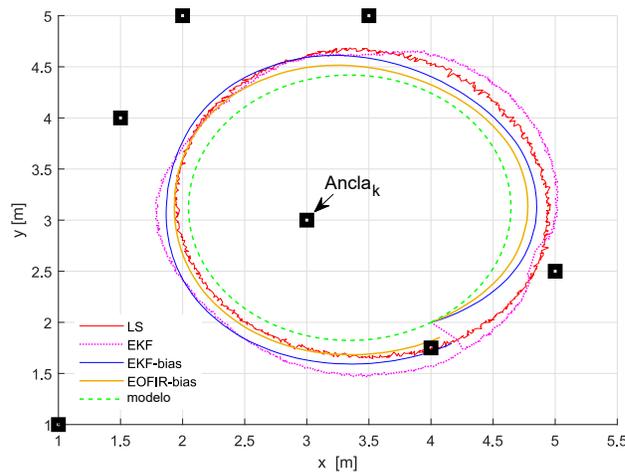


Figura 8.17. Posición estimada del TWOR

Los errores de estimación se pueden ver en la figura 8.18. Es relevante destacar lo siguiente:

- En rojo podemos observar el resultado del método no probabilístico. Incluso sin conocimiento del modelo, el estimador LS puede recuperar la trayectoria pero con un gran sesgo. El RMSE para este estimador fue de 0,1572 m.
- En magenta, se utilizó el algoritmo EKF clásico asumiendo que no había errores debido al desvanecimiento. La estimación fue mejor que en el estimador LS. El RMSE era 0,0124
- En azul, el filtro EKF con corrección de sesgo, las mejoras con respecto a los algoritmos de localización anteriores son claras, el RMSE es 0,0091.
- Finalmente, en naranja, el algoritmo EOFIR con  $N_{opt} = 8$ , este es el mejor resultado con un RMSE de 0.0065.

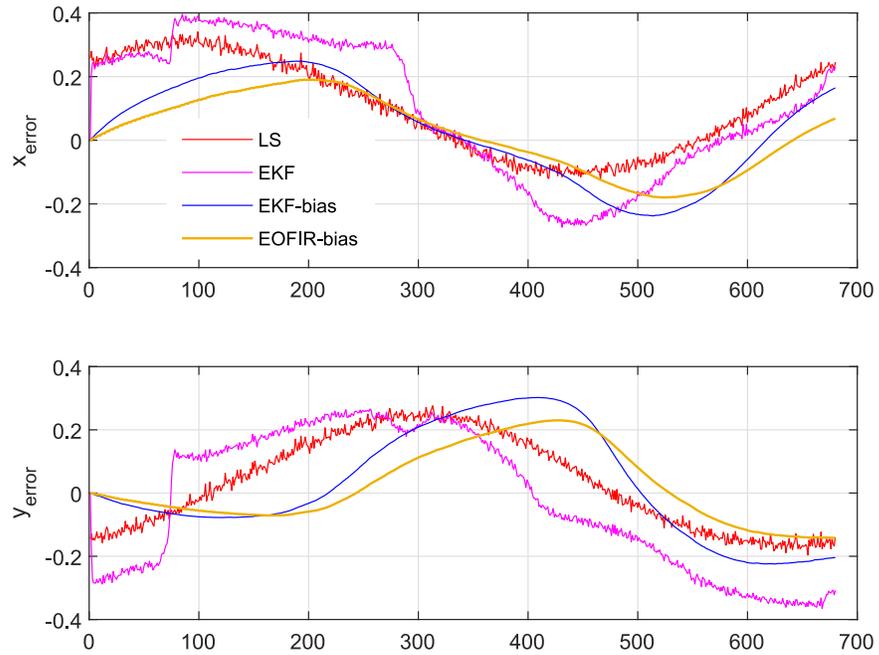


Figura 8.18. Error obtenido con los distintos algoritmos de localización

# Capítulo 9

## Conclusiones

El predictor H2-OUFIR derivado bajo perturbaciones y errores de medición ha demostrado un mejor rendimiento que el predictor de Kalman debido a la capacidad de operar con matrices de error completas. Esto ha sido posible gracias a la introducción de la norma ponderada de Frobenius que esencialmente simplificó el procedimiento de derivación. Se demostró que el predictor H2-OUFIR tiene la forma de predictor ML FIR y es equivalente al predictor OUFIR en un caso especial de procesos gaussianos. Además, el predictor H2 FIR subóptimo se ha derivado para estructuras numéricas con robustez mejorada, como el predictor  $H_\infty/H_2$

El filtro a posteriori  $H_2$ -OFIR desarrollado en este artículo ha demostrado un mejor rendimiento incluso para coloraciones moderadas que el filtro KF y UFIR en términos de precisión y robustez. Además, su capacidad para proporcionar seguimiento en horizontes cortos significa que la complejidad computacional no es un problema, especialmente para las computadoras modernas. Esto ha sido posible mediante el uso del enfoque mejorado de estimación del estado  $H_2$ -FIR y el desarrollo de la robusta estructura de filtro por lotes  $H_2$ -OFIR para matrices completas de perturbaciones y errores. Tenga en cuenta que el filtro OFIR y KF requieren que estas matrices sean gaussianas blancas y, por lo tanto, diagonales. En cuanto al mejor rendimiento de seguimiento del filtro  $H_2$ -OFIR.

Los algoritmos de filtrado extendido desarrollados en esta tesis han producido errores aceptables para la autocalización de TWOR en entornos interiores de etiquetas RFID bajo CMN. Esto ha sido posible mejorando el modelo TWOR utilizando interpretación geométrica para perfiles de velocidad que varían suavemente y representación de tiempo incremental en tiempo discreto. Los resultados obtenidos mediante simulaciones y experimentalmente han revelado que bajo CMN el filtro cEFIR supera al cEKF, al filtro EFIR y al EKF en términos de precisión y robustez. El PF demuestra una robustez un poco mejor, pero este filtro es menos preciso que el filtro

---

cEFIR en la localización en tiempo real. También se deduce que el filtro cEFIR tiene la mayor robustez en condiciones de funcionamiento duras con CMN pesado. Ahora modificamos el modelo dinámico TWOR para la representación del espacio de estados lineal para utilizar los enfoques FIR y Kalman directamente.

En el desarrollo de la prueba ambiental para rastrear un TWOR en una red de sensores UWB se determinaron de forma experimental los efectos del desvanecimiento en condiciones LOS y proponemos una corrección de sesgo al modelo de observación. Finalmente, realizamos una comparación entre algoritmos de posicionamiento considerando mediciones imparciales/sesgadas. Cabe destacar que el filtro EOFIR tiene mejores resultados que el filtro EKF, pero en teoría ambos filtros son equivalentes. En un escenario real, cada ancla se encuentra en diferentes condiciones (LOS, NLOS y multitrayecto), y el filtro EOFIR tiene más robustez contra perturbaciones que el filtro EKF. El primero siempre estuvo acotado y, por tanto, es inherentemente estable. El segundo se derivó en forma IIR, por lo que tiene un rechazo de perturbación lento.

# Bibliografía

- [1] C. L. Sang, M. Adams, T. Hörmann, M. Hesse, M. Porrmann, and U. Rückert, “Numerical and experimental evaluation of error estimation for two-way ranging methods,” *Sensors*, vol. 19, p. 616, Feb. 2019.
- [2] X. Nan and X. Xiao-wen, “Robot experiment simulation and design based on Festo Robotino,” in *2011 IEEE 3rd Int. Conf. Commun. Software Networks*, pp. 160–162, 2011, Xi’an, China.
- [3] H. Mallinson, S. Hodges, and A. Thorne, “Determining a better metric for rfid performance in environments with varying noise levels,” in *12th IEEE International Conference on Methods and Models in Automation and Robotics*, IEEE, January 2006.
- [4] S. Pagano, S. Peirani, and M. Valle, “Indoor ranging and localisation algorithm based on received signal strength indicator using statistic parameters for wireless sensor networks,” *IET Wireless Sensor Systems*, vol. 5, pp. 243–249, Oct. 2015.
- [5] M. A. Khan, “Tracking methodologies in rfid network,” in *Radio Frequency Identification Fundamentals and Applications* (C. Turcu, ed.), ch. 10, Rijeka: IntechOpen, 2010.
- [6] Y. Shmaliy, S. Khan, O. Ibarra-Manzano, and S. Zhao, “Factors affecting self-localization in uhf rfid tag networks,” 02 2016.
- [7] R. Aliberti, E. Di Giampaolo, and G. Marrocco, “A model to estimate the rfid read-region in real environments,” in *2008 European Conference on Wireless Technology*, pp. 290–293, 2008.
- [8] A. Yaro, “Position estimation performance evaluation of a linear lateration algorithm with an SNR-based reference station selection technique,” *Nigerian Journal of Technology*, vol. 37, p. 484, July 2018.
- [9] H.-K. Ryu, G. Jung, D.-K. Ju, S. Lim, and J.-M. Woo, “An electrically small spherical uhf rfid tag antenna with quasi-isotropic patterns for wireless sensor

- 
- networks,” *IEEE Antennas and Wireless Propagation Letters*, vol. 9, pp. 60–62, 2010.
- [10] R. Wang, R. Xie, T. Wang, D. Chen, T. Song, L. Kuang, and S. Zhu, “Design of semi-active rfid antenna,” in *2017 Progress In Electromagnetics Research Symposium - Spring (PIERS)*, pp. 199–202, 2017.
- [11] M. Boccadoro, F. Martinelli, and S. Pagnottelli, “Constrained and quantized kalman filtering for an RFID robot localization problem,” *Autonomous Robots*, vol. 29, pp. 235–251, June 2010.
- [12] J. J. Pomárico-Franquiz and Y. S. Shmaliy, “Accurate self-localization in rfid tag information grids using fir filtering,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1317–1326, 2014.
- [13] J. Pomarico-Franquiz, Y. Shmaliy, and S. Zhao, “Self-localization in rfid-tag information networks using extended ufir filtering,” 07 2014.
- [14] M. Granados-Cruz, J. Pomárico-Franquiz, Y. S. Shmaliy, and L. J. Morales-Mendoza, “Triangulation-based indoor robot localization using extended fir/kalman filtering,” in *2014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–5, 2014.
- [15] O. G. Ibarra-Manzano, F. Ramirez-Echeverria, and Y. S. Shmaliy, “Extended ufir filtering of nonlinear models corrupted by white gaussian noise,” in *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 345–348, 2012.
- [16] X. Sun, J. Duan, X. Li, and X. Wang, “State estimation under non-gaussian levy noise: A modified kalman filtering method,” 2013.
- [17] R. Haendel, “Design of an omnidirectional universal mobile platform,” *Journal of Educational Psychology - J EDUC PSYCHOL*, 01 2005.
- [18] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, “State-space solutions to standard  $H_2$  and  $H_\infty$  control problems,” in *1988 American Control Conference*, pp. 1691–1696, 1988.
- [19] B. Hassibi, A. H. Sayed, and T. Kailath, *Indefinite-Quadratic Estimation and Control*. Society for Industrial and Applied Mathematics, 1999.
- [20] I. Kammerer, P. Khargonekar, and M. Rotea, “Mixed  $H_2/H_\infty$  control for discrete-time systems via convex optimization,” pp. 1363 – 1367, 07 1992.
- [21] M. Souza, A. Fioravanti, and J. Geromel, “ $H_2$  sampled-data filtering of linear systems,” *IEEE Transactions on Signal Processing*, vol. 62, pp. 4839–4846, 09 2014.

- 
- [22] Z. Tan, Y. C. Soh, and L. Xie, “Envelope-constrained  $H_2$  FIR filter design,” *Circuits, Systems, and Signal Processing*, vol. 18, pp. 539–551, Nov. 1999.
- [23] B.-S. Chen and J.-C. Hung, “Fixed-order  $H_2$  and  $H_\infty$  optimal deconvolution filter designs,” *Signal Processing*, vol. 80, pp. 311–331, Feb. 2000.
- [24] S. Wang, L. Xie, and C. Zhang, “ $H_2$  optimal inverse of periodic fir digital filters,” *IEEE Transactions on Signal Processing*, vol. 48, no. 9, pp. 2696–2700, 2000.
- [25] S. Wang, L. Xie, and C. Zhang, “Mixed  $H_2/H_\infty$  deconvolution of uncertain periodic fir channels,” *Signal Process.*, vol. 81, no. 10, pp. 2089–2103, 2001.
- [26] D. M. Wiberg, “Optimal and robust estimation with an introduction to stochastic control theory,” *IEEE Control Systems Magazine*, vol. 30, no. 4, pp. 103–106, 2010.
- [27] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Feher/Prentice Hall Digital and, Prentice Hall, 1996.
- [28] E. Gershon, U. Shaked, and I. Yaesh,  *$H_\infty$  Control and Estimation of State-multiplicative Linear Systems*. Lecture Notes in Control and Information Sciences, Springer London, 2005.
- [29] X. Lihua, Y. C. Soh, D. Chunling, and Y. Zou, “Robust  $h_2$  estimation and control,” *Journal of Control Theory and Applications*, vol. 2, pp. 20–26, Feb. 2004.
- [30] X. Lihua, “On robust  $H_2$  estimation,” *ACTA AUTOMATICA SINICA*, vol. 31, no. 1, pp. 1–12, 2005.
- [31] D.-Q. Wang and F. Ding, “Hierarchical least squares estimation algorithm for hammerstein-wiener systems,” *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 825–828, 2012.
- [32] M. Gan, X.-X. Chen, F. Ding, G.-Y. Chen, and C. L. P. Chen, “Adaptive rbf-ar models based on multi-innovation least squares method,” *IEEE Signal Processing Letters*, vol. 26, no. 8, pp. 1182–1186, 2019.
- [33] Y. Zhou and F. Ding, “Modeling nonlinear processes using the radial basis function-based state-dependent autoregressive models,” *IEEE Signal Processing Letters*, vol. 27, pp. 1600–1604, 2020.
- [34] Y. Wang, F. Ding, and L. Xu, “Some new results of designing an IIR filter with colored noise for signal processing,” *Digital Signal Processing*, vol. 72, pp. 44–58, Jan. 2018.
- [35] W.-H. Kwon and S. Han, *Receding Horizon Control: Model Predictive Control for State Models*. 01 2005.

- 
- [36] Y. S. Shmaliy, “Linear optimal fir estimation of discrete time-invariant state-space models,” *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3086–3096, 2010.
- [37] Y. Lee, S. Han, and W. Kwon, “ $H_2/H_\infty$  fir filters for discrete-time state space models,” vol. 4, pp. 645–652, Oct. 2006.
- [38] C. K. Ahn, Y. S. Shmaliy, S. Zhao, and H. Li, “Continuous-time deadbeat  $H_2$  FIR filter,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, pp. 987–991, Aug. 2017.
- [39] C. K. Ahn, Y. S. Shmaliy, and S. Zhao, “A new unbiased fir filter with improved robustness based on frobenius norm with exponential weight,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 4, pp. 521–525, 2018.
- [40] C. K. Ahn, S. Zhao, Y. S. Shmaliy, and H. Li, “On the  $L_2$ - $L_\infty$  and  $H_\infty$  performances of continuous-time deadbeat  $H_2$  fir filter,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, pp. 1798–1802, Nov. 2018.
- [41] J. Makhoul, “Linear prediction: A tutorial review,” *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.
- [42] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, “Unbiased finite impulse response filtering: An iterative alternative to kalman filtering ignoring noise and initial conditions,” *IEEE Control Systems Magazine*, vol. 37, no. 5, pp. 70–89, 2017.
- [43] C. Scherer and S. Weiland, *Linear Matrix Inequalities in Control*. Stuttgart: University of Stuttgart, 2015.
- [44] R. J. Caverly and J. R. Forbes, “Lmi properties and applications in systems, stability, and control theory,” 2019.
- [45] A. H. S. T. Kailath and B. Hassibi, *Linear Estimation*. Upper Saddle River, NJ: Prentice Hall, 2000.
- [46] S. Zhao and Y. S. Shmaliy, “Unified maximum likelihood form for bias constrained fir filters,” *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1848–1852, 2016.
- [47] M. A. Khan, *Handbook of research on industrial informatics and manufacturing intelligence: Innovations and solutions: Innovations and solutions*. IGI Global, 2012.
- [48] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.
- [49] L. Gao, G. Battistelli, L. Chisci, and A. Farina, “Fusion-based multidetection multitarget tracking with random finite sets,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2438–2458, 2021.

- 
- [50] D. Cormack and J. R. Hopgood, “Message passing and hierarchical models for simultaneous tracking and registration,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, pp. 1524–1537, June 2021.
- [51] S. Stergiopoulos, ed., *Advanced Signal Processing Handbook*. CRC Press, Dec. 2000.
- [52] S. M. Patole, M. Torlak, D. Wang, and M. Ali, “Automotive radars: A review of signal processing techniques,” *IEEE Signal Processing Magazine*, vol. 34, pp. 22–35, Mar. 2017.
- [53] D. Wilson, “Convolution and hankel operator norms for linear systems,” *IEEE Transactions on Automatic Control*, vol. 34, no. 1, pp. 94–97, 1989.
- [54] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Feher/Prentice Hall Digital and, Prentice Hall, 1996.
- [55] S. Wang, L. Xie, and C. Zhang, “Mixed  $H_2/H_\infty$  deconvolution of uncertain periodic FIR channels,” *Signal Processing*, vol. 81, pp. 2089–2103, Oct. 2001.
- [56] R. M. Palhares and P. L. Peres, “Robust filtering with guaranteed energy-to-peak performance — an LMI approach,” *Automatica*, vol. 36, pp. 851–858, June 2000.
- [57] J. Abedor, K. Nagpal, and K. Poolla, “A linear matrix inequality approach to peak-to-peak gain minimization,” *International Journal of Robust and Nonlinear Control*, vol. 6, pp. 899–927, Nov. 1996.
- [58] D. Simon, *Optimal State Estimation*. John Wiley & Sons, Inc., Jan. 2006.
- [59] A. Cohen and U. Shaked, “Linear discrete-time  $H_\infty$ -optimal tracking with preview,” *IEEE Transactions on Automatic Control*, vol. 42, pp. 270–276, 1997.
- [60] J. Ball, P. Kachroo, and A. Krener, “ $H_\infty$  tracking control for a class of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 44, pp. 1202–1206, June 1999.
- [61] K. M. M. P. L. Rawicz, P. R. Kalata and T. A. Chmielewski, “Explicit formula for two state kalman,  $H_2$  and  $H_\infty$  target tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, pp. 53–69, Jan. 2003.
- [62] I. Yaesh and U. Shaked, “Discrete-time min-max tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, pp. 540–547, Apr. 2006.
- [63] T. Gaspar and P. Oliveira, “Model-based  $\mathcal{H}_2$  adaptive filter for 3d positioning and tracking systems,” *Automatica*, vol. 50, pp. 225–232, Jan. 2014.
- [64] Z. Tan, Y. C. Soh, and L. Xie, “ $H_2$  optimal envelope-constrained FIR filter design: An LMI approach,” *Signal Processing*, vol. 75, pp. 141–149, June 1999.

- 
- [65] C. K. Ahn, S. Zhao, and Y. S. Shmaliy, “Frequency-efficient receding horizon  $H_\infty$ -FIR filtering in discrete-time state-space,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, pp. 2945–2953, Nov. 2017.
- [66] J. M. Pak, P. S. Kim, S. H. You, S. S. Lee, and M. K. Song, “Extended least square unbiased FIR filter for target tracking using the constant velocity motion model,” *International Journal of Control, Automation and Systems*, vol. 15, pp. 947–951, Jan. 2017.
- [67] C. K. Ahn, “A new solution to the induced  $l_\infty$  finite impulse response filtering problem based on two matrix inequalities,” *International Journal of Control*, vol. 87, pp. 404–409, Sept. 2013.
- [68] C. K. Ahn and P. S. Kim, “New energy-to-peak fir filter design for systems with disturbance: a convex optimization approach,” *International Journal of Innovative Computing, Information and Control*, vol. 9, no. 5, pp. 1987–1993, 2013.
- [69] C. K. Ahn, Y. S. Shmaliy, P. Shi, and Y. Zhao, “Receding-horizon  $l_2-l_\infty$  fir filter with embedded deadbeat property,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 2, pp. 211–215, 2016.
- [70] Y. S. Shmaliy and O. Ibarra-Manzano, “Time-variant linear optimal finite impulse response estimator for discrete state-space models,” *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 2, pp. 95–104, 2012.
- [71] J. Ortega-Contreras, E. Pale-Ramon, Y. S. Shmaliy, and Y. Xu, “A novel  $h_2$  approach to fir prediction under disturbances and measurement errors,” *IEEE Signal Processing Letters*, vol. 28, pp. 150–154, 2020.
- [72] Y. S. Shmaliy, Y. Xu, J. A. Andrade-Lucio, and O. Ibarra-Manzano, “Predictive tracking under persistent disturbances and data errors using  $h_2$  fir approach,” *IEEE Transactions on Industrial Electronics*, vol. 69, no. 6, pp. 6121–6129, 2021.
- [73] S. Zhao, Y. S. Shmaliy, and F. Liu, “Optimal fir filter for discrete-time ltv systems and fast iterative algorithm,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 4, pp. 1527–1531, 2021.
- [74] L. Marín, “Modular open hardware omnidirectional platform for mobile robot research,” in *2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA)*, pp. 1–6, 2018.
- [75] D. Betancur-Vásquez, M. Mejia-Herrera, and J. Botero-Valencia, “Open source and open hardware mobile robot for developing applications in education and research,” *HardwareX*, vol. 10, p. e00217, Oct. 2021.

- 
- [76] “Ubuntu install of ros melodic.” <http://wiki.ros.org/melodic/Installation/Ubuntu>, 2020. Accessed: 2023-09-20.
- [77] “Ros wrapper for intel realsense devices.” <https://github.com/leggedrobotics/realsense-ros-rsl>, 2018. Accessed: 2023-09-20.
- [78] P. F. Muir and C. P. Neuman, “Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot,” in *IEEE Int. Conf. Robot. Autom.*, pp. 1772–1778, Raleigh, NC, USA, 1987.
- [79] H.-C. Huang, “SoPC-Based parallel ACO algorithm and its application to optimal motion controller design for intelligent omnidirectional mobile robots,” *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 1828–1835, Nov. 2013.
- [80] P. F. Muir and C. Neuman, “Kinematic modeling of wheeled mobile robots,” *J. Robot. Syst.*, vol. 4, no. 2, pp. 281–340, 1987.
- [81] B. J. Yi and W. K. Kin, “The kinematics for redundantly actuated omnidirectional mobile robots,” *J. Robot. Syst.*, vol. 19, no. 6, pp. 255–267, 2002.
- [82] R. Holmberg, *Design and Development of Powered-caster Holonomic Mobile Robots*. PhD thesis, Stanford University, Aug. 2000.
- [83] P. F. Muir, *Modeling and Control of Wheeled Mobile Robots*. PhD thesis, Carnegie Mellon University, Aug. 1988.
- [84] S. Ostrovskaya, J. Angeles, and R. Spiteri, “Nonholonomic systems revisited within the framework of analytical mechanisms,” *Appl. Mechan. Review*, vol. 51, no. 7, pp. 415–433, 1998.
- [85] J. C. Lins-Barreto, A. G. Scolari Conceição, C. E. T. Dórea, L. Martinez, and E. R. de Pieri, “Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot,” *IEEE/ASME Trans. Mechatron*, vol. 19, pp. 467–476, Apr. 2014.
- [86] A. R. Jimenez-Ruiz and S.-G. F., “Comparing Ubisense, BeSpoon, and DecaWave UWB location systems: Indoor performance analysis,” *IEEE Trans. Instrum. Meas.*, vol. 66, pp. 1–12, Aug. 2017.
- [87] Y. Xu, Y. S. Shmaliy, C. K. Ahn, T. Shen, and Y. Zhuang, “Coupled integration of INS and UWB using fixed-lag extended UFIR smoothing for quadrotor localization,” *IEEE Intern. Things J.*, vol. 8, pp. 1716–1727, Feb. 2021.
- [88] J. Pomarico-Franquiz and Y. S. Shmaliy, “Accurate self-localization in RFID tag information grids using FIR filtering,” *IEEE Trans. Ind. Informat.*, vol. 10, pp. 1317–1326, May 2014.

- 
- [89] X. Liu, J. Zhang, S. Jiang, Y. Yang, K. Li, J. Cao, and J. Liu, “Accurate localization of tagged objects using mobile RFID-augmented robots,” *IEEE Trans. Mobile Comput.*, vol. 20, pp. 1273–1284, Apr. 2021.
- [90] H. Wu, X. Wu, and G. Tian, “Indoor robot localization based on single RFID tag,” *Artificial Life and Robotics*, vol. 23, no. 3, pp. 373–379, 2018.
- [91] H. Chen, X. Ai, K. Lin, N. Yan, Z. Wang, N. Jiang, and J. Yu, “BFVP: A probabilistic UHF RFID tag localization algorithm using Bayesian filter and a variable power RFID model,” *IEEE Trans. Ind. Informat.*, vol. 1, pp. 345–355, Jan. 2022.
- [92] B. Tao, H. Wu, Z. Gong, Z. Yin, and H. Ding, “An RFID-based mobile robot localization method combining phase difference and readability,” *IEEE Trans. Autom. Sci. Eng.*, vol. 18, pp. 1406–1416, Jul. 2020.
- [93] R. Chen, X. Huang, Y. Zhou, Y. Hui, and N. Cheng, “Uhf-rfid-based real-time vehicle localization in gps-less environments,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–8, 2021.
- [94] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, “Unbiased finite impulse response filtering: an iterative alternative to Kalman filtering ignoring noise and initial conditions,” *IEEE Contr. Syst. Mag.*, vol. 37, no. 5, pp. 70–89, 2017.
- [95] Y. S. Shmaliy, “Suboptimal FIR filtering of nonlinear models in additive white Gaussian noise,” *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5519–5527, 2012.
- [96] C. Jing, T. Sun, Q. Chen, M. Du, S. Wang, and J. Wang, “A robust noise mitigation method for the mobile RFID location in built environment,” *Sensors*, vol. 19, no. 9, pp. 1–16, 2019.
- [97] K. E. Sundstrom, P. Dietrich, C. J. Diorio, and A. Friefeld, “RFID readers mitigating colored noise,” Dec. 4 2012. US Patent 8,325,014.
- [98] A. E. Bryson and L. J. Henrikson, “Estimation using sampled data containing sequentially correlated noise,” *J. Spacecraft Rockets*, vol. 5, no. 6, pp. 662–665, 1968.
- [99] M. G. Petovello, K. O’Keefe, G. Lachapelle, and M. E. Cannon, “Consideration of time-correlated errors in a Kalman filter applicable to GNSS,” *Journal of Geodesy*, vol. 83, pp. 51–56, January 2009.
- [100] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, “Kalman and UFIR state estimation with colored measurement noise using backward Euler method,” *IET Signal Process.*, vol. 14, no. 2, pp. 64–71, 2020.

- 
- [101] Z. Zhou, J. Wu, Y. Li, C. Fu, and H. Fourati, “Critical issues on Kalman filter with colored and correlated system noises,” *Asian J. Control*, vol. 19, no. 6, p. 1905–1919, 2017.
- [102] F. Ding, D. Meng, J. Dai, Q. Li, A. Alsaedi, and T. Hayat, “Least squares based iterative parameter estimation algorithm for stochastic dynamical systems with ARMA noise using the model equivalence,” *Int. J. Control, Automat. Syst*, vol. 16, no. 2, pp. 630–639, 2018.
- [103] J. Pomarico-Franquiz, S. Khan, and Y. S. Shmaliy, “Combined extended FIR/Kalman filtering for indoor robot localization via triangulation,” *Measurement*, vol. 50, pp. 236–242, Apr. 2014.
- [104] Y. S. Shmaliy, “An iterative Kalman-like algorithm ignoring noise and initial conditions,” *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2465–2473, 2011.
- [105] F. Ramirez-Echeverria, A. Sarr, and Y. Shmaliy, “Optimal memory for discrete-time FIR filters in state-space,” *IEEE Trans. Signal Process.*, vol. 62, pp. 557–561, Feb. 2014.
- [106] A. S. Kundu, O. Mazumder, P. K. Lenka, and S. Bhaumik, “Omnidirectional assistive wheelchair: Design and control with isometric myoelectric based intention classification,” *Procedia Computer Science*, vol. 105, pp. 68–74, 2017.
- [107] J. Moreno and E. Clotet, “Design, implementation and validation of the three-wheel holonomic motion system of the assistant personal robot,” *Sensors*, vol. 16, no. 10, p. 1658, 2016.
- [108] J. Qian, B. Zi, D. Wang, Y. Ma, and D. Zhang, “The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system,” *Sensors*, vol. 17, p. 2073, Sept. 2017.
- [109] A. S. Rajawat, P. Bedi, S. B. Goyal, P. K. Shukla, A. Zaguia, A. Jain, and M. M. Khan, “Reformist framework for improving human security for mobile robots in industry 4.0,” *Mobile Information Systems*, vol. 2021, pp. 1–10, Oct. 2021.
- [110] G. I. R. K. Galgamuwa, L. K. G. Liyanage, M. P. B. Ekanayake, and B. G. L. T. Samaranyake, “Simplified controller for three wheeled omni directional mobile robot,” in *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*, IEEE, Dec. 2015.
- [111] T. Terakawa, M. Komori, and K. Matsuda, “Motion analysis of an omnidirectional mobile robot with wheels connected by passive sliding joints,” in *Advances in Mechanism and Machine Science*, pp. 2279–2288, Springer International Publishing, 2019.

- 
- [112] L. Gracia and J. Tornero, “A practical approach for motion planning of wheeled mobile robots,” in *Motion Planning*, InTech, June 2008.
- [113] T. Borangiu, ed. Springer International Publishing, 2016.
- [114] M. von Tschirschnitz, M. Wagner, M.-O. Pahl, and G. Carle, “Clock error analysis of common time of flight based positioning methods,” in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, Sept. 2019.
- [115] F. B. Sorbelli, C. M. Pinotti, and G. Rigoni, “Range-free localization algorithms with mobile anchors at different altitudes,” in *Proceedings of the 21st International Conference on Distributed Computing and Networking*, ACM, Jan. 2020.
- [116] N. Ayoobi, M. Ghavami, and A. M. Rabiei, “A novel motion-model-free UWB short-range positioning method,” *Signal, Image and Video Processing*, vol. 14, pp. 815–820, Dec. 2019.
- [117] A. Ledergerber and R. D’Andrea, “Ultra-wideband range measurement model with gaussian processes,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1929–1934, 2017.
- [118] J. Youssef, B. Denis, C. Godin, and S. Lesecq, “Enhanced UWB indoor tracking through NLOS TOA biases estimation,” in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, IEEE, 2008.
- [119] A. Ledergerber and R. D’Andrea, “Calibrating away inaccuracies in ultra wideband range measurements: A maximum likelihood approach,” *IEEE Access*, vol. 6, pp. 78719–78730, 2018.
- [120] N. M. Senevirathna, O. De Silva, G. K. I. Mann, and R. G. Gosine, “Kalman filter based range estimation and clock synchronization for ultra wide band networks,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9027–9032, 2020.
- [121] B. van der Heijden, A. Ledergerber, R. Gill, and R. D’Andrea, “Iterative bias estimation for an ultra-wideband localization system,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1391–1396, 2020.
- [122] K. Yu, K. Wen, Y. Li, S. Zhang, and K. Zhang, “A novel nlos mitigation algorithm for uwb localization in harsh indoor environments,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 686–699, 2019.
- [123] T. Zhou, Y. Cheng, B. Lian, and Y. Zhang, “Research on uwb localization based on toa in indoor nlos environment,” in *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 983–988, 2018.

- [124] A. Gfrerrer, “Geometry and kinematics of the mecanum wheel,” *Computer Aided Geometric Design*, vol. 25, pp. 784–791, Dec. 2008.
- [125] G. Campion, G. Bastin, and B. Dandrea-Novel, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, 1996.
- [126] Y. Jiang and V. C. Leung, “An asymmetric double sided two-way ranging for crystal offset,” in *2007 International Symposium on Signals, Systems and Electronics*, pp. 525–528, 2007.
- [127] D. Neiryneck, E. Luk, and M. McLaughlin, “An alternative double-sided two-way ranging method,” in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, IEEE, Oct. 2016.
- [128] R. Zetik, M. Eschrich, S. Jovanoska, and R. S. Thoma, “Looking behind a corner using multipath-exploiting UWB radar,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, pp. 1916–1926, July 2015.
- [129] T. Borangiu, ed. Wiley-IEEE Press, 2022.
- [130] Y. Xu, Y. S. Shmaliy, C. K. Ahn, G. Tian, and X. Chen, “Robust and accurate UWB-based indoor robot localisation using integrated EKF/EFIR filtering,” *IET Radar, Sonar & Navigation*, vol. 12, pp. 750–756, July 2018.
- [131] O. Ibarra-Manzano, L. Morales-Mendoza, and Y. S. Shmaliy, “FIR smoothing of discrete-time state-space models with applications to clocks,” in *2011 19th European Signal Processing Conference*, pp. 1800–1804, 2011.
- [132] J. Gamba, *Radar signal processing for autonomous driving*, vol. 1456. Springer, 2020.
- [133] P. Vorst, A. Koch, and A. Zell, “Efficient self-adjusting, similarity-based location fingerprinting with passive UHF RFID,” in *2011 IEEE Int. Conf. on RFID-Techn. Appl.*, pp. 160–167, Sitges, Spain, 2011.

# Anexo A

## Algoritmos

---

**Algoritmo:**  $H_2$ -OFIR a posteriori iterativo para matrices de error diagonales

---

**Datos:**  $\hat{x}_m, y_k, u_k, \mathbf{P}_m, \mathbf{Q}, \mathbf{R}, \mathbf{N}$

**Resultado:**  $\hat{x}_k$

**inicio**

**para**  $k = 1, 2, \dots$  **hacer**

**si**  $k > N - 1$

$m = k - N + 1$

**de otra forma**

$m = 0$

**para**  $i = m + 1, m + 2, \dots k$  **hacer**

$\hat{x}_i^- = \mathbf{F}\hat{x}_{i-1} + \mathbf{E}u_i$

$\mathbf{P}_i^- = \mathbf{F}\mathbf{P}_{i-1}\mathbf{F}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T$

$\mathbf{S}_i = \mathbf{H}\mathbf{P}_i^-\mathbf{H}^T + \mathbf{R}$

$\mathbf{K}_i = \mathbf{P}_i^-\mathbf{H}^T\mathbf{S}_i^{-1}$

$\hat{x}_i = \hat{x}_i^- + \mathbf{K}_i(y_i - \mathbf{H}\hat{x}_i^-)$

$\mathbf{P}_i = (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i)\mathbf{P}_i^-$

---

**Algoritmo:** cEKF para  $w_n$  y  $\bar{v}_n$  correlacionadas

---

**Datos:**  $\varsigma_n, \hat{x}_0, u_k, \Psi_n, P_0, Q_n, R_n$

**Resultado:**  $\hat{x}_n, P_n$

inicio

para  $n = 1, 2, \dots$  hacer

$$\begin{aligned}
 & z_n = \varsigma_n - \Psi_n \varsigma_{n-1}; \\
 & \hat{x}_n^- = f(\hat{x}_{n-1}, u_n); \\
 & P_n^- = F_n P_{n-1}^- F_n^T + Q_n; \\
 & S_n = H_n P_n^- H_n^T - \Psi_n H_{n-1}^- F_n^{-1} (P_n^- - Q_n) H_n^T \\
 & \quad + \Psi_n H_{n-1}^- F_n^{-1} (P_n^- - Q_n) F_n^{-T} H_{n-1}^T \Psi_n \\
 & \quad - H_n (P_n^- - Q_n) F_n^{-T} H_{n-1}^T \Psi_n + R_n; \\
 & K_n = [P_n^- H_n^T - (P_n^- - Q_n) F_n^{-T} H_{n-1}^T \Psi_n] S_n^{-1}; \\
 & \hat{x}_n = \hat{x}_n^- + K_n [z_n - h(\hat{x}_n^-)]; \\
 & P_n = (I - K_n H_n) P_n^-; \\
 & \quad + K_n \Psi_n H_{n-1}^- F_n^{-1} (P_n^- - Q_n);
 \end{aligned}$$


---

---

**Algoritmo:** cEFIR para ruido coloreado

---

**Datos:**  $\varsigma_n$

**Resultado:**  $\hat{x}_k$

inicio

para  $k = N - 1, N, \dots$  hacer

$$m = k - N + 1, s = m + K - 1$$

$$G_s = I$$

para  $j = 1 : K - 1$  hacer

$$\lfloor \bar{x}_{m+j} = f(\bar{x}_m)$$

para  $l = s + 1 : k$  hacer

$$z_l = \varsigma_l - \Psi_l \varsigma_{l-1}$$

$$\hat{x}_l^- = f(\bar{x}_{l-1})$$

$$G_l = [D_l^T D_l + (F_l G_{l-1} F_l^T)^{-1}]^{-1}$$

$$K_l = G_l D_l^T$$

$$\bar{x}_l = F_l \bar{x}_{l-1} + K_l [z_l - h(\hat{x}_l^-) + \Psi_l h(\bar{x}_{l-1})]$$

$$\hat{x}_k = \bar{x}_k$$


---

---

**Algoritmo:** EOFIR iterativo

---

**Datos:**  $y_k, \hat{x}_m, P_m, Q_k, R_k, N$

**Resultado:**  $\hat{x}_k$

**para**  $k = 1, 2 \dots$  **hacer**

**si**  $k > N - 1$  **entonces**

$m = k - N + 1$

**en otro caso**

$m = 0$

**para**  $l = m + 1 : k$  **hacer**

$\bar{x}_l^- = f_l(\bar{x}_{l-1})$

$P_l^- = F_l P_{l-1}^- F_l^T + Q_l$

$S_l = H_l P_l^- H_l^T + R_l$

$K_l = P_l^- H_l^T S_l^{-1}$

$\bar{x}_l = \bar{x}_l^- + K_l (y_l - h_l(\bar{x}_l^-))$

$P_l = (I - K_l H_l) P_l^-$

$\hat{x}_k = \bar{x}_k$

---

# Anexo B

## Códigos en Matlab

```
1 function [x_est,Pn]=OFIR_diag(xm,Y_mk,UFIR_gain,F,B_mk,W_mk,H_mk,D_mk,...
2                               ,G_mk,Q_mk,R_mk)
3     Z_x = H_mk*x_m*H_mk';
4     Z_1 = D_mk(end-1:end,:)*Q_mk*G_mk';
5     Z_2 = G_mk*Q_mk*G_mk';
6
7     OFIR_gain = ((F^(N-1))*x_m*H_mk' + Z_1)*inv(Z_x + Z_2 + R_mk);
8     x_est = OFIR_gain*Y_mk;
9     B_mk = (F^(N-1))-OFIR_gain*H_mk;
10    W_mk = D_mk(end-1:end,:)-OFIR_gain*G_mk;
11    Pn = B_mk*x_m*B_mk' + W_mk*Q_mk*W_mk' + OFIR_gain*R_mk*OFIR_gain';
```

**Listing B.1:** *Función para calcular una estimación OFIR para matrices diagonales*

---

```

1 function [x_est,Pn]=SubH2OFIR(xm,Y_mk,UFIR_gain,N,K,F,B_mk,W_mk,H_mk,D_mk,...
2                                     ,G_mk,Q_mk,R_mk)
3     %Suboptimal H_2 Filter
4     subH_2_gain=UFIR_gain;
5     Z = sdpcov(K);
6     A = (F^(N-1))*x_m*(F^(N-1))' + ...
7         D_mk(end-1:end,:)*Q_mk*D_mk(end-1:end,:)' ;
8     C = (F^(N-1))*x_m*H_mk' + D_mk(end-1:end,:)*Q_mk*G_mk';
9     Omega_mk = G_mk*Q_mk*G_mk' + R_mk;
10
11     M = [Z-A+C*subH_2_gain'+subH_2_gain*C' subH_2_gain;
12         subH_2_gain' inv(H_mk*x_m*H_mk'+Omega_mk)];
13
14     subH_2_gain = (F^(N-1)*x_m*H_mk'+D_mk(end-1:end,:)*Q_mk*G_mk')...
15                 /(H_mk*x_m*H_mk'+G_mk*Q_mk*G_mk'+R_mk);
16
17     x_est = subH_2_gain*Y_mk;
18     B_mk = (F^(N-1))-subH_2_gain*H_mk;
19     W_mk = D_mk(end-1:end,:)-subH_2_gain*G_mk;
20     Pn = B_mk*x_m*B_mk' + W_mk*Q_mk*W_mk' + subH_2_gain*R_mk*subH_2_gain';

```

**Listing B.2:** *Función para el filtro H2 subóptimo*

```

1 function [x_est,Pn,Pn_]=CMNKFCORR(xp,zn,Psi,A,B,C,Pnm,Qn,Rn)
2     % Measurement differencing
3     Gn=Psi*C/A;
4     Dn=C-Gn;
5     Phin=B*Qn*B'*Gn';
6     % Perform the time update
7     Pn_=A*Pnm*A'+B*Qn*B';
8     Sn=Dn*Pn_*Dn'+Rn+C*Phin+Phin'*Dn';
9     Kn=(Pn_*Dn'+Phin)/Sn;
10    xnm=A*xp;
11    x_est=xnm+Kn*(zn-Dn*xnm);
12    Pn=(eye(numel(xp))-Kn*Dn)*Pn_-Kn*Phin';

```

**Listing B.3:** *Función para Filtro de Kalman para ruido coloreado con correlación al ruido de proceso*

```

1 function [x_est,Pn]=cEKF(xp,zn,Psi,Fn,Hn,Hnm,hn_,hn__,Pnm,Qn,Rn,Bn)
2     Gn=Psi.*Hnm*Fn';
3     Dn=Hn-Gn;
4     Phin=Bn*Qn*Bn'*Gn';
5     %perform the time update
6     Pn_=Fn*Pnm*Fn'+Bn*Qn*Bn';
7     Sn=Dn*Pn_*Dn'+Rn+Hn*Phin+Phin'*Dn';
8     Kn=(Pn_*Dn'+Phin)/Sn;
9     x_est=xp+Kn*(zn-hn_+Psi.*hn__);
10    Pn=(eye(3)-Kn*Dn)*Pn_-Kn*Phin';

```

**Listing B.4:** *Función para calcular una estimación OFIR para matrices diagonales*

---

```

1 function x_est = cEFIR(s,N,x_s,u,y_k,Psi,Hnm,tag,G_s,fn,hn)
2     ntags=12;
3     for l = s+1:N-1
4         z_km=y_k(:,l+1)-Psi*y_k(:,l);
5         x_lm = fn(x_s,u,0.55);
6         F_ms=EvalF(x_s,u);
7         H_ms=EvalH(x_lm,tag);
8         Gn=Psi*Hnm*F_ms';
9         D_ms=H_ms-Gn;
10
11         G_s = inv(D_ms'*D_ms+inv(F_ms*G_s*F_ms'));
12         K_l = G_s*H_ms';
13         hn_=hn(x_lm(1:2),tag');
14         hn_(ntags+1)=x_lm(3);
15         hn__=hn(x_s(1:2),tag');
16         hn__(ntags+1)=x_s(3);
17         x_s = x_lm + K_l*(z_km-hn_+Psi*hn__);
18         Hnm=EvalH(x_s,tag);
19     end
20     x_est = x_s;

```

**Listing B.5:** *Función para calcular una estimación OFIR para matrices diagonales*

```

1 % EKF filter
2 % This filter was taken from Shmaliy, Y. S., Zhao, S., & Ahn, C. K. (2017).
3 % Unbiased finite impulse response filtering:
4 % An iterative alternative to Kalman filtering ignoring noise
5 % and initial conditions. IEEE Control Systems Magazine, 37(5), 79."
6 function [x_est,P_k] = EKF(x_km,h_k,A_k,C_k,T_k,Y_k,W_k,Q_k,R_k,P_k)
7     P_km=A_k*P_k*A_k'+W_k*Q_k*W_k';
8     K_k=P_km*C_k'/(C_k*P_km*C_k'+T_k*R_k*T_k');
9     x_est=x_km+K_k*(Y_k-h_k);
10    P_k=(eye(size(K_k,1))-K_k*C_k)*P_km;

```

**Listing B.6:** *Función para el Filtro de Kalman Extendido*

# Anexo C

## Producción Académica

### C.1. Artículos en revista

Ortega-Contreras, J.-A., Alvarado-Méndez, E., Almanza-Rodríguez, G., Hernández, M. del C., & Celaya-García, L. (2023). Glucose Sensor Using Sol–Gel Coating Layer Deposited on PMMA Optical Fiber: An Enzyme Activity Measurement System. In *Gels* (Vol. 9, Issue 8, p. 608). MDPI AG. <https://doi.org/10.3390/gels9080608>

Pale-Ramon, E. G., Morales-Mendoza, L. J., González-Lee, M., Ibarra-Manzano, O. G., Ortega-Contreras, J. A., & Shmaliy, Y. S. (2023). Improving Visual Object Tracking Using General UFIR and Kalman Filters Under Disturbances in Bounding Boxes. In *IEEE Access* (Vol. 11, pp. 57905–57915). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/access.2023.3280420>

Ortega-Contreras, J. A., Shmaliy, Y. S., Andrade-Lucio, J. A., & Ibarra-Manzano, O. G. (2022). Robust  $H_2$ -OFIR Filtering: Improving Tracking of Disturbed Systems Under Initial and Data Errors. In *IEEE Transactions on Aerospace and Electronic Systems* (Vol. 58, Issue 5, pp. 4761–4770). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/taes.2022.3155588>

Pale-Ramon, E. G., Shmaliy, Y. S., Morales-Mendoza, L. J., Gonzalez-Lee, M., Ortega-Contreras, J. A., & Uribe-Murcia, K. (2022). Estimation of states with data under Colored Measurement Noise (CMN). In *WSEAS TRANSACTIONS ON SYSTEMS AND CONTROL* (Vol. 17, pp. 359–367). World Scientific and Engineering Academy and Society (WSEAS). <https://doi.org/10.37394/23203.2022.17.40>

Ortega-Contreras, J. A., S. Shmaliy, Y., Andrade-Lucio, J. A., Vazquez-Olguin, M., Pale-Ramon, E. G., & Uribe-Murcia, K. (2022). Tracking a Mobile Robot in a UWB-sensor Grid. In WSEAS TRANSACTIONS ON SYSTEMS (Vol. 21, pp. 134–139). World Scientific and Engineering Academy and Society (WSEAS). <https://doi.org/10.37394/23202.2022.21.14>

Uribe-Murcia, K., Ortega-Contreras, J. A., Pale-Ramon, E. G., Vazquez-Olguin, M., & Shmaliy, Y. S. (2022). Comparison of the Kalman Filter and the Unbiased FIR Filter for Network Systems with Multiples Output Delays and Lost Data. In WSEAS TRANSACTIONS ON CIRCUITS AND SYSTEMS (Vol. 21, pp. 176–181). World Scientific and Engineering Academy and Society (WSEAS). <https://doi.org/10.37394/23201.2022.21.19>

Pale-Ramon, E. G., Ortega-Contreras, J. A., Uribe-Murcia, K. J., & Shmaliy, Y. S. (2022). Effect of sampling time jitter on robust  $H_2$  filtering estimates. In Signal Processing (Vol. 198, p. 108597). Elsevier BV. <https://doi.org/10.1016/j.sigpro.2022.108597>

Ortega-Contreras, J. A., Andrade-Lucio, J. A., Ibarra-Manzano, O. G., & Shmaliy, Y. S. (2023). TWOR: Improving Modeling and Self-Localization in RFID-Tag Networks Under Colored Noise. In IEEE Access (Vol. 11, pp. 25583–25592). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/access.2022.3222397>

Ortega-Contreras, J. A., Shmaliy, Y. S., & Andrade-Lucio, J. A. (2021). Three-Wheeled Omnidirectional Robot Localization in RFID-Tag Environments using UFIR Filtering. In WSEAS TRANSACTIONS ON SIGNAL PROCESSING (Vol. 17, pp. 16–21). World Scientific and Engineering Academy and Society (WSEAS). <https://doi.org/10.37394/232014.2021.17.2>

Ortega-Contreras, J., Pale-Ramon, E., Shmaliy, Y. S., & Xu, Y. (2021). A Novel  $H_2$  Approach to FIR Prediction Under Disturbances and Measurement Errors. In IEEE Signal Processing Letters (Vol. 28, pp. 150–154). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/lsp.2020.3048621>

## C.2. Artículos en Congreso

Ortega-Contreras, J. A., Pale-Ramon, E. G., Vazquez-Olguin, M.-A., Andrade-Lucio, J. A., Ibarra-Manzano, O., & Shmaliy, Y. S. (2023). Tracking a Holonomic Mobile Robot with Systematic Odometry Errors. In 2023 International Conference on Control, Artificial Intelligence, Robotics and Optimization (ICCAIRO).

2023 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO). IEEE. <https://doi.org/10.1109/iccairo58903.2023.00023>

Pale-Ramon, E. G., Shmaliy, Y. S., Morales-Mendoza, L. J., Ortega-Contreras, J. A., & Vazquez-Olguin, M. A. (2023). Improvement Object Tracking Process Through Filtering Methods. In 2023 International Conference on Control, Artificial Intelligence, Robotics and Optimization (ICCAIRO). 2023 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO). IEEE. <https://doi.org/10.1109/iccairo58903.2023.00029>

Pale-Ramon, E. G., Shmaliy, Y. S., Morales-Mendoza, L. J., González-Lee, M., Ortega-Contreras, J. A., & Uribe-Murcia, K. J. (2022). Object tracking with data under Colored Measurement Noise (CMN). In *Frontiers in Optics + Laser Science 2022 (FIO, LS)*. Frontiers in Optics. Optica Publishing Group. <https://doi.org/10.1364/fio.2022.jw4a.64>

Ortega-Contreras, J., Shmaliy, Y. S., Andrade-Lucio, J., Vazquez-Olguin, M.-A.. Kalman-like disturbance observer-based control, November 2021, CNCA 2021

Shmaliy, Y. S., Andrade-Lucio, J., Pale-Ramon, E. G., Ortega-Contreras, J., Morales-Mendoza, L. J., & Gonzalez-Lee, M. (2020). Visual Object Tracking with Colored Measurement Noise using Kalman and UFIR Filters. In 2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). 2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). IEEE. <https://doi.org/10.1109/cce50788.2020.9299345>

Ortega-Contreras, J. A., Ramon, E. P., Lucio, J. A., Shmaliy, Y., & Zhao, S. (2020). Algorithms Design for Tracking Moving Objects with Colored Process Noise. In 2020 24th International Conference on Circuits, Systems, Communications and Computers (CSCC). 2020 24th International Conference on Circuits, Systems, Communications and Computers (CSCC). IEEE. <https://doi.org/10.1109/csc49995.2020.00012>

Ramon, E. P., Contreras, J. O., Lucio, J. A., Shmaliy, Y., & Zhao, S. (2020). Recursive Algorithms for Visual Target Tracking Under Colored Measurement Noise. In 2020 24th International Conference on Circuits, Systems, Communications and Computers (CSCC). 2020 24th International Conference on Circuits, Systems, Communications and Computers (CSCC). IEEE. <https://doi.org/10.1109/csc49995.2020.00010>

### C.3. Capítulos en Libro

Ortega-Contreras, J. A., Espinoza-Torres, I., Pale-Ramón, E. G., Andrade-Lucio, J. A., Ibarra-Manzano, O., & Shmaliy, Y. (2023). Off-Line and Real-Time Estimators for Monocular Position-Based Visual Servoing. In *Lecture Notes in Electrical Engineering* (pp. 467–473). Springer Nature Singapore. [https://doi.org/10.1007/978-981-99-2730-2\\_46](https://doi.org/10.1007/978-981-99-2730-2_46)

Pale-Ramon, E. G., Shmaliy, Y. S., Morales-Mendoza, L. J., González-Lee, M., Ortega-Contreras, J. A., & Uribe-Murcia, K. J. (2023). Aerial Object Tracking Using Filtering Methods. In *Lecture Notes in Electrical Engineering* (pp. 603–616). Springer Nature Singapore. [https://doi.org/10.1007/978-981-99-2730-2\\_58](https://doi.org/10.1007/978-981-99-2730-2_58)

Ortega-Contreras, J. A., Shmaliy, Y. S., & Andrade-Lucio, J. A. (2022). Improvements on PIV Servo-Controller Using Online Disturbance Estimation. In *Lecture Notes in Electrical Engineering* (pp. 3–13). Springer Nature Singapore. [https://doi.org/10.1007/978-981-19-3927-3\\_1](https://doi.org/10.1007/978-981-19-3927-3_1)

### C.4. Producción de Capital Humano

Titulación del alumno Victor Manuel Vega Gutiérrez con la Tesis de Licenciatura titulada “Corte por plasma automatizado aplicado en sistemas de tubería” el día 10 de Noviembre del 2022.