



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PROYECTO DE INVESTIGACIÓN

**MODELO MATEMÁTICO BASADO EN EKF PARA SEGUIMIENTO DE TRAYECTORIA Y
POSICIÓN DE UN ROBOT MÓVIL**

PREVIO A LA OBTENCIÓN DEL TÍTULO:

INGENIERO EN MECATRÓNICA

PRESENTADO POR:

21711392 ESTEFANY PAMELA ALMENDARES AVILA

ASESOR:

ING. JOSÉ LUIS ORDOÑEZ ÁVILA

SAN PEDRO SULA, CORTÉS

CAMPUS: SAN PEDRO SULA; JULIO, 2021

DERECHOS DE AUTOR

©Copyright 2021

Todos los derechos son reservados

DEDICATORIA

Naturalmente el primer homenajeado en todo es Dios y también es mi caso, dedico este y todos los momentos de mi vida plenamente a él, por acompañarme segundo a segundo en la travesía de la vida, en ningún momento por difícil que fue se apartó de mí lado y siempre me dio la fuerza para seguir y dar lo mejor de mí.

A mi madre, Martina Avila que es el pilar de mi vida y que sin duda gran parte de lo que soy y quiero llegar a ser ha sido influenciado por ella, por su amor y valentía y porque no conozco a otra mejor persona en el mundo.

A mis hermanos, Deiby Almendares y Celeste Avila que son mi condimento diario y por quienes muchas cosas toman sentido.

A mi abuela, Martina Hernández que ha sido mi compañera desde siempre y nunca me ha dejado tirar la toalla.

AGRADECIMIENTO

A lo largo de la vida y en ciertos momentos específicos son muchas los rayos de luz en forma de personas que nos marcan y dan un poco de sí mismos para ser de nuestra experiencia mucho mejor en comparación que si lo hiciéramos solos, poder contar con ellas es algo maravilloso y que de corazón nunca podré compensar porque dinero y bienes se pueden devolver, pero el tiempo, el amor, la amistad, las risas, los llantos y cada momento a memoria son cosas eternas.

Agradezco al divino creador de este vasto universo por este preciso momento y por haberme permitido vivirlo y sentirlo al máximo.

A mi madre que no puedo estar más agradecida de ser su hija, por enseñarme lo dura y bella que puede ser la vida, por mostrarme que no hay límites cuando se tiene fe y que muchas veces un abrazo y una oración juntas lo puede todo, por confiar en mí y quedarse a mi lado cada madrugada de desvelo.

Agradezco profundamente a mi hermano Deiby Almendares, el cual nunca ha dejado de velar por nuestra familia y de ser un apoyo emocional enorme, por nunca cruzarse de brazos en los momentos de crisis y decirme siempre “vos puedes, concéntrate”, a mi hermanita Celeste Avila que, aunque no tiene idea de lo que hace en mí, es la luz que alegra cada mañana, sus muestras de amor y su forma tan inocente de ver la vida me han enseñado a ser mejor para que pueda ver un buen ejemplo en mí.

A mi tía Mirian Bustillo y su esposo Manuel Mendoza quienes abrieron las puertas de su casa y me hicieron parte de su pequeña familia por casi 5 años para que pudiese cumplir una de mis metas y nunca dejaron de ver por mí.

A mi abuela Martina Avila, por ser como es y de esa manera formar en mí más confianza y determinación. A mis tías y primos que a lo largo de mi vida estudiantil jamás dejaron de confiar en mí y que hasta hoy siento su apoyo. A mi mejor amigo Milton Flores quien me ha acompañado durante todo este tiempo y siempre ha tenido palabras de aliento para mí.

Al Ing. José Luis Ordoñez por retarme a tomar esta buena decisión y ser un excelente guía académico, por todo su apoyo y comprensión a lo largo de cada clase en la que tuve el honor de tenerle como docente.

A mis amigos, que me han acompañado en el trascurso de esta travesía desde el primer instante hicieron de mi vida universitaria una aventura hermosa, a Odalis Lizardo que a pesar de la distancia fue mi apoyo emocional siempre, a Nohelia Cruz, Dennys Cantarero, Cristhian Méndez y Abraham Rodríguez que han estado pendiente de cada momento su ayuda y apoyo incondicional no han faltado, a Wilfredo Lontero que a lo largo de las 10 semanas de proyecto ha sido el compañero de todos los días; retándonos y motivándonos mutuamente, a Astrid Aguilar, Ariana Garrido, Jorge Matute, Riccy Gómez, Jonathan Rivera, Emanuel Lara, Gustavo Romero, Julia Maldonado, Alejandro Valle, Cesar Núñez, Rolando Maradiaga y Mario Aguilar a quienes llevo en mi corazón por cada momento compartido y por seguir presentes en mi vida.

Y no podría parar de agradecer a cada persona que ha está cerca de mí y de quienes he aprendido muchas cosas, a mis maestros, compañeros y jefes de área de trabajo, a la Lic. Erika Mejía, Al Dr. Rodolfo Sorto y a la Dra. Dania Mena de quienes me llevo el mejor concepto, me mostraron paciencia, confianza y fue un placer el poder trabajar junto a ellos y conocer su humanidad y un poco de esa labor tan formidable que realizan para la educación de los demás.

EPÍGRAFE

"Investigación es lo que hago cuando no sé lo que estoy haciendo"

Wernher von Braun

RESUMEN EJECUTIVO

Con el avance tecnológico crece la necesidad de optimización en las tareas realizadas por robots móviles, estos deben ser dotados de autonomía en su navegación y otras decisiones, esto se debe a que los entornos de operación se vuelven más complejos y con obstáculos y objetivos que cambian de posición, sumando a esto las interferencias del espacio de trabajo como perturbaciones en los sensores y actuadores, en el ambiente y otros ruidos del sistema. Para hacer frente a esta problemática se recurre al uso de algoritmos de navegación, creación de rutas, estimación de posición, seguimiento de caminos, reconocimiento de objetos y otros. En el presente estudio se utilizó una metodología en espiral con la que se desarrolló un modelo matemático basado en el filtro de Kalman extendido para el seguimiento de trayectoria y estimación de posición de un robot diferencial utilizando la cinemática de este. Para la validación del modelo matemático se utilizó Matlab junto a CoppeliaSim donde se realizaron los distintos escenarios de prueba, posteriormente se comparan los resultados de EKF con los del algoritmo de Kevin Passino, encontrando superior en la mayoría de los aspectos al EKF excepto en el tiempo donde el algoritmo de Kevin Passino logra un menor tiempo de simulación.

Palabras claves: escenario, filtro de Kalman, obstáculos, posición del robot, trayectoria.

ABSTRACT

With technological advancement, the need for optimization in the tasks carried out by mobile robots grows, they must be endowed with autonomy in their navigation and other decisions, this is because operating environments become more complex with obstacles and targets that change position, adding to this interference from the workspace such as disturbances in sensors and actuators, in the environment, and other system noise. To deal with this problem, use of algorithms for navigation, route creation, position estimation, path tracking, object recognition, and others are used. In the present study, a spiral methodology was used with which a mathematical model based on the extended Kalman filter was developed to track the trajectory and position estimation of a differential robot using its kinematics. For the validation of the mathematical model, MATLAB was used together with CoppeliaSim where the different test scenarios were carried out, later the EKF results are compared with those of the Kevin Passino algorithm, finding superior in most aspects to the EKF except in the time where Kevin Passino's algorithm achieves a lower simulation time.

Keywords: scenario, Kalman filter, obstacles, robot position, trajectory.

ÍNDICE DE CONTENIDO

CAPÍTULO I. INTRODUCCIÓN.....	1
CAPÍTULO II. PLANTEAMIENTO DEL PROBLEMA.....	3
2.1 PRECEDENTES DEL PROBLEMA.....	3
2.2 DEFINICIÓN DEL PROBLEMA.....	4
2.3 JUSTIFICACIÓN.....	4
2.4 PREGUNTAS DE INVESTIGACIÓN.....	5
2.5 OBJETIVOS.....	5
2.5.1 OBJETIVO GENERAL.....	5
2.5.2 OBJETIVOS ESPECÍFICOS.....	5
CAPITULO III. MARCO TEÓRICO.....	6
3.1 MODELOS SEGÚN SU ALEATORIEDAD.....	6
3.1.1 ESTOCÁSTICOS.....	6
3.1.2 DETERMINISTA.....	7
3.2 ROBOTS MÓVILES CON RUEDAS.....	9
3.2.1 TRANSMISIÓN DIFERENCIAL.....	9
3.2.2 ACKERMANN.....	11
3.2.3 OMNIDIRECCIONAL.....	12
3.3 ALGORITMOS PARA TRAYECTORIA DE ROBOT MÓVILES.....	14
3.3.1 TÉCNICA DE A*.....	14
3.3.2 ARBOLES ALEATORIOS DE EXPLORACIÓN RÁPIDA O RRT.....	15
3.3.3 ALGORITMO DE MONTE CARLO.....	16
3.3.4 ALGORITMO DE DIJKSTRA.....	17
3.3.5 ALGORITMO DE KEVIN PASSINO.....	19

3.3.6 FILTRO DE KALMAN	20
3.3.6.1 ESPACIO DE ESTADOS LINEAL	21
3.3.6.2 FUNCIONAMIENTO DEL FILTRO DE KALMAN	22
3.4 VARIACIONES DEL FILTRO DE KALMAN.....	23
3.4.1 FILTRO DE KALMAN EXTENDIDO.....	24
3.4.2 FILTRO DE KALMAN SIN PERFUME	25
3.4.3 FILTRO DE KALMAN DE CUBICACIÓN DE RAÍZ CUADRADA.....	26
3.5 APLICACIONES DEL FILTRO DE KALMAN.....	27
3.5.1 PREDICCIÓN DE UBICACIÓN EN VANET´S	28
3.5.2 MOTORES.....	29
3.5.3 VIGILANCIA EN AVIACIÓN.....	30
3.5.4 MEJORA EN LA SEÑAL LiDAR 3D DE VEHÍCULOS AUTÓNOMOS	30
3.5.5 SEGUIMIENTO DE OBJETOS	31
3.5.6 PROCESAMIENTO DE DATOS DE RADAR TERRESTRE INTERFERÓMETRICO	31
CAPÍTULO IV. METODOLOGÍA	33
4.1 ENFOQUE	33
4.2 VARIABLES DE INVESTIGACIÓN.....	33
4.3 TÉCNICAS E INSTRUMENTOS APLICADOS.....	34
4.4 METODOLOGÍA DE ESTUDIO.....	35
4.4.1 ITERACIÓN I. DISEÑO DEL DIAGRAMA CINEMÁTICO DEL ROBOT	35
4.4.2 ITERACIÓN II. IMPLEMENTACIÓN ALGORITMO DE FILTRO DE KALMAN	36
4.4.3 ITERACIÓN III. DESARROLLO DE MATRICES	37
4.4.4 ITERACIÓN IV. ELABORACIÓN DE ESCENARIOS Y SIMULACIONES	38
4.5 CRONOGRAMA DE ACTIVIDADES	39

CAPÍTULO V. ANÁLISIS Y RESULTADOS	40
5.1 ANÁLISIS DE TEORÍA DE SUSTENTO.....	40
5.2 RESULTADOS	41
5.2.1 DISEÑO DEL DIAGRAMA CINEMÁTICO DEL ROBOT	41
5.2.2 IMPLEMENTACIÓN DEL ALGORITMO DE FILTRO DE KALMAN.....	45
5.2.3 DESARROLLO DE MATRICES DEL SISTEMA.....	48
5.2.4 ELABORACIÓN DE ESCENARIOS Y SIMULACIONES.....	52
5.2.4.1 SIMULACIÓN Y COMPARACIONES	54
5.3 DISCUSIÓN	70
CAPÍTULO VI. CONCLUSIONES.....	72
CAPÍTULO VII. RECOMENDACIONES.....	73
BIBLIOGRAFÍA.....	74
ANEXOS	83
ROBOT PIONEER P3DX EN SOLIDWORKS	83
ESCENARIO 1 Y 2.....	83
ESCENARIO 3 Y 4.....	84
ESCENARIO 5	84

ÍNDICE DE FIGURAS

Figura 1. Nuevos entornos de trabajo de robots móviles	9
Figura 2. Geometría de robot diferencial de 3 ruedas	10
Figura 3. Geometría del giro de robot tipo Ackermann.....	12
Figura 4. Diagrama de velocidades de robot omnidireccional de 3 ruedas.....	13
Figura 5. Función obstáculos y contorno de las formas.....	19
Figura 6. Rudolf Emil Kalman.....	20
Figura 7. Comparación de, ASRCKF, SRCKF y EKF en RMSE	29
Figura 8. Comparación de señal estimada por filtro de Kalman y la original	31
Figura 9. Variables de investigación.....	34
Figura 10. Primera iteración	36
Figura 11. Segunda iteración	37
Figura 12. Tercera iteración.....	38
Figura 13. Cuarta iteración.....	39
Figura 14. Cronograma de actividades para el proyecto.....	39
Figura 15. Dimensiones de robot diferencial Pioneer P3DX en mm	42
Figura 16. Diagrama cinemático del Pioneer P3DX	43
Figura 17. Linealización mediante serie de Taylor.....	46
Figura 18. Nueva posición del robot con respecto a la anterior.....	49
Figura 19. Modelo matemático de filtro de Kalman extendido para robot móvil diferencial	52
Figura 20. Prueba inicial escenario 1, trazado de ruta	54
Figura 21. Prueba inicial escenario 1, llegada a la meta.....	55
Figura 22. Simulación escenario 2, prueba inicial.....	56
Figura 23. Mapeo en escenario 1 con EKF.....	59
Figura 24. Mapeo en escenario 1 con algoritmo de Kevin Passino	59
Figura 25. Meta alcanzada y trayectoria en escenario 1	60
Figura 26. Mapeo en escenario 2.....	61
Figura 27. Meta alcanzada y trayectoria en escenario 2	62
Figura 28. Mapeo en escenario 3.....	63

Figura 29. Meta alcanzada y trayectoria en escenario 3	64
Figura 30. Mapeo en escenario 4 con EKF.....	65
Figura 31. Mapeo en escenario 4 con algoritmo de Kevin Passino	66
Figura 32. Meta alcanzada y trayectoria en escenario 4, ruta izquierda.....	66
Figura 33. Meta alcanzada y trayectoria en escenario 4, ruta derecha.....	67
Figura 34. Mapeo en escenario 5.....	68
Figura 35. Meta alcanzada y trayectoria en escenario 5	69

ÍNDICE DE TABLAS

Tabla 1. Resumen de modelo estocástico y determinístico.....	8
Tabla 2. Algoritmo RRT tradicional.....	15
Tabla 3. Parámetros de espacio de estados.....	22
Tabla 4. Cuadro comparativo de algoritmos	23
Tabla 5. Parámetros desconocidos de SRCKF	27
Tabla 6. Matriz de riesgo en iteración I	42
Tabla 7. Datos del robot.....	43
Tabla 8. Matriz de riesgo en iteración II	45
Tabla 9. Matriz de riesgos en iteración III.....	49
Tabla 10. Matriz de riesgo en iteración IV.....	53
Tabla 11. Parámetros y resultados de escenarios 1 y 2 en pruebas iniciales	57
Tabla 12. 3 ejemplos de mejores resultados del EKF en pruebas iniciales.....	57
Tabla 13. Parámetros del EKF final	58
Tabla 14. Comparación entre resultado de los algoritmos en escenario 1.....	61
Tabla 15. Comparación entre resultado de los algoritmos en escenario 2.....	62
Tabla 16. Comparación entre resultado de los algoritmos en escenario 3.....	64
Tabla 17. Comparación entre resultado de los algoritmos en escenario 4.....	67
Tabla 18. Comparación entre resultado de los algoritmos en escenario 5.....	69

ÍNDICE DE ECUACIONES

Ecuación 1. Proceso infinitesimal	7
Ecuación 2. Modelo de crecimiento poblacional	8
Ecuación 3. Crecimiento poblacional exponencial	8
Ecuación 4. Valor de G en el eje x.....	11
Ecuación 5. Valor de G en el eje y.....	11
Ecuación 6. Derivada de x.....	11
Ecuación 7. Derivada de y.....	11
Ecuación 8. Derivada de la orientación.....	11
Ecuación 9. Giro de un robot tipo Ackermann.....	12
Ecuación 10. Ecuaciones cinemáticas del robot omnidireccional.....	13
Ecuación 11. Ecuaciones cinemáticas expresadas en matriz	13
Ecuación 12. Ángulo de dirección.....	13
Ecuación 13. Costo real de avance	15
Ecuación 14. Número de simulaciones del algoritmo.....	17
Ecuación 15. Función de costo total	19
Ecuación 16. Modelo de espacio de estados	21
Ecuación 17. Modelo de espacio de estados simplificado de forma lineal	21
Ecuación 18. Estado estimado a priori	22
Ecuación 19. Covarianza del error a priori.....	22
Ecuación 20. Ganancia de Kalman	23
Ecuación 21. Actualización del estado con z_k	23
Ecuación 22. Covarianza del error a posteriori	23
Ecuación 23. Ecuaciones generales de un sistema no lineal.....	24
Ecuación 24. Etapa predictiva del EKF	24
Ecuación 25. Etapa correctiva del EKF	25
Ecuación 26. Matriz para puntos sigma.....	25
Ecuación 27. Estado estimado a priori UKF.....	26
Ecuación 28. Covarianza de error a priori UKF.....	26

Ecuación 29. Media a posteriori UKF	26
Ecuación 30. Covarianza a posteriori UKF.....	26
Ecuación 31. Covarianza entre los dos estados.....	26
Ecuación 32. Ganancia de UKF	26
Ecuación 33. Puntos de cubicación del estado.....	27
Ecuación 34. Valores medios de los puntos de estado	27
Ecuación 35. Raíz cuadrada de la covarianza a priori.....	27
Ecuación 36. Matriz de ganancia SRCKF.....	27
Ecuación 37. Estado actualizado SRCKF	27
Ecuación 38. Raíz cuadrada de la covarianza de error a posteriori.....	27
Ecuación 39. Modelo cinemático de robot diferencial	44
Ecuación 40. Velocidad lineal tangencial y velocidad angular.....	44
Ecuación 41. Expansión de Taylor.....	46
Ecuación 42. Ecuaciones de estado.....	46
Ecuación 43. Vector de estado.....	46
Ecuación 44. Señal de control	46
Ecuación 45. Ruido del sistema	46
Ecuación 46. Covarianza a priori	47
Ecuación 47. Jacobiano de predicción respecto del vector de estado	47
Ecuación 48. Jacobiano de predicción respecto de la matriz de control.....	47
Ecuación 49. Cálculo de ganancia del EKF.....	47
Ecuación 50. Jacobiano de medición respecto del estado z_k	47
Ecuación 51. Actualización del estado a posteriori EKF.....	47
Ecuación 52. Actualización de la covarianza EKF	48
Ecuación 53. Modelo cinemático de robot diferencial en un tiempo actual.....	50
Ecuación 54. Señal de control del robot	50
Ecuación 55. Señal de medición	50
Ecuación 56. Matriz jacobiana de predicción respecto del vector de estado.....	50
Ecuación 57. Matriz Jacobiana de predicción respecto del ruido.....	50

Ecuación 58. Matriz jacobiana de corrección respecto de la señal de medición.....	51
Ecuación 59. Matriz diagonal de covarianza a priori	51
Ecuación 60. Matriz diagonal de covarianza a posteriori	51

GLOSARIO

- 1) Campo potencial: es un campo escalar que describe la energía potencial por unidad de alguna cantidad debido a un campo vectorial,
- 2) Función de densidad de probabilidad: descripción de la probabilidad relativa según la cual la variable aleatoria tomara determinado valor.
- 3) Distancia euclidiana: número positivo que indica la separación que tienen dos puntos en un espacio donde se cumplen los axiomas y teoremas de la geometría de Euclides.
- 4) Distribución gaussiana: la distribución normal es aquella distribución de probabilidad cuya grafica adquiere forma de campana simétrica. Sirve para evaluar variables aleatorias que tendrán un comportamiento normal o aproximadamente normal.
- 5) Filtrado bayesiano: herramienta muy eficaz para poder calcular la probabilidad de que ocurra un suceso determinado basándose en la experiencia de casos anteriores.
- 6) Heurística: relacionado con técnicas exploratorias de resolución de problemas que utilizan técnicas de autoeducación para mejorar el desempeño.
- 7) Holonómica (o): referencia al caso en el que el número de grados de libertad es el mismo que el número de coordenadas generalizadas que describen un sistema.
- 8) Jacobiano (a): determinante que se define para un número finito de funciones del mismo número de variables y en el que cada fila consta de las primeras derivadas parciales de la misma función con respecto a cada una de las variables.
- 9) Probabilístico (a): los métodos probabilísticos se basan en la idea de que no puede estar seguro de los resultados o eventos futuros, pero puede juzgar si son probables o no y actuar sobre la base de este juicio.
- 10) Recursivo (a): implica el uso de un procedimiento, subrutina, función o algoritmo que se llama a sí mismo una o más veces hasta que se cumple una condición específica, momento en el que el resto de cada repetición se procesa desde la última llamada hasta la primera.

LISTA DE SIGLAS

AGV	Automated Guided Vehicle.
ASRCKF	Adaptative Square Root Cubature Kalman Filter.
DT-RRT*	Dual Tree – Rapidly-exploring Random trees modified.
EKF	Extended Kaman Filter.
KP	Kevin Passino.
KPND	Kalman Prediction-based Neighbor Discovery.
PDF	Probability Density Function.
RMSE	Root Mean Square Error.
RSSI	Received Signal Strength Indicator.
UKF	Unscented Kalman Filter.
WMR	Wheeled Mobile Robot.

CAPÍTULO I. INTRODUCCIÓN

Por su facilidad de operación en los distintos entornos de trabajo en la actualidad los robots móviles con ruedas han sido foco de innumerables estudios científicos y aplicados en distintas áreas de operación, además de adaptarse a entornos diversos también son muy fáciles de maniobrar, por esto cada día crece la necesidad de dotarlos con mayores competencias y que sus funciones y forma de afrontar las tareas sea lo más cercano a la inteligencia y autonomía de estos, uno de los puntos sin duda a mejorar y que seguirá siendo punto de estudio es si esta inteligencia puede lograrse a partir de métodos probabilísticos, preguntas como ¿Qué método es el mejor? ¿Realmente son la ruta para la navegación autónoma? seguirán abordándose en los próximos años, así que en esta investigación como una piedra más en la construcción de esta solución y por su naturaleza probabilística se propone la aplicación de un filtro de Kalman extendido para el modelado matemático de un robot tipo diferencial a partir de las ecuaciones de su cinemática llegando a comparar este con el algoritmo para trazado de trayectoria de Kevin Passino. A continuación, se desglosa el contenido de cada capítulo presentes en este trabajo de investigación.

Capítulo II. Se aborda el planteamiento del problema, desde la definición de este y la razón por la que se considera importante la realización de la investigación, así como la descripción de cada objetivo que se pretende alcanzar a lo largo del desarrollo de esta.

Capítulo III. Se desarrolla la teoría necesaria y sus fundamentos para la comprensión de los procesos a los que pueden estar sometidos los robots móviles, este no es un espacio para la conceptualización de términos conocidos o bastante abordados es muchas otras investigaciones en cuanto a robótica móvil, si no que en él se pretende dar un panorama general tanto de los robot con ruedas, como de los algoritmos utilizados para seguimiento de trayectoria, trazabilidad, estimación de posición y otras funciones, también se presenta diversas aplicaciones y resultados de otras investigaciones en torno al filtro de Kalman y algunas de sus versiones.

Capítulo IV. Una parte muy interesante de la investigación es la metodología en ella se define cual será el camino a seguir para poder lograr los objetivos propuestos, la presente tesis

se rige bajo la metodología en espiral, así que se describen las variables del sistema tanto dependientes como independientes y presenta el enfoque de la investigación, así como las herramientas a utilizar para su desarrollo, finalizando con un cronograma de actividades con el que se deberá guiar la realización de cada etapa de la investigación para poder finalizar satisfactoriamente y en el tiempo estipulado.

Capítulo V. En Análisis y resultados se presenta el seguimiento de la metodología propuesta en conjunto con la teoría de sustento del marco teórico, hasta alcanzar la simulación y validación del algoritmo en cuestión comparándolos resultados y la funcionalidad de este con otro algoritmo, para finalmente discutir dichos resultados y el aporte general de la investigación.

Capítulo VI. En las conclusiones se presentan la respuesta de la investigación en cuanto a los objetivos propuestos a partir del análisis de los resultados obtenidos para el filtro de Kalman extendido y el algoritmo de Kevin Passino.

CAPÍTULO II. PLANTEAMIENTO DEL PROBLEMA

El fin de toda investigación es el de suplir distintas necesidades que se detectan a partir de un problema conocido o que se va desarrollando a través del tiempo, en esta sección se plantea el problema en cuanto al seguimiento de trayectoria y estimación de posición de los robots móviles conociendo sus precedentes y el argumento del ¿Por qué? es necesaria esta investigación, estableciendo los objetivos que se esperan alcanzar al finalizar.

2.1 PRECEDENTES DEL PROBLEMA

La aplicación de modelos matemáticos para la autonomía de los robots móviles y el cumplimiento de las diferentes tareas de estos en formas cada vez más satisfactorias se ve cada vez más necesaria y es que los entornos de trabajo se vuelven más complicados y ya no se tratan de puntos estáticos si no que estos pueden llegar a cambiar, también otro punto importante es que las aplicaciones y la exigencia de rendimiento y operación de estos robot también va en aumento, en el estudio de (Cardona et al., 2018) se presenta un modelo matemático basado en cinemática, dinámica y electromecánica de un robot diferencial para la posición del centro de masa de este, en cambio (Ortiz A. et al, 2011) para el control de trayectoria, posición y seguimiento de ruta desarrolla un modelo matemático de la cinemática y dinámica de un robot móvil a partir de sus restricciones no holonómicas, el modelado matemático de robots móviles no es fácil en él se involucran muchos factores, (Ríos G. & Bueno L., 2008) presenta un modelo matemático de un robot diferencial a partir de la cinemática, dinámica y parte eléctrica de los motores, por otro lado en el estudio de (Freire C., 2009) el modelo matemático que se desarrolla para el movimiento de un robot de 3 ruedas y trazado de rutas se basa en la cinemática y dinámica y sus restricciones no holonómicas a partir de la utilización de las ecuaciones de Appel, el objeto principal de estos y otros estudios es el de lograr que los robots sean capaz de hacer su ruta, esquivar obstáculos, llegar a su destino y conocer la posición en cada momento, esto además de la aplicación de sensores apropiados se logra a partir del uso de algoritmos o de la creación de nuevos algoritmos a partir de otros, uno de los problemas que suelen dar dolores de cabeza es la cantidad de fórmulas matemáticas que se necesitan y se generan en la aplicación de estos algoritmos sumado a todos los factores que

se tomen en cuenta para modelar para eso en (Tzafestas, 2018) se hace una visualización general de los distintos algoritmos y métodos de navegación autónomas así como los libros de robótica móvil más utilizados en las últimas décadas.

2.2 DEFINICIÓN DEL PROBLEMA

Los entornos de operación y trabajos de los distintos robots móviles con ruedas y colaboradores están lejos de ser iguales, al contrario, cada vez se hacen más complejos, con personas a su alrededor, objetos que son cambiados de posición o incluso con un punto de meta en distintas posiciones del espacio, por esto la robótica móvil e investigadores corren una carrera contra reloj por dotar a estos robots de características cada vez más cercanas a la inteligencia y que estos hagan sus tareas de forma independiente y autónoma, tareas que hasta hace poco tiempo eran realizadas por personas, para atender a estas necesidades se han desarrollado un sin número de algoritmos donde los de naturaleza probabilística están siendo ampliamente aplicados pues brindan una estimación más cerca a la realidad y logran un efecto parecido al que se busca, que el robot decida en cada punto de su ruta hacia donde ir donde cada espacio del entorno tiene la misma probabilidad de encontrarse ocupado o no y es tarea del robot descubrirlo y planificar su camino en base a esto.

2.3 JUSTIFICACIÓN

En la búsqueda de un algoritmo que proporcione óptimos resultados en la autonomía móvil, se encuentra el filtro de Kalman extendido que brinda una estimación de estado con menor error que se combina con los datos que este recoge por su principio de recursividad, este no necesita tomar cada punto en el pasado para conocer un nuevo punto en el futuro, al contrario solo necesita de uno y lo hace a través de principios de probabilidad y espacios de estados no lineales, el conjunto de ecuaciones que forman este algoritmo se rigen bajo dos etapas complementarias entre sí, una de predicción y otra de actualización o corrección con el único objetivo de disminuir al máximo el error de covarianza estimado y aunque en ocasiones no se suelen presentar las condiciones precisas para su funcionamiento del filtro, el tal se puede desempeñar de manera eficiente en la mayoría de casos, para el modelado matemático a partir de este filtro de solamente la cinemática un robot móvil se busca dar a conocer que es posible

tratar los casos de estimación de posición y seguimiento de trayectoria sin necesidad de aplicar la dinámica y el conjunto de restricciones de tipo no holonómicas del robot pues este solo aplica su velocidad mínima para el control final, además de proporcionar resultados fiables en comparación al uso de otro algoritmo.

2.4 PREGUNTAS DE INVESTIGACIÓN

- 1) ¿Se puede obtener a partir de la cinemática de un robot diferencial un modelo matemático del filtro de Kalman extendido para el seguimiento de trayectoria y la estimación de posición del robot?
- 2) ¿Además de las distintas velocidades de sus ruedas de que más depende la posición del robot?
- 3) ¿Para la estimación de posición de un robot y mapeo simultaneo es lo mismo trabajar con el EKF que con el algoritmo de Kevin Passino?

2.5 OBJETIVOS

2.5.1 OBJETIVO GENERAL

Desarrollar un modelo matemático utilizando el algoritmo del filtro de Kalman extendido para la estimación de posición y seguimiento de trayectoria de un robot móvil tipo diferencial a partir de su cinemática.

2.5.2 OBJETIVOS ESPECÍFICOS

- 1) Diseñar el diagrama cinemático de un robot diferencial que permita conocer los parámetros necesarios para la movilidad del robot y como esto se vincula al EKF en el modelo matemático final.
- 2) Comparar el nivel de estimación de posición, seguimiento de trayectoria y eficiencia de ambos algoritmos en entornos iguales.

CAPITULO III. MARCO TEÓRICO

3.1 MODELOS SEGÚN SU ALEATORIEDAD

Cada proceso que se conoce se ve influenciado o afectado por estímulos, eventos o variables de entrada que serán las encargadas de saber cómo se comportará el sistema en un punto futuro, en tanto se estudiaran dos modelos y se conocerá la diferencia entre ambos, así como funciona su relación entre los datos estimados en la salida con su entrada.

3.1.1 ESTOCÁSTICOS

Los fenómenos naturales o ambientales, sin importar del que se hable, se ven perjudicados por los denominados eventos aleatorios esto se da porque dichos fenómenos son de naturaleza estocástica, en su mayoría los sistemas de ingeniería están expuestos a estas condiciones ambientales por lo que a la hora de su diseño el termino de aleatoriedad es suma importancia. Uno de los procesos aleatorios más comunes en ingeniería es el ruido, este genera una señal de perturbación no deseada en el sistema y su rendimiento en general, para tratar este tipo de problemas se hace uso de las matemáticas o ecuaciones estocásticas y de aleatoriedad, porque a diferencia de los procesos deterministas, los aleatorios no suelen ser tan precisos, así que se recurre al uso de análisis de probabilidad y métodos estocásticos, generalmente en forma de ecuaciones diferenciales o ecuaciones diferenciales estocásticas mediante las cuales procesos como el ruido suelen tratarse. (Elfatah, Zaky & Gharib, 2019)

De una forma análoga se puede referir a lo dicho por Castañeda, Nieto & Ortiz (2013) que afirman que los métodos que utilizan mediciones en los procesos para la estimación de estados desconocidos, generalmente no toman en cuenta el ruido y su naturaleza aleatoria, por ejemplo, en un proceso de rastreo de gráficos computacionales, la fuente de información nunca cambia, lo cual no sucede así con las exigencias de información de rastreo la cuales serán diferentes dependiendo del método o aplicación que se les dé. Además, que la determinación de la posición real siempre estará dada por señales ruidosas, lo cual equivale a que deberá ser modelado estadísticamente a través de ecuaciones estocásticas.

De tal forma que el carácter estocástico de las ecuaciones diferenciales estocásticas las caracteriza y existen colecciones infinitas arbitrarias de variables aleatorias, así estará indexada por la variable t que representa el tiempo, un X_t será el estado del sistema al tiempo t , para definirla finalmente como $\{X(t), \in T\}$, donde $X(t)$ es la variable aleatoria en el tiempo t y T es un conjunto o intervalo de donde t toma valores. (Elfatah et al., 2019)

Los operadores de divergencia (∇) pueden ser correspondientes a las ecuaciones diferenciales estocásticas y su simulación numérica mediante la (ecuación 1). Donde L representa en un proceso al generador infinitesimal $(X_t)_{0 \leq t \leq T}$ es decir, cuando el estado toma valores en el intervalo $[0, T]$ y a, b y ρ viene a ser coeficientes de discontinuidad acotados.

$$L = \frac{\rho}{2} \nabla \cdot (a \nabla) + b \cdot \nabla.$$

Ecuación 1. Proceso infinitesimal

Fuente (Benabdallah & Bourza, 2018)

3.1.2 DETERMINISTA

A diferencia de los modelos estocásticos, los deterministas no se ven afectados por eventos aleatorios, al contrario, la relación predominante es de uno a uno, es decir que para el seguimiento de un estado actual es necesario la restricción de los eventos posteriores, haciendo de esta manera al sistema un tanto predecible. (Elfatah et al., 2019)

Para entender este modelo se estudiará la aplicación en crecimiento poblacional expuesto por Mehta et al. (2020) donde afirman que los modelos deterministas están en ventaja por ser altamente investigados de manera científica por el uso de ecuaciones diferenciales relativamente fáciles, donde los parámetros del sistema en cuestión se encargan de estimar la solución de los factores dependientes, así se puede dirigir la distribución geográfica de enfermedades en una determinada población.

El modelo del crecimiento poblacional se definirá si el tamaño de su población en un tiempo t está dado por $N(t)$, su tasa de mortalidad por d , la tasa de natalidad por b y los intervalos de tiempo por $(t+\Delta t)$, esta se representa en la (ecuación 2), pero si se estima al conjunto N como un número muy grande, la solución real del problema estará dada por la (ecuación 3), donde N_0 es el tamaño inicial de la población, a' representa la tasa de incremento

natural y tomará distintos valores y de esto dependerá el caso a estudiar, se distinguen tres casos:

- 1) Si $a > 0$ es decir, que toma valores positivos, se tendrá una gráfica creciente según incrementa el tiempo.
- 2) Si $a < 0$ o que toma valores negativos, la gráfica decrecerá según el tiempo avance.
- 3) Si $a = 0$ se observará un comportamiento constante en el valor de la población inicial.

$$\frac{dN}{dt} = (b - d) = aN$$

Ecuación 2. Modelo de crecimiento poblacional

$$N = N_0 e^{at}$$

Ecuación 3. Crecimiento poblacional exponencial

Fuente (Mehta et al., 2020)

De la misma manera si se tomaran variables aleatorias de la cantidad de nacimientos y muertes, donde los decesos sean mayores a los nacimientos, los decesos sean menores a los nacimientos o tanto decesos como nacimientos sean iguales, cada caso tenderá a diferentes valores y no será fácil predecir el crecimiento poblacional, por lo que se tendría que hacer uso de un modelo probabilístico o estocástico para acercarse a la respuesta. En el estudio anterior se toman ambos modelos (determinista y estocástico) como complementarios ya que, si bien se puede determinar una parte mediante el primero, también se hace uso del segundo para la parte de aleatoriedad, este ejemplo se hace para la población de India, pero según los resultados no es posible obtener datos futuros para inmigración en el modelo determinista por la poca información con la que se cuenta.

Tabla 1. Resumen de modelo estocástico y determinístico

Aspecto	Estocástico	Determinista
Tipo de eventos	Aleatorios	Entradas directamente relacionadas con las salidas
Procesos de tipo	Irreversible	Reversible
Formulas	Ecuaciones diferenciales estocásticas (complejidad)	Ecuaciones diferenciales típicas.
Comportamiento	Impredecible	Conocido o predecible.
Relación	Pueden ser modelos complementarios cuando un proceso llegue a un punto de no equilibrio por algunos parámetros adicionales, por ejemplo, con el crecimiento poblacional.	

Fuente elaboración propia

3.2 ROBOTS MÓVILES CON RUEDAS

En la actualidad por su fácil operación en los distintos espacios de trabajo, los robots móviles con ruedas (WMR) son foco de muchos estudios científicos y aplicaciones industriales, fáciles de maniobrar y aptos para diferentes entornos. (Mao, Gao & Guo 2019)

Existen varios tipos de robot móviles con ruedas, pero dentro de los más utilizados por la robótica se encuentran los de transmisión diferencial, los Ackermann y los de transmisión omnidireccional, según el tipo también se dividen en modelos con diferentes números de ruedas, pero el principio de funcionamiento es el mismo. Uno de los problemas a los que cada tipo de robot móvil se ve expuesto es la creciente complejidad en sus áreas de trabajo, así como la variedad de tareas a las cuales son encomendados, por ello es preciso conocer su funcionamiento a partir de los denominados diagramas cinemáticos que muestran las medidas, velocidades y otras especificaciones para la movilidad del robot y en algunos casos que se requieran se pueden estudiar incluso los principios dinámicos de estos robots.



Figura 1. Nuevos entornos de trabajo de robots móviles

Fuente (Morales, 2021)

3.2.1 TRANSMISIÓN DIFERENCIAL

Una estructura simplificada, flexibilidad en el movimiento y costos relativamente bajos en términos de producción han hecho que los robots diferenciales resalten más en aplicaciones de la robótica móvil, sin mencionar que tiene la capacidad de funcionamiento autónomo en periodos de tiempo relativamente amplios sin la acción directa del humano. Estos robots se

encuentran en muchas presentaciones dependiendo de la cantidad de ruedas, desde 2 ruedas con una tercera de apoyo de tipo pasivo, de 3 ruedas y de 4 ruedas siendo el más utilizado el de 2 ruedas. Mediante el cambio de velocidad de las ruedas que, aunque comparten un eje a una distancia determinada son independientes de la otra, en otras palabras, que funcionan por separado se define el funcionamiento de los robots diferenciales para el cambio de dirección de este. Sin duda que el seguimiento y planeación son problemas de suma importancia en estos robots, pero debido a que la operatividad del robot y su precisión pueden verse afectados de forma directa, el seguimiento de trayectoria está en un grado de importancia más alto. (Stefek, Pham T., Krivanek, Pham, 2020)

Antes de conocer el modelo cinemático de este tipo de robot es necesario apuntar que existen restricciones para la transmisión diferencial, generalmente estas son de tipo no holonómicas, es decir que los grados de libertad del robot son mayores a los grados de libertad que se pueden controlar. (Mao et al., 2019)

Modelo cinemático

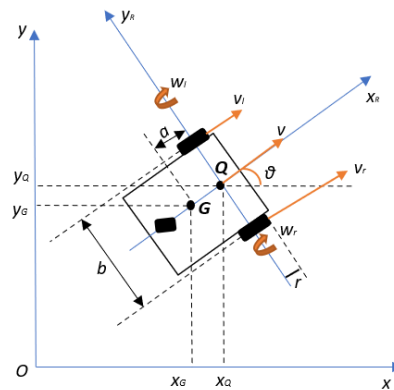


Figura 2. Geometría de robot diferencial de 3 ruedas

Fuente (Stefek et al., 2020)

Donde:

v =velocidad lineal

θ =orientación

ω_r =velocidad angular de la rueda derecha

ω_l =velocidad angular de la rueda izquierda

v_r =velocidad lineal de la rueda derecha

v_l =velocidad lineal de la rueda izquierda

r =radio de las ruedas
 b =distancia entre las ruedas
 Q =centro del eje entre ruedas
 G =centro de gravedad
 a =distancia entre Q y G

Stefek et al. (2020) afirma que:

Un tema importante de la conducción diferencial de los robots móviles que debe tenerse en cuenta es que el diseño de su controlador de movimiento se basa principalmente en modelos cinemáticos. La razón es porque los modelos dinámicos son más complejos que los modelos cinemáticos y los robots móviles generalmente usan solo la baja velocidad del motor para controlar el lazo. (p. 2)

Para el modelado cinemático se asumirá que, G y Q son coincidentes, que el plano del robot es perpendicular al eje guía y que las llantas estarán rodando sin llegar a patinar lo que daría que $v_r=r\omega_r$ y $v_l=r\omega_l$. Donde X_G y X_Q son los valores que G y Q respectivamente toman del eje x , y Y_G e Y_Q los que se toman del eje y .

$$X_G = X_Q - a \cos \theta$$

Ecuación 4. Valor de G en el eje x

$$Y_G = Y_Q - a \sin \theta$$

Ecuación 5. Valor de G en el eje y

Las ecuaciones 6, 7 y 8 representan el conjunto de ecuaciones que forman parte del modelo cinemático de robot diferenciales según la geometría propuesta en la (figura 2).

$$\dot{x} = \dot{x}_Q = v \cos \theta$$

Ecuación 6. Derivada de x

$$\dot{y} = \dot{y}_Q = v \sin \theta$$

Ecuación 7. Derivada de y

$$\dot{\theta} = \omega$$

Ecuación 8. Derivada de la orientación

Fuente (Stefek et al., 2020)

3.2.2 ACKERMANN

Para entender el funcionamiento de los robots de tipo Ackermann asociados generalmente a 4 ruedas similar al modelo de un automóvil se presenta en el estudio en el que Romero N. & Tuiran V. (2018) afirman que para omitir el uso de complejas ecuaciones derivadas

se puede obtener un método de optimización de un mecanismo específicamente de 6 barras a partir de la utilización de un algoritmo genético y coordenadas naturales para un mejor modelo cinemático del mecanismo en un robot tipo Ackermann, aunque el robot como tal no es el objeto de estudio si no que el mecanismo visto desde los laterales, este favorece la estimación y su cinemática ya que brinda estabilidad y dirección. En la (ecuación 9) se representa la geometría de giro de un robot tipo Ackermann, donde el eje delantero se encarga de la orientación mientras que el trasero se alinea en un punto común, δ_i y δ_o representan a los ángulos de dirección de las ruedas los cuales se pueden observar en la (figura 3), "l" representa la distancia entre las ruedas delanteras y traseras y "w" la distancia entre las ruedas en un mismo eje.

$$\text{Cot } \delta_i - \text{Cot } \delta_o = \frac{w}{l}$$

Ecuación 9. Giro de un robot tipo Ackermann

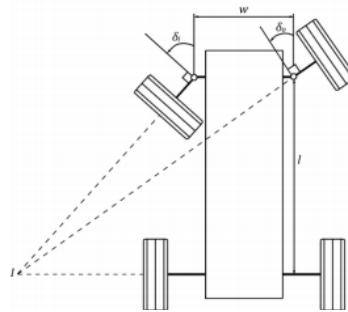


Figura 3. Geometría del giro de robot tipo Ackermann

Fuente (Romero N. & Tuiran V., 2018)

3.2.3 OMNIDIRECCIONAL

Un estudio realizado en el año 2018 por Nacer Hacene y Boubekeur Medil hace una compilación de muchos artículos relacionados a los robots omnidireccionales, en el mismo ellos pretenden brindar una visión general y complementaria a lo no abordado en esos estudios, tomando como referencia un robot omnidireccional de 3 ruedas toman en cuenta todas posibles velocidades en ambas direcciones del robot para poder calcular así su movimiento. A continuación, se presenta el diagrama utilizado en el estudio, pero solo se incluirán las fórmulas cinemáticas directas y no todo el estudio documentado para posteriormente comentar el resultado.

Modelo cinemático

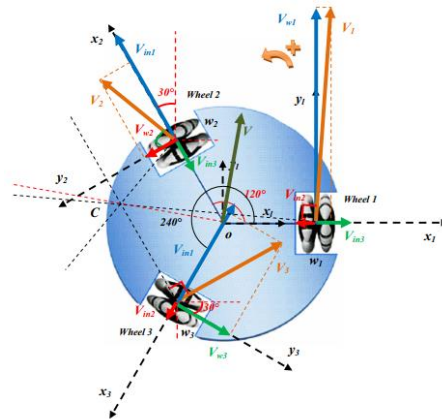


Figura 4. Diagrama de velocidades de robot omnidireccional de 3 ruedas

Fuente (Hacene & Mendil, 2019)

Donde:

V_w =velocidad lineal

V_{in} =velocidad inducida

V =velocidad resultante

R = radio de rotación del robot

L = radio del robot

o = centro de masa

C = punto de rotación

θ = dirección

$$V_x = \dot{x} = -\frac{\sqrt{3}}{3}V_{w2} + \frac{\sqrt{3}}{3}V_{w3}$$

$$V_y = \dot{y} = \frac{2}{3}V_{w1} - \frac{1}{3}V_{w2} - \frac{1}{3}V_{w3}$$

$$\dot{\phi} = \frac{1}{3L}V_{w1} + \frac{1}{3L}V_{w2} + \frac{1}{3L}V_{w3}$$

Ecuación 10. Ecuaciones cinemáticas del robot omnidireccional

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & -\sqrt{3}/3 & \sqrt{3}/3 \\ 2/3 & -1/3 & -1/3 \\ 1/3L & 1/3L & 1/3L \end{bmatrix} \begin{bmatrix} V_{w1} \\ V_{w2} \\ V_{w3} \end{bmatrix}$$

Ecuación 11. Ecuaciones cinemáticas expresadas en matriz

$$\theta = \text{atan 2}(V_y, V_x)$$

Ecuación 12. Ángulo de dirección

Fuente (Hacene & Mendil, 2019)

En el estudio para el problema de la trayectoria del robot se implementó un controlador difuso ya que se distingue 3 movimientos distintos que implican el movimiento de traslación normal, rotación alrededor del centro de masa y traslación alrededor de otro punto que no

fuese el centro de masa, mediante la implementación y diseño de este controlador se observaban resultados favorables en las simulaciones de cada uno de los movimientos, brindando flexibilidad y un control más simple para este tipo de robot. (Hacene & Mendil, 2019)

3.3 ALGORITMOS PARA TRAYECTORIA DE ROBOT MÓVILES

Para el desarrollo de robot autónomos se ha hecho necesaria la implementación de algoritmos de trazado de trayectoria, pues las condiciones del entorno pueden afectar las mediciones de los sensores y no lograr el objetivo para el cual se desarrolló el robot y aunque existe una gran cantidad de métodos para estimar la trayectoria de un robot, se presentaran 5 de los más utilizados en la actualidad de los cuales se ha hecho investigaciones que comparan y validan el funcionamiento de estos con el de otros.

3.3.1 TÉCNICA DE A*

Por su fácil codificación la técnica de A* brinda soluciones a los problemas comunes de planificación considerándose dentro de los métodos completos para la evaluación rápida, su punto de estudio es la posición inicial del objeto desde donde encontrara la trayectoria para llegar a otro punto deseado o final, no sin antes detallar toda la información de la escena, como los espacios ocupados o despejados facilitando así la elección de la mejor ruta entre los 2 puntos. (Solaque, Velasco & Riveros, 2018)

Para poder estimar la trayectoria o ruta más conveniente que pasa por los nodos la técnica A* utiliza la denominada estimación heurística representada por $h(x)$ proporcionando soluciones rápidas.

Pasos para técnica A:*

- 1) Con excepción del punto inicial se dispone la distancia entre los puntos al infinito.
- 2) Se constituyen como nodos disponibles todos los puntos incluyendo el inicial
- 3) El nodo actual "C" se establece vinculando la distancia actual más corta al nodo no visitado.
- 4) Si el valor que resulta que a cada punto cercano al nodo actual "N" se le añade el peso del punto final más el peso entre "C" - "N" con la distancia actual "C" es menor a la distancia actual de "N", se determinara como la nueva distancia actual "N".
- 5) Señalar como visitado el nodo actual "C".

6) Reiterar cada paso desde el 3) hasta que uno de los puntos cercanos "N" se convierta en el punto final. (Rachmawati & Gustin, 2020)

En el estudio de la técnica A* realizado en 2018 por L.E Solaque, A.E Velasco y A. Rivero se brinda una ecuación para la estimación de la ruta del efector final de un robot agricultor para la remoción de maleza. Donde el costo real de avance esta dado por $f(n)$, el desplazamiento del nodo actual al siguiente esta dado por $g(n)$ y $h(n)$ brinda la información de la distancia que falta por recorrer hasta el punto final es decir la estimación heurística. La aplicación de la técnica A* muestra buenos resultados porque además también se aplicó un suavizador de curvas para no tener discontinuidades en la ruta.

$$f(n) = g(n) + h(n)$$

Ecuación 13. Costo real de avance

Fuente (Solaque et al., 2018)

3.3.2 ARBOLES ALEATORIOS DE EXPLORACIÓN RÁPIDA O RRT

El RRT se ha convertido en un algoritmo muy utilizado en la trayectoria de robots móviles de varios grados de libertad, es considerado probabilísticamente completo, eficiente y rápido, además que se puede ubicar en áreas totalmente desconocida a partir de información del entorno, tampoco necesita de un procesamiento previo para la planificación de trayectorias, aunque algunos problemas de búsqueda, aleatoriedad y respuesta final en su diseño original han llevado al desarrollo de mejoras al algoritmo el cual cuenta con varias versiones cada una aplicada a optimizar el objetivo inicial del RRT. (Ren, Tan, Shi & Lian, 2021)

Tabla 2. Algoritmo RRT tradicional

Algoritmo RRT
1: $T.V \leftarrow x_{init};$
2: while <i>NotReachStop</i> do
3: $x_{rand} \leftarrow Sample_free;$
4: $x_{nearest} \leftarrow NearestNode(x_{rand}, T.V);$
5: $x_{new} \leftarrow Steer(x_{nearest}, x_{rand});$
6: if <i>Collision_check</i> ($x_{nearest}, x_{new}$) == <i>True</i> then
7: $T.V \leftarrow U x_{new};$
8: end if
9: end while

Fuente (Chen, Shan, Tian, Li & Cao, 2018)

Donde:

x_{init} = estado inicial

Sample_free= Devuelve un estado aleatorio con configuración libre

x_{rand} = estado aleatorio

NearestNode= Devuelve el nodo más cercano

Steer= Sirve para conducir de un punto a otro punto nuevo.

Collision_check= devuelve el valor verdadero si la función es factible

NotReachStop= Si los criterios de parada no son suficientes para iterar devuelve un valor verdadero.

T, V = nodos del árbol T

En ambos artículos citados se hace un estudio comparativo entre las diferentes variantes del algoritmo RRT, Chen et al. (2018) introduce una nueva variación del algoritmo denominado DT-RRT* al tiempo que la compara con las últimas actualizaciones del algoritmo, básicamente el DT-RRT* después de encontrar la solución se optimiza el proceso mediante el uso de uno de los árboles en tanto que el otro se usó anteriormente para la exploración del espacio, se menciona que incluso este método puede llegar a mejorar, pero que aun así los resultados han sido satisfactorios ya que se ahorran recursos, mejor eficiencia y suavidad de la ruta.

En tanto que Ren et al. (2021) compara el tradicional RRT con 3 de sus actualizaciones incluyendo el RRT mejorado que es su algoritmo de estudio principal y menciona que bajo limitaciones dinámicas el algoritmo RRT mejorado es capaz de planificar rutas de multiobjetivo óptimas, el tiempo se puede acortar agregando más nodos en el proceso, además de estrategias de paso dinámico, de búsqueda heurística entre otros, con el que se trata de reducir el problema generado por el RRT para optimizar la planificación de rutas debido a la aleatoriedad característica del proceso.

3.3.3 ALGORITMO DE MONTE CARLO

Para que la estimación de la filtración sea óptima se representa Monte Carlo mediante una función de probabilidad posterior de la densidad, que es análogo a que las muestras se aproximen a infinito, es decir, que se toman muestras aleatorias para ponderar la densidad a posteriori y así obtener la estimación más óptima posible, para poder establecer un filtrado del tipo bayesiano la simulación no debe estar parametrizada, en otras palabras, una muestra se encargará de representar la densidad de probabilidad en lugar de una función. (Du, Yu, Gay & Jian, 2018)

Para el cálculo de iteraciones se utiliza la (ecuación 14), esta dependerá de la confianza y exactitud que se escojan para que la respuesta sea favorable, es decir, que se puede reducir la probabilidad de que el mínimo no sea encontrado a partir de las veces que el algoritmo sea repetido aleatoriamente de forma independiente en cada tiempo. Donde la exactitud viene dada por “ ϵ ” y la confianza por “ δ ”. (Fernández, Pantano, Godoy & Scaglia, 2019)

Pasos del Algoritmo de Monte Carlo:

- 1) Determinación de parámetros a optimizar.
- 2) Definir la cantidad de simulaciones “N”, donde los valores de confianza y exactitud dependerán de precisión que se quiera.
- 3) Asignación de valores de los parámetros a estimar.
- 4) Cálculo de J mediante la simulación del proceso
- 5) Reiteración de los dos pasos anteriores 3) y 4) hasta completar las “N” establecidas
- 6) Selección de parámetros que reduzcan el J.

$$N \geq \left\lceil \frac{\log \frac{1}{\delta}}{\log \frac{1}{1-\epsilon}} \right\rceil$$

Ecuación 14. Número de simulaciones del algoritmo

Fuente (Fernández et al., 2019)

Sin duda que el algoritmo de Monte Carlo es muy utilizado en sistemas de aleatoriedad, en el estudio para analizar la capacidad no lineal de procesamiento del algoritmo, Du et al. (2018) hace una comparación de este y el filtro de Kalman extendido (EKF) donde menciona la aplicación de un algoritmo extra junto al de Monte Carlo para mejorar el procesamiento no lineal, el argumento para esta utilización es que el EKF para trabajar con procesos no lineales utiliza modelos lineales que funcionan muy bien excepto cuando el sistema es altamente no lineal donde se presentan fallas o variaciones, por otro lado el algoritmo de Monte Carlo genera resultados más favorables gracias a la implementación del algoritmo extra.

3.3.4 ALGORITMO DE DIJKSTRA

Para dividir un entorno generalmente se hace uso del método de cuadrícula estática la cual es utilizada por el algoritmo de Dijkstra para estimar la mejor trayectoria, el modelo se ve

afectado de forma positiva o negativa en términos de complejidad y almacenamiento de los datos del entorno por la proporción de la rejilla, en la cuadrícula utilizada en el estudio los espacios ocupados son señalados por el color negro y en blanco los que están libres. (Sun, Fang & Su, 2021)

Pasos para algoritmo de Dijkstra según Rachmawati & Gustin (2020):

- 1) Con excepción del punto inicial se dispone la distancia entre los puntos al infinito.
- 2) Se constituyen como nodos disponibles todos los puntos incluyendo el de partida.
- 3) El nodo actual "C" se establece vinculando la distancia actual más corta al nodo no visitado.
- 4) Se debe añadir el peso entre "C" – "N" junto a la distancia actual de "C" para cada punto cercano "N" al nodo actual, se determina como la nueva distancia actual si el resultado el inferir a la distancia actual "N".
- 5) Señalar como visitado el nodo actual "C".
- 6) Reiterar cada paso desde el 3) hasta que se llegue al punto final.

Uno de los problemas que vehículos de guía automática tiene normalmente es el planear las rutas de seguimiento, un estudio enfoca a estos vehículos conocidos como AGV por sus siglas en ingles es el de Sun et al. (2021) afirma que este problemas se puede evitar a partir de una versión mejorada del algoritmo de Dijkstra que busca ocho ángulos a diferencia del original que solo trabaja con la búsqueda de cuatro los cuales pueden aumentar la redundancia porque se limita el ángulo de dirección, cabe decir que el estudio se basa en la simulación de los algoritmos mediante el programa de Matlab.

Por otro lado en un estudio comparativo de algoritmo Dijkstra y A*, Rachmawati & Gustin (2020) menciona que ambos algoritmos pueden comportarse de forma similar aunque el método A* difiere y tiene ventaja del Dijkstra en su estimación heurística que la caracteriza y solo busca la ruta más corta a su objetivo en tanto el Dijkstra puede ser redundante y analizar otras partes antes de llegar al punto final cuando se trabaja a gran escala, pero si el entorno es mejor como en el estudio ambos algoritmos ofrecen el mismo resultado.

3.3.5 ALGORITMO DE KEVIN PASSINO

Este no es más que la formulación hecha en 2005 por Kevin Passino de su algoritmo de optimización basado en forrajeo de bacterias aplicado para la generación de trayectorias en robots móviles donde lo principal es el proceso de dispersión después de converger con un mínimo local u obstáculo para lograr el escape o evasión de este en otras palabras existe una función que se asocia tanto a los obstáculos como al objetivo, los primeros a partir de campos potenciales y el segundo por su distancia euclidiana. (Cüchango & Esmeral, 2014)

Este algoritmo inicialmente parte de la idea de selección natural, es decir que se pueden omitir aquellas partes del sistema que tienden a tener un menor desarrollo en localización y otras actividades de los individuos, dándole prioridad a aquellos dominantes que, si cumplen con un buen desarrollo, esto también se aplica a elementos grupales o colonias que ejecutan sus actividades a partir de la comunicación entre sus miembros. (Noriega et al., 2010)

$$J(x, y) = \omega_1 J_o(x, y) + \omega_2 J_g(x, y)$$

Ecuación 15. Función de costo total

Fuente (Passino, 2005)

Donde $J(x, y)$ representa la función de costo total del sistema y está dada por la suma de la multiplicación de las matrices de costo y el peso tanto de los obstáculos como del punto meta, ω_1 y ω_2 son los pesos de los obstáculos y el objetivo respectivamente, $J_o(x, y)$ es la función de costo de los obstáculos, $J_g(x, y)$ es la función de costo de la meta.

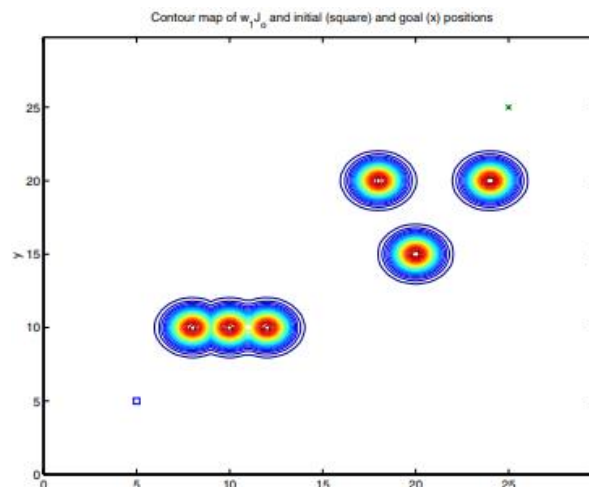


Figura 5. Función obstáculos y contorno de las formas

Fuente (Passino, 2005)

En la (figura 5) se observan la función obstáculo donde cada uno de ellos está dado por el contorno de su forma y la función meta está dada por la equis verde y la salida o posición inicial por un cuadrado azul.

3.3.6 FILTRO DE KALMAN

Rudolf E. Kalman húngaro nacido en Budapest el 19 de mayo de 1930 y que falleció el 2 de julio de 2016 a sus 86 años en Zúrich, Suiza fue un reconocido matemático, ingeniero eléctrico, industrial y de sistemas, profesor de investigación en el Instituto para Estudios Avanzados (RIAS), fundador y director del Centro de Teoría de Sistemas Matemáticos (CMST), miembro de la Academia Nacional de ingeniería, de la Academia Nacional de Ciencias y la Academia Estadounidense de Artes y Ciencias todas estas en EE. UU. Acreedor de numerosos doctorados y premios por sus importantes aportes a la ciencia e ingeniería. (Sontag, 2010)



Figura 6. Rudolf Emil Kalman

Fuente (Antoulas et al., 2017)

Desde sus inicios e incluso antes de presentar su tesis doctoral Kalman revolucionó a la ingeniería de control y sus ideas y aportes de investigaciones fueron base del campo moderno de la teoría de sistemas matemáticos, sin duda el más destacado aporte que hizo fue el ahora conocido como filtro de Kalman de 1960, el cual permanece en un sinnúmero de aplicaciones de los avances tecnológicos presentes. (Georgiou, 2018)

Sus conocimientos sobre la estructura de los sistemas dinámicos y, más específicamente, su trabajo fundamental sobre la teoría del espacio de estados y los conceptos de controlabilidad y observabilidad marcaron un cambio de paradigma que allanó el camino para las metodologías de principios para el diseño de controladores como la síntesis de elementos lineales. (Antoulas et al., 2017)

El filtro de Kalman es un algoritmo de naturaleza recursiva, que no necesita una muestra de cada momento en el pasado para poder estimar la salida o estado posterior del sistema, al contrario, solo requiere de una que cuenta con la suficiente información como para determinar la solución deseada, trabaja en dos secciones una de predicción y otra de corrección, su origen está en un modelo de espacio de estados lineal, pudiéndose comportar como su nombre lo dice de filtro (generalmente del ruido o la señal del sistema que siempre viene con perturbaciones no deseadas) o de estimador de parámetros, debido a que los eventos que lo acompañan son aleatorios es esencialmente probabilístico y su aplicación es enorme en campos como economía, procesamiento de señal, demografía, navegación y otros, por lo tanto su importancia se hace cada vez más notable en la comunidad científica, la cual se ha encargado de seguir su comportamiento y aplicaciones a otros sectores para utilizarlo como herramienta de estimación de resultados favorables. (Castañeda C. et al., 2013)

3.3.6.1 ESPACIO DE ESTADOS LINEAL

“Los modelos de espacio de estados son esencialmente una noción conveniente para estimación y control, desarrollados para hacer trazable lo que sería de otra manera un análisis intrazable” (Castañeda C. et al., 2013, p. 268).

$$\begin{aligned} X_{t+1} &= f(X_t, u_t, w_t, \theta) \\ Y_t &= g(X_t, e_t, \theta) \end{aligned}$$

Ecuación 16. Modelo de espacio de estados

$$\begin{aligned} X_{t+1} &= f(AX_t + Bu_t + w_t) \\ Y_t &= CX_t + e_t \end{aligned}$$

Ecuación 17. Modelo de espacio de estados simplificado de forma lineal

Fuente (Zhang, 2018)

Tabla 3. Parámetros de espacio de estados

Parámetro	Descripción
$Y_t \in R_p$	Datos del sensor
$X_t \in R_m$	Estado del sistema
$u_t \in R_m$	Entrada del sistema
$A \in R_{m \times m}$	Matriz de estado
$B \in R_{m \times m}$	Matriz de entrada
$C \in R_{p \times m}$	Matriz de medición
$Q \in R_{m \times m}$	Matriz de covarianza de ruido del proceso
$R \in R_{p \times p}$	Matriz de covarianza para el ruido que se mide
w_t	Ruido del sistema
e_t	Ruido de medición
θ	parámetros del modelo

Fuente Elaboración propia a partir de (Zhang, 2018)

3.3.6.2 FUNCIONAMIENTO DEL FILTRO DE KALMAN

El funcionamiento del filtro de Kalman se basa en dos importantes secciones que se mencionaron anteriormente, las cuales son la etapa de predicción donde ocurre una actualización de tiempo y la etapa de corrección donde se podrán actualizar los datos medidos, básicamente el filtro de Kalman se divide en 5 pasos o ecuaciones necesarias, donde la primera etapa abarca los 2 primeros pasos y el resto son del lado correctivo. Donde se observa muchos de los parámetros de espacio de estado enumerados anteriormente.

Pasos del filtro de Kalman

Predicción

- 1) Se debe proyectar el estado hacia delante, donde k-1 representa el paso anterior en el tiempo k.

$$\hat{x}_k^- = A\hat{x}_{k-1} + B\mu_k$$

Ecuación 18. Estado estimado a priori

- 2) Proyectar hacia delante la covarianza. Donde P_k^- representa la covarianza del error hacia delante o a priori.

$$P_k^- = AP_{k-1}A^T + Q$$

Ecuación 19. Covarianza del error a priori

Corrección

- 3) Cálculo de la ganancia de Kalman (K_k), donde H relaciona el estado con la medición, R es la covarianza del error de medición

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

Ecuación 20. Ganancia de Kalman

- 4) Actualización del estimado con z_k o medida real

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-)$$

Ecuación 21. Actualización del estado con z_k

- 5) Actualización de la covarianza del error

$$P_k = (I - K_k H) P_k^-$$

Ecuación 22. Covarianza del error a posteriori

Fuente (Castañeda C. et al., 2013)

Tabla 4. Cuadro comparativo de algoritmos

Algoritmo	Ventaja	Desventaja
Método A*	Consta de una estimación heurística.	Objetos bastante cerca de la estimación de ruta.
RRT	No necesita de un procesamiento previo y es rápido	Problemas de aleatoriedad y búsqueda por lo que no siempre encuentra el camino más corto
Monte Carlo	Convergencia con otros algoritmos para suavización de curvas.	Soluciones no tan optimas, para mejorar respuestas se usan otros algoritmos.
Dijkstra	Optimización de ruta	Limitación del ángulo de dirección causando divergencia.
Filtro de Kalman	Modelo predictivo - correctivo (solo necesita un punto del pasado para la estimación)	En su forma tradicional no es apto para procesos no lineales.

Fuente elaboración propia

3.4 VARIACIONES DEL FILTRO DE KALMAN

En la sección anterior se introdujeron las ecuaciones para un espacio de estados lineal, pero la mayoría de sistemas se comportan de forma no lineal y el filtro de Kalman tradicionalmente es exclusivo para sistemas lineales por lo que utilizarlo sería prácticamente inapropiado y sin sentido, para ello el algoritmo ha sido sometido a estudios a través de los años para poder ser utilizado tanto en sistemas lineales como no lineales y se han generado

algunas variaciones de este con mejoras, pero con el mismo principio de predicción- corrección y han sido aplicadas en muchas áreas que se estarán viendo más adelante, a continuación se presentan algunas de las actualizaciones del filtro, no sin antes introducir la forma de las ecuaciones de los sistemas no lineales.

Espacio de estados no lineal

$$\begin{aligned}x_k &= f(x_{k-1}, u_{k-1}) + W_{k-1} \\z_k &= h(x_k) + V_k\end{aligned}$$

Ecuación 23. Ecuaciones generales de un sistema no lineal

Fuente (Anbu & Jayaprasanth, 2019)

Donde:

W_k y V_k = representan gaussiana del ruido

f y h = representación no lineal

x_k = vector de estado

z_k = señal de medición

u_k = entrada conocida

3.4.1 FILTRO DE KALMAN EXTENDIDO

Existe un tipo de función de densidad de probabilidad más conocida como PDF que en el filtro de Kalman extendido (EKF) se puede llegar a aproximar como de tipo gaussiana después de que mediante la utilización de la serie de expansión de Taylor, el EKF linealice el sistema no lineal que se quiere resolver, en otras palabras el EKF funciona igual que el filtro de Kalman tradicional con 2 etapas de estimación con la diferencia de que debido a que son sistemas no lineales el EKF linealiza el sistema en torno a una trayectoria de estado, siendo este su enfoque para la estimación de resultados. (Cevallos, Intriago, Plaza & Idrovo, 2018)

También el EKF ha servido de inspiración o como base de otros algoritmos como, por ejemplo, uno de estimación de momento inercial en satélites donde se utiliza el principio del EKF tomando medidas del giroscopio para posteriormente acoplar el algoritmo de Monte Carlo a fin de la validación final. (Bellar & Mohammed, 2019)

Algoritmo del EKF

$$\begin{aligned}x_t^f &= f_{t,t-1}(x_{t-1}^a) \\P_t^f &= F_{t,t-1}P_{t-1}^aF_{t,t-1}^T + Q_t\end{aligned}$$

Ecuación 24. Etapa predictiva del EKF

$$K_t = P_t^f H_t^T (H_t P_t^f H_t^T + R_t)^{-1}$$

$$x_t^a = x_t^f + K_t (z_t - h_t(x_t^f))$$

$$P_t^a = P_t^f - K_t H_t P_t^f$$

Ecuación 25. Etapa correctiva del EKF

Fuente (Cevallos U. et al., 2018)

Si se observa bien el algoritmo del EKF es igual al del filtro tradicional con diferencia en algunas matrices o variables, por ejemplo x_{t-1}^a representa al vector de estado, P_{t-1}^a es la matriz de covarianza de error, H_t es matriz jacobiana de la etapa predictiva propia de la linealización de estados no lineales.

3.4.2 FILTRO DE KALMAN SIN PERFUME

Otra de las actualizaciones del filtro de Kalman es el filtro de Kalman sin perfume (UKF) generado después del EKF a fin de disminuir los problemas que tiene este al linealizar los sistemas de naturaleza no lineal, el UKF también hace uso de una PDF, pero en lugar de linealizar el sistema no lineal, trabaja con la densidad de probabilidad haciendo una aproximación de esta. (Viveros, Yuz & Pérez, 2014)

El UKF está diseñado para sistemas no lineales, pero su enfoque no incluye una matriz jacobiana como en el EKF, si no que utiliza la transformación sin perfume (UT) para tomar muchos puntos de muestra conocidos como puntos de sigma y mapear transformando de forma no lineal cada uno de los puntos muestrales para obtener una PDF de aproximación, además de comparar la covarianza y la media original a fin de disminuir los errores en la predicción. (Deng, Zeng & Zhou, 2020)

Básicamente el UKF utiliza las mismas etapas de trabajo que el filtro de Kalman original con la diferencia que antes de ello se los puntos sigma de muestra.

Algoritmo de UKF

Sacar puntos sigma

$$X^{(i)} = \bar{X} + \sqrt{P} \sigma_i^{(n)} \quad i = 0, \dots, n + 1$$

Ecuación 26. Matriz para puntos sigma

Predicción

$$\hat{x}_k^- = \sum_{i=1}^{2n} w_i \hat{x}_k^{(i)}$$

Ecuación 27. Estado estimado a priori UKF

$$P_k^- = \sum_{i=1}^{2n} w_i (\hat{x}_k^{(i)} - \hat{x}_k^-)(\hat{x}_k^{(i)} - \hat{x}_k^-)^T + Q_{K-1}$$

Ecuación 28. Covarianza de error a priori UKF

Corrección

$$\hat{y}_k = \sum_{i=1}^{2n} w_i \hat{y}_k^{(i)}$$

Ecuación 29. Media a posterior UKF

$$\hat{P}_y = \sum_{i=0}^{2n} w_i (\hat{y}_k^{(i)} - \hat{y}_k)(\hat{y}_k^{(i)} - \hat{y}_k)^T + R_K$$

Ecuación 30. Covarianza a posteriori UKF

Fuente (Darvishi & Doroudi, 2019)

El cálculo de la ganancia no se encuentra en el paso 3 como en el original pues la estimación del estado y la covarianza no dependen de ella. Pero de igual forma se puede calcular a partir de la covarianza entre los dos estados a priori y a posteriori.

$$P_{xy} = \sum_{i=1}^{2n} w_i (\hat{x}_k^{(i)} - \hat{x}_k)(\hat{y}_k^{(i)} - \hat{y}_k)^T$$

Ecuación 31. Covarianza entre los dos estados

$$K_k = P_{xy} P_y^-$$

Ecuación 32. Ganancia de UKF

Fuente (Darvishi & Doroudi, 2019)

3.4.3 FILTRO DE KALMAN DE CUBICACIÓN DE RAÍZ CUADRADA

Otro método de estimación para sistemas no lineales es el filtro de Kalman de cubicación de raíz cuadrada (SRCLF) que presenta características favorables de estabilidad y aproximación, este surge después de EKF y UKF como respuesta a algunos problemas de ambos, puede hacer un cálculo directo en comparación al EKF ya que no necesita linealizar el sistema y al igual que el UKF no necesita de la matriz jacobiana, también comparte algunos criterios de este último, pero no selecciona los parámetros de la misma forma. (Zhu, Hu, Li, Ming, Yang & Zhang, 2019)

El SRCKF es un estimador de estado óptimo basado en un mecanismo de muestreo determinista. Mas precisamente, la raíz cuadrada de la covarianza del error de predicción y la covarianza del error a posteriori se transmiten en el proceso de filtrado, lo que garantiza tanto la simetría como la definición positiva de la matriz de covarianza. (Shen, Zhang, Guo, Chen, Cao, Tang, Li & Liu, 2021)

Algoritmo de SRCKF

$$X_{i,k-1|k-1} = S_{k-1|k-1} \xi_i + \hat{x}_{k-1|k-1}, i = 1, \dots, 2n$$

Ecuación 33. Puntos de cubicación del estado

$$\hat{x}_{k|k-1} = \sum_{i=1}^{2n} w_i X_{i,k|k-1}^*$$

Ecuación 34. Valores medios de los puntos de estado

$$S_{k|k-1} = \text{Triang}([X_{k|k-1}^*, S_Q])$$

Ecuación 35. Raíz cuadrada de la covarianza a priori

$$W_k = P_{xy,k|k-1} (S_{yy,k|k-1} S_{yy,k|k-1}^T)^{-1}$$

Ecuación 36. Matriz de ganancia SRCKF

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + W_k (y_k - \hat{y}_{k|k-1})$$

Ecuación 37. Estado actualizado SRCKF

$$S_{k|k} = \text{Triang}([X_{k|k-1} - W_k \zeta_{k|k-1}, S_R])$$

Ecuación 38. Raíz cuadrada de la covarianza de error a posteriori

Fuente (Zhu et al., 2019)

Tabla 5. Parámetros desconocidos de SRCKF

Parámetro	Descripción
ξ_i	Punto de cubicación
w_i	Valor de ponderación
Triang	Se consigue por triangulación la matriz cuadrada
S_Q y S_R	raíz cuadra de la covarianza del ruido
P	Matriz de covarianza

Fuente elaboración propia

3.5 APLICACIONES DEL FILTRO DE KALMAN

Son muchas las áreas de aplicación del filtro de Kalman una de las razones es porque a diferencia de otros algoritmos de filtrado de señal, este no necesita de muchos puntos en el pasado para dar un resultado preciso, solamente necesita uno, ya que al ser recursivo ese punto

guarda toda la información necesaria para predecir y luego corregir el valor, a continuación se conocerán algunos en los campos de aplicación donde tanto se estudia el filtro en su forma original, sus variaciones conocidas y otras nuevas basadas en el mismo.

3.5.1 PREDICCIÓN DE UBICACIÓN EN VANET'S

Son muchas las aplicaciones del filtro de Kalman y sus actualizaciones, una de ellas es en las redes vehiculares Ad Hoc, a continuación, se presentan 2 estudios donde se conocerá el área de aplicación específica del filtro como los resultados de dicho estudio.

Se utiliza el filtro de Kalman extendido para la localización, la premisa del proyecto es que debido a las interferencias que ocasionan las estructuras, objetos y fenómenos ambientales muchas veces las señales GPS no captan bien la señal y no dan datos del todo correctos, como visión general la utilización de EKF en este estudio mejora en un 15 % el rendimiento en comparación cuando no se utilizó.

Uno de los problemas en las redes vehiculares Ad Hoc es que debido a obstáculos, numerosos caminos y otros fenómenos la intensidad de la señal se puede ver afectada, en el estudio para la determinación de la ubicación de los nodos se hace uso de un sistema del tipo RSSI y aun este no es del todo fiable por lo que se utiliza el EKF, en la simulación se tomó como modelos la ciudad de Manhattan y a través de trazas reales se evaluó el rendimiento de VANET, el uso de EKF se debe a su naturaleza predictiva - correctiva que le hace un método más confiable, porque no solo se pretendía conocer la ubicación sino que también corregir los errores de esa ubicación. (Al Malki et al., 2021)

Otro de los problemas en las VANETS es el descubrimiento de vehículos cercanos, en siguiente ejemplo se utiliza el filtro de Kalman tradicional como base para el desarrollo de otro algoritmo para la estimación de la variable llamado KPND. El algoritmo KPND al ser derivado del filtro de Kalman tendrá una parte donde se desarrolle dicho filtro y la otra referida al descubrimiento de cercanos, el problema con la detección de vecinos está en que la conectividad o comunicación de con estos dentro de un rango valido, como resultados el filtro de Kalman resultó ser una buena base para la predicción de cada nodo y vecino, un problema que resulta a veces en los algoritmos de descubrimiento de vehículos cercanos con la elevación

del número de nodos es una tasa de error alta, pero el KPND inclusive en este caso logra su objeto. (Liu et al., 2020)

3.5.2 MOTORES

Una de las aplicaciones en motores fue realizada específicamente para un motor turbofán para una estimación confiable del deterioro de las tuberías, donde se comparan dos las actualizaciones del filtro, el EKF, el SRCKF y una adaptación del SRCKF llamada ASRCKF, a fin de determinar cuál de los 3 muestra mejores resultados para el cuidado de la salud, es decir el control y estimación del deterioro se hace para prevenir los daños en la salud que puede ocasionar las fugas de gas en un ambiente. Según resultados obtenidos en las simulaciones se encuentra a ASRCKF como la opción que brinda mejor adaptabilidad, estimación y precisión en control y seguimiento del gas en un motor turbofán, para la estimación de su tiempo de deterioro y comportamiento. (Zhu et al., 2019)

En la (figura 7) se muestra uno de los resultados de las simulaciones donde se puede ver las 3 versiones del filtro y donde el valor RMSE representa el valor del error medio de la raíz cuadrada, pudiendo claramente ver que el EKF es el más alto en error, cabe mencionar que este no se basa en la raíz cuadra y ese podría ser la razón del porque el error sea mucho mayor a los otros dos métodos que si lo hacen.

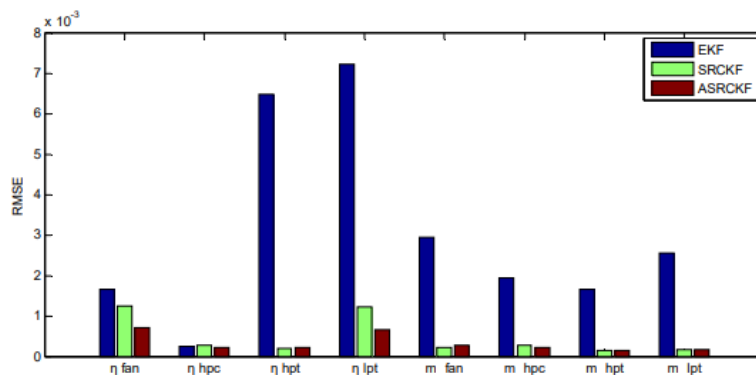


Figura 7. Comparación de, ASRCKF, SRCKF y EKF en RMSE

Fuente (Zhu et al., 2019)

En motores de inducción se puede aplicar el filtro de Kalman para la estimación de los parámetros de este, para ello se compara el EKF y el UKF para este tipo de sistema no lineal, específicamente los parámetros deseados son el torque y la velocidad del motor no se utiliza

ningún otro tipo de apoyo en cuanto a algoritmos por lo que la estimación de ambos se hace solamente a partir de las ecuaciones y condiciones de los filtros, en resultados se encuentra que el UKF supera al EKF ya que en perturbación o cambios abruptos en la señal este no puede determinar la velocidad del motor cosa que el UKF hace sin problema, esto se debe a que el UKF no cuenta con la matriz jacobiana. (Darvishi & Doroudi, 2019)

3.5.3 VIGILANCIA EN AVIACIÓN

El filtro de Kalman ha sido uno de los algoritmos más utilizados en la aviación, algo que vale resaltar es que en muchos estudios y ejemplos abordados aquí se utiliza este filtro como base para la creación de nuevos algoritmos aplicados en área específica, en este caso se utiliza el filtro de Kalman original para el desarrollo y prueba de un nuevo método para la vigilancia en aviación comparándolo los resultados de ambos métodos, las pruebas se basaron en 2 etapas una de entrenamiento y otra de verificación, el estudio se deja abierta a nuevas etapas de investigación para la implementación de otros métodos de entrenamiento y hacer más optimo el proceso.

La utilización del filtro de Kalman como centro de una red neuronal recurrente en este caso se denomina filtro de Kalman entrenable, este funcionara como otra capa de la red pudiéndose utilizar un algoritmo de entrenamiento para los parámetros de Kalman, los resultados muestran mayor precisión en esta nueva versión del filtro para la tarea de datos confiables y más acertados del radar de vigilancia, aunque se observa un excelente filtrado en las secciones iniciales del filtro tradicional. (Y. Jiang & Nong, 2020)

3.5.4 MEJORA EN LA SEÑAL LiDAR 3D DE VEHÍCULOS AUTÓNOMOS

El estudio en soluciones de problemas en vehículos autónomos es realmente importante, en este estudio realizado en un laboratorio que se acondicionó de tal forma que asemejara un ambiente adverso se muestra el uso del filtro de Kalman para el filtrar el ruido del humo ambiental.

La señal en los sistemas LiDAR puede verse afectada o limitada por el mismo o por otras razones como el ruido, se introduce la utilización del filtro junto a otro algoritmo de segmentación de los puntos de la nube bajo condiciones ambientales difíciles siendo el modelo

utilizado para las pruebas y verificación es el LiDAR PUNCK VLP-16, la utilización de ambos algoritmos en este muestra resultados favorables en %15 en comparación a si se utilizara solo uno de los dos, cabe mencionar que la simulación se realizó bajo un espacio controlado así que si se desea la utilización en el ambiente real se deberán tomar algunas consideraciones de medición y verificación propuestas para futuras investigaciones. (Lin & Wu, 2021)

3.5.5 SEGUIMIENTO DE OBJETOS

Para el análisis de movimiento en objetos un factor de mejora es el seguimiento de objetos, así que aunque el EKF y el UKF tiene distintos enfoques y forma de estimar resultados en sistemas no lineales, en este estudio se comparan para la estimación de seguimiento de objetos sin la ayuda de otro tipo de algoritmo para el proceso de filtrado u optimización de medición, solamente bajo sus condiciones de trabajo, los resultados muestran mejor rendimiento, filtrado y precisión en UKF debido a que aborda el proceso de forma no lineal, en cambio los datos de estimación se muestran muy por debajo en el EKF que debido a matriz jacobiana y el hecho de linealizar el estado se acumulan errores en las mediciones haciendo que sea un poco más complicado la validación de resultados fiables. (Deng et al., 2020)

3.5.6 PROCESAMIENTO DE DATOS DE RADAR TERRESTRE INTERFERÓMETRICO

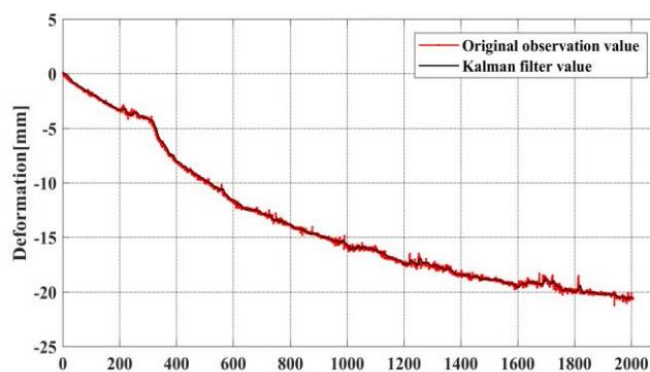


Figura 8. Comparación de señal estimada por filtro de Kalman y la original

Fuente (Tan et al., 2019)

La estimación de la deformación terrestres es de suma importancia, pero al igual que otros sistemas se muestran interferencias y ruidos, una buena estimación y adquisición de datos de deformación puede llegar a prevenir desastres naturales por erosiones u otros fenómenos que afectan la superficie terrestre, se considera al filtro de Kalman para el procesamiento de

datos de deformación por radar interferométrico de tipo diferencial el cual utiliza un equipo SAR para la tarea de vigilancia, para la simulación del filtrado de Kalman se utiliza el simulador de Matlab y básicamente el valor de inicio será arbitrario, en los resultados se observa que el filtrado de Kalman arroja una curva mucho más suave en comparación a los datos originales y es justamente lo que se esperaba llegando a estimar una señal mucho más limpia y libre de perturbaciones por ambiente e incluso el mismo radar. (Tan et al., 2019)

CAPÍTULO IV. METODOLOGÍA

4.1 ENFOQUE

Debido a que el análisis de las distintas variables de interés se dirigen a la obtención de un modelo matemático para la trayectoria de robots móviles mediante un algoritmo recursivo, se estima el enfoque de este como cuantitativo, este consta de una serie de etapas donde se analiza el comportamiento para poder predecir la posición del robot en un determinado tiempo mediante métodos numéricos.

En un estudio de tipo cuantitativo para el desarrollo de interfaz de usuarios de aplicaciones móviles realizado por Jiang et al. (2019) aseguran que el uso de análisis cuantitativo puede optimizar el diseño de interfaz y utiliza estas funciones como pautas para tener una visión original, objetiva y única. En tanto que en su investigación para sistemas de entrega voladores Jeong et al. (2020) utiliza un enfoque cuantitativo para el análisis del funcionamiento de estos sistemas y debido a los excelentes resultados se recomienda dicho enfoque para futuras investigaciones con la posibilidad de implementar y optimizar la colaboración de sistemas distintos por medio del desarrollo de un modelo matemático.

4.2 VARIABLES DE INVESTIGACIÓN

Sin duda una de las partes importantes a estudiar son el conjunto de variables del sistema, en él se distinguen las de tipo dependientes que van a tomar valores distintos según sea el de las variables independientes, una sola variable dependiente deberá ser estimada a partir de varias variables independientes las cuales afectan de manera directa la primera.

Básicamente, las variables dependientes serán las que en este estudio se trata de encontrar y dar solución, mediante la manipulación y estudio de las variables independientes que en este caso para la utilización de un robot de tipo diferencial se cuenta con velocidades diferentes en las ruedas tanto lineales como angulares, el tiempo en el que se desee estimar la posición del robot, también se necesita una buena estimación en la corrección de señal mediante la implementación del filtro de Kalman, la que además para ser óptima necesita una buena predicción así como de una ganancia correcta.

Variables dependientes

- 1) Posición del robot: a simple vista no se puede saber en dónde se encuentra ubicado el robot, se deberá manipular otras variables para su correcta estimación.

Variables Independiente

- 1) Velocidad lineal del robot: en un robot diferencial la velocidad lineal tangencial en sus ruedas es diferente de esto depende el tipo de movimiento que realice y por ende las coordenadas donde se ubique el robot.
- 2) Velocidad angular del robot: esta velocidad es la encargada de hacer que el robot pueda girar hacia el objetivo, evadir obstáculos o girar sobre su propio eje.
- 3) Corrección de filtro de Kalman: la correcta eliminación del ruido del proceso dependerá de la predicción que se haga, que a su vez será actualizada en la etapa correctiva dando así el valor esperado, dada por metros para la posición en (x, y) y radianes para la orientación.

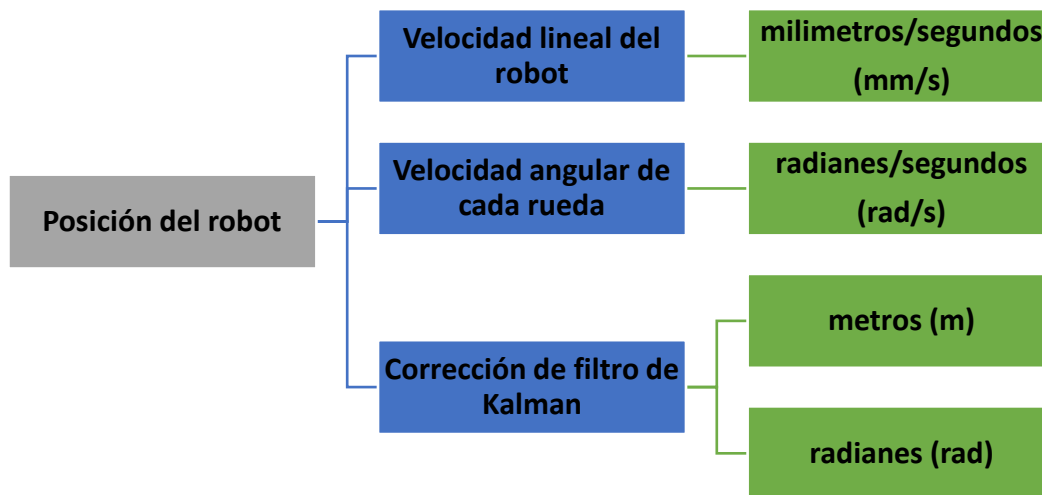


Figura 9. Variables de investigación

Fuente elaboración propia

4.3 TÉCNICAS E INSTRUMENTOS APLICADOS

El desarrollo del modelo matemático será a partir de un algoritmo de naturaleza recursiva, para el cual se necesita la simulación y comprobación de este mediante la resolución de matrices de las ecuaciones de estado, para ello se utilizará el software de MATLAB en

conjunto con simulador de CoppeliaSim versión institucional para observar el comportamiento del robot.

4.4 METODOLOGÍA DE ESTUDIO

Para llegar a la resolución de los problemas planteados en una investigación es necesario la implementación de un método que brinde un procedimiento confiable y apto para la obtención de datos, la metodología aplicada para el desarrollo de modelo matemático será de tipo espiral que brinda un enfoque iterativo compuesto de 4 pasos o etapas (planificación, análisis de riesgo, desarrollo e implementación, evaluación) que se conectan una tras otra, donde se destaca el de análisis de riesgos necesaria para la reducción de errores en la estimación de trayectoria y posición del robot.

El modelo en espiral es un método naturalmente flexible que permite en cada iteración la aplicación de enfoques de construcción, debido a la disminución de riesgos en cada etapa es aplicado a proyectos grandes, en la gráfica de iteraciones es posible medir el progreso de cada una, donde cada fase comienza justo después de la anterior formando así un bucle de iteraciones. (Velásquez et al., 2019)

4.4.1 ITERACIÓN I. DISEÑO DEL DIAGRAMA CINEMÁTICO DEL ROBOT

Etapa I. Planificación

En esta se deben determinar los objetivos para la iteración, así como el diseño del diagrama de cinemática directa e inversa el robot de tipo diferencial disponible en la hoja de especificaciones del simulador de CoppeliaSim, con toda las partes y dimensiones necesarias para su modelado, también se deberá estimar los riesgos a los que se pueden incurrir, al final esta etapa constará de un 15% del proceso.

Etapa II. Análisis de riesgos

Se deben analizar todos y cada uno de los riesgos ya encontrados en el diseño del diagrama y buscar la forma de reducirlos a partir de estrategias de modelado.

Etapa III. Desarrollo e implementación

Teniendo clara la idea de diseño y haber tomado en cuenta la reducción de riesgos se debe iniciar la trazabilidad del diagrama de cinemática con todos los elementos simbólicos, como determinación de las variables en el diagrama y sus respectivas ecuaciones.

Etapa IV. Evaluación

Se hace la revisión de todas las etapas y cumplimiento de los objetivos iniciales para evaluar si pasar a la siguiente espiral, de ser este el caso se debe planificar el desarrollo de la iteración 2.

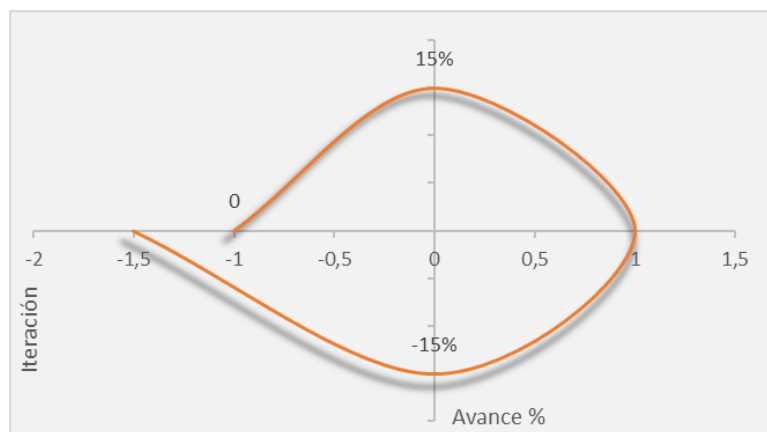


Figura 10. Primera iteración

4.4.2 ITERACIÓN II. IMPLEMENTACIÓN ALGORITMO DE FILTRO DE KALMAN

Etapa I. Planificación

Después de tener la primera espiral totalmente terminada se procede a la determinación de los objetivos de la segunda, determinación de cual variación del filtro de Kalman es más conveniente para el diseño anterior con cada uno de sus parámetros y pasos, al final de esta iteración el proceso se encuentra en un avance del 40%, siendo específicamente un 25% correspondiente a la implementación del filtro de Kalman.

Etapa II. Análisis de riesgos

En esta etapa los riesgos se hacen evidentes pues dependerá de que versión del filtro se utilice también del seguimiento correcto de cada paso de este, es preciso identificar cada riesgo que se corre y corregirlo para que no afecten la siguiente iteración.

Etapa III. Desarrollo e implementación

Después de la corrección de riesgos de debe desarrollar el algoritmo de Kalman el cual consta de dos etapas una de predicción y otra de corrección.

Etapa IV. Evaluación

Se revisan los resultados de toda la espiral y se procede a la planificación de la siguiente.

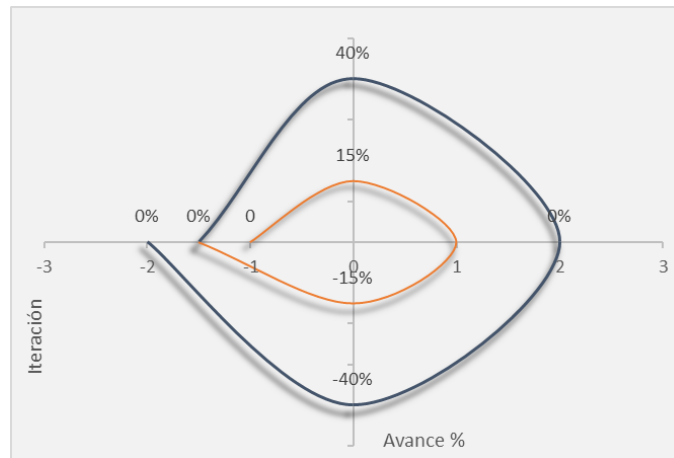


Figura 11. Segunda iteración

4.4.3 ITERACIÓN III. DESARROLLO DE MATRICES

Etapa I. Planificación

Los objetivos en esta iteración son claros pues lo primordial es dar solución a cada una de las matrices implicadas en el modelado cinemático como en el filtrado para poder ser evaluadas en MATLAB, al finalizar la tercera iteración el proceso estará en un 70% terminado donde un 30% equivale al desarrollo de matrices para el modelo matemático final.

Etapa II. Análisis de riesgos

Los riesgos en esta iteración son proporcionales a la anterior donde si la implementación del filtro no fue correcta afectara de forma directa las matrices, por lo que de igual manera se evalúan y se estiman las soluciones de ellos mismos.

Etapa III. Desarrollo e implementación

En esta etapa se desarrollan cada una de las matrices y ecuaciones de estado haciendo uso de MATLAB, para proceder a su correspondiente evaluación.

Etapa IV. Evaluación

Se hace la revisión de las matrices y el cumplimiento de los objetivos iniciales, así como del modelo matemático general y se procede a la planificación de la última iteración.

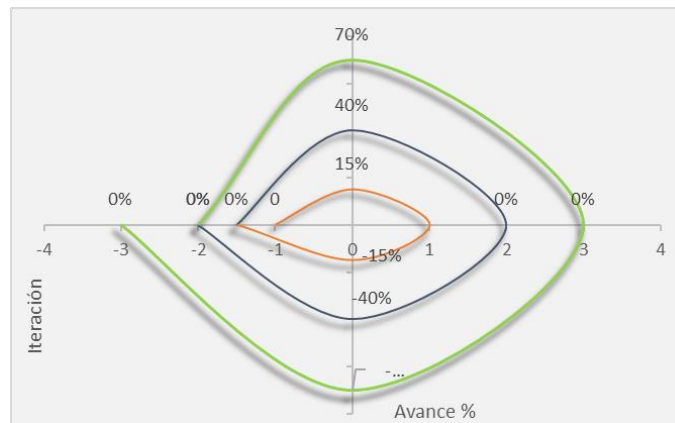


Figura 12. Tercera iteración

4.4.4 ITERACIÓN IV. ELABORACIÓN DE ESCENARIOS Y SIMULACIONES

Etapa I. Planificación

Esta parte corresponde a la última iteración para ellos los objetivos están dirigidos a la realización de escenarios de simulación en CoppeliaSim y el simulado del modelo matemático obtenido en la espiral anterior para ser comprobados en los escenarios anteriormente elaborados, también se pretende encontrar estudios científicos aplicados a este tipo de robot para la planificación de trayectoria, pero con otros algoritmos.

Etapa II. Análisis de riesgos

En esta espiral los riesgos están en la conectividad de los simuladores y respuesta que estos den, para ello se deberán estimar las soluciones óptimas para que estos desaparezcan.

Etapa III. Desarrollo e implementación

Se realizan las simulaciones correspondientes del modelo matemático y la estimación de la posición del robot en determinado tiempo, como el correcto trazado de trayectoria.

Etapa IV. Evaluación

Se procede a la comparación de los resultados del modelo propuesto a partir del filtro de Kalman y otros estudios con otros algoritmos, pero para el mismo tipo de robot.

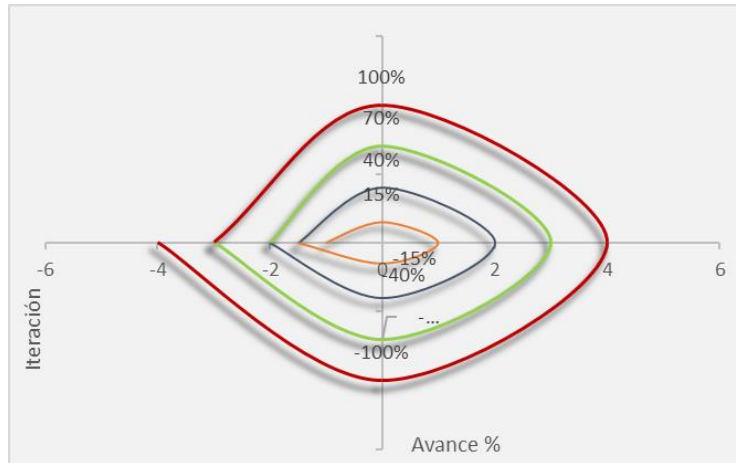


Figura 13. Cuarta iteración

4.5 CRONOGRAMA DE ACTIVIDADES

Para el seguimiento y estimación de tiempo de cada actividad a realizar para el desarrollo del modelo matemático se realizó un cronograma de actividades según el tiempo dispuesto para la investigación que consta de 10 semanas y las entregas estipuladas para cada una de ellas, comenzando desde la presentación de propuesta, investigación clave para el marco teórico, seguido de la metodología y así sucesivamente hasta llegar a la última semana en la que se hará la practica final del proyecto ante el asesor.

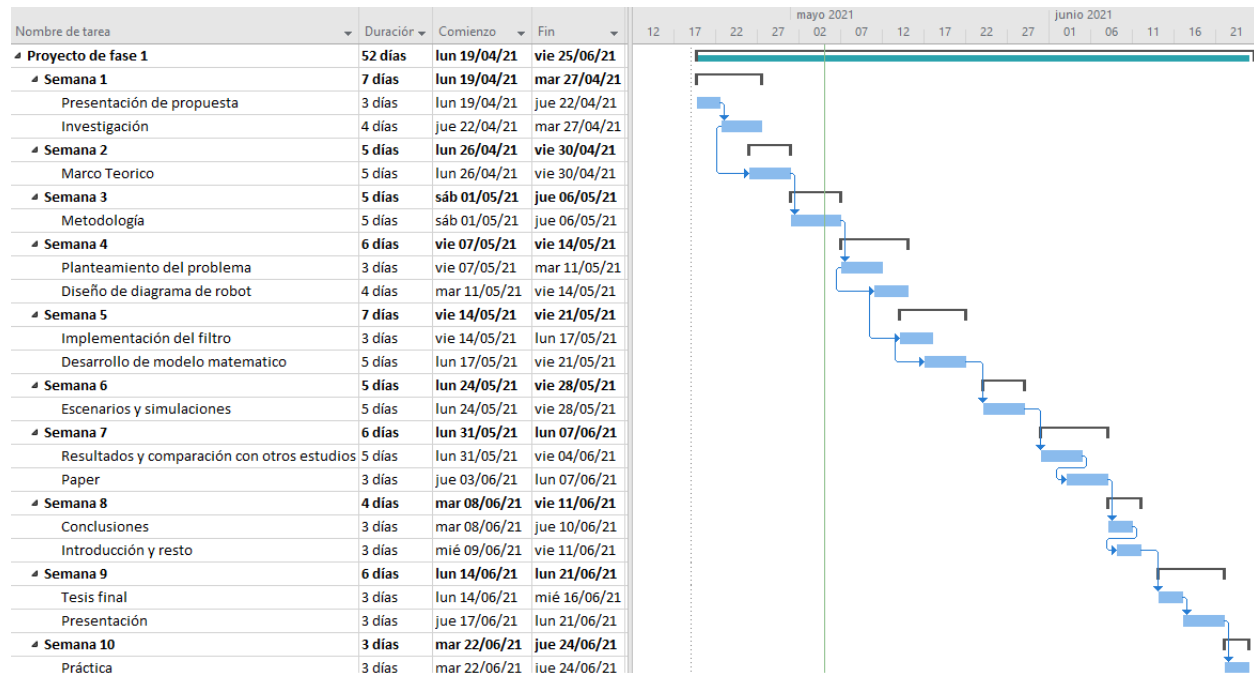


Figura 14. Cronograma de actividades para el proyecto

CAPÍTULO V. ANÁLISIS Y RESULTADOS

El presente capítulo está dirigido al desarrollo completo del modelo matemático basado en el Filtro de Kalman sustentado a partir de la teoría propuesta en el capítulo 3 y mediante una metodología de tipo espiral la cual se dividirá en 4 iteraciones, desde el análisis del robot y la implementación del filtro como en el desarrollo de matrices y simulaciones para comprobar el funcionamiento del modelo final.

5.1 ANÁLISIS DE TEORÍA DE SUSTENTO

Los modelos matemáticos según la aleatoriedad de sus eventos pueden ser de carácter estocástico o probabilísticos y determinista, así se establece que el estudio de un robot móvil debido a muchas influencias internas de este o externas como ruidos o perturbaciones en las mediciones de sensores, calibración y otras es un proceso de tipo estocástico, es decir que está sujeto a eventos sumamente aleatorios por lo cual no se puede predecir su trayectoria o posición con mucha facilidad por lo que es necesario la utilización de algoritmos diseñados para darle al robot una característica de autonomía y mayor eficiencia.

Dentro los robot de ruedas se distinguen 3 de los tipos más utilizados donde se encuentran, los omnidireccionales, los estilo auto o Ackerman y los de tracción diferencial, este último es altamente utilizado en robótica por su fácil modelado y la característica de que cada rueda es independiente y al tener diferentes velocidades le da la facilidad de girar, esquivar y tomar diferentes trayectorias, para ello el software de CoppeliaSim o V-Rep dispone de algunos modelos de robot dentro de los cuales está el Pioneer 3-DX de tipo diferencial y sobre el cual se estará probando el modelo matemático final, dentro de los algoritmos mencionados para la estimación de trayectoria y posición de robot móviles se encuentra el A*, RRT, Dijkstra y Monte Carlo, Kevin Passino, pero para este estudio se pretende trabajar con el filtro de Kalman ya que al ser un algoritmo probabilístico y de tipo recursivo brinda una buena estimación de la señal, además que ha sido altamente utilizado en robótica, aunque en este estudio no se estará aplicando el filtro tradicional que aunque es altamente óptimo para sistemas lineales, en robótica y en la mayoría de áreas los procesos están lejos de ser lineales, podría utilizarse el filtro en casos sumamente triviales e ideales donde se realicen tareas sencillas y se movilice en

trayectorias lineales, pero si el objetivo es trabajar en escenarios que aunque controlados también sean más parecidos a la realidad donde los entornos no siempre son armoniosos y se pueden generar trayectorias diversas, se puede recurrir a la utilización de una de sus actualizaciones como ser el filtro de Kalman extendido donde antes de aplicar las dos etapas características del filtro se deberán linealizar las ecuaciones de estado, el filtro de Kalman sin perfume puede ser utilizado en sistemas altamente no lineales donde el EKF no brindaría datos precisos debido al cálculo de la matriz Jacobiana, así que en el UKF no se linealizan las ecuaciones si no que se trabaja mediante la muestra de muchos puntos aplicando una transformación sin perfume y el filtro de Kalman de cubicación de raíz cuadrada al igual que el UKF se utiliza en sistemas altamente lineales aunque se hacen mejoras a los dos tipos anteriores se hace en base a un muestreo determinista, pero para la aplicación que se busca en el presente estudio bastará la utilización del filtro de Kalman extendido del cual se espera un buen resultado.

5.2 RESULTADOS

Una de las ventajas de utilizar una metodología en espiral es su etapa de reducción de riesgos en cada iteración la cual está directamente conectada a su predecesora, obteniendo así una forma muy confiable para la estimación de buenos resultados, cada iteración o espiral se compone de 4 pasos o etapas: planificación, análisis de riesgos, desarrollo e implementación y la evaluación, al final de cada espiral también se planifica la siguiente iteración.

5.2.1 DISEÑO DEL DIAGRAMA CINEMÁTICO DEL ROBOT

Planificación

Para el diseño del diagrama del robot de tracción diferencial se toman en cuentas varios aspectos el primero es que se basa en el diseño de un robot Pioneer P3DX con sus dimensiones, estructura y otras especificaciones propias de este modelo de dos ruedas independientes y una tercera de apoyo en la parte trasera, su navegación es de tipo reactiva, pero es altamente personalizable es decir que se le pueden incluir otras partes para hacerlo un tanto más inteligente, además que su trabajo para la evasión de obstáculos es muy confiable. Así que

como objetivo general para la primera espiral se tiene el diseñar el diagrama cinemático de un robot Pioneer P3DX de tipo diferencial con sus respectivas ecuaciones características.

Análisis de riesgos

Al momento de realizar el diagrama de un robot se puede disponer tanto de uno de tipo cinemático y uno dinámico, pero como ya se mencionaba en el marco teórico el control de movimiento en robot diferenciales se hace generalmente con el cinemático por dos razones, la primera el modelo dinámico puede tornarse complejo y la segunda que no utilizan una velocidad alta para el control del lazo al contrario se usa la más baja.

Matriz de riesgos

Tabla 6. Matriz de riesgo en iteración I

Escenario de riesgo	Probabilidad de que ocurra	Impacto si ocurriera	Riesgo inicial
No considerar todos los parámetros en el diagrama cinemático.	Baja	Alto	Medio
A la hora de la implementación del algoritmo necesitar el diagrama dinámico y no disponer de él.	Baja	Medio	Bajo

Fuente elaboración propia

Desarrollo e implementación

Antes de dibujar el robot es necesario conocer las especificaciones de este, para después tomar el diagrama y simbología que se está utilizando para dicho robot, se hace uso del manual del robot, así como de otra literatura para validar las demás partes.

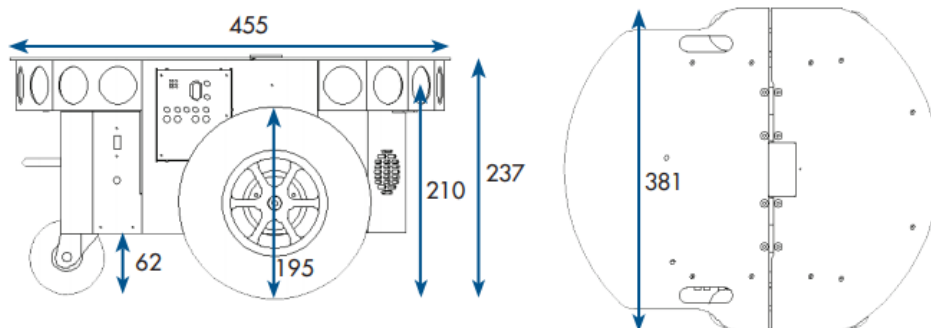


Figura 15. Dimensiones de robot diferencial Pioneer P3DX en mm

Fuente (Adept Technology Inc, 2021)

Especificaciones de Pioneer P3DX

Estas son dadas por el fabricante y en su mayoría son las características del robot físico como el tiempo de carga, peso y otros, aunque también aplican al entorno simulado como la velocidad, dimensiones y están detalladas en la siguiente tabla:

Tabla 7. Datos del robot

Parámetro	Medida
Velocidad máx.	1.2 m/s
Dimensiones	455x381x237mm
Giro	300°/s
Voltaje	12V
Capacidad de carga	17Kg
Peso del robot	9Kg
Horas de trabajo	8-10 h
Diámetro de ruedas	195mm
Tiempo de carga	12h estándar, 2.4h especial
Diámetro de rueda de apoyo	62mm

Fuente (Adept Technology Inc, 2021)

Diagrama cinemático

Para este se utiliza el diseño del Pioneer P3DX y las ecuaciones 6, 7 y 8 propuestas en el capítulo 3 para robot diferenciales y mediante el uso de SolidWorks se realiza el dibujo.

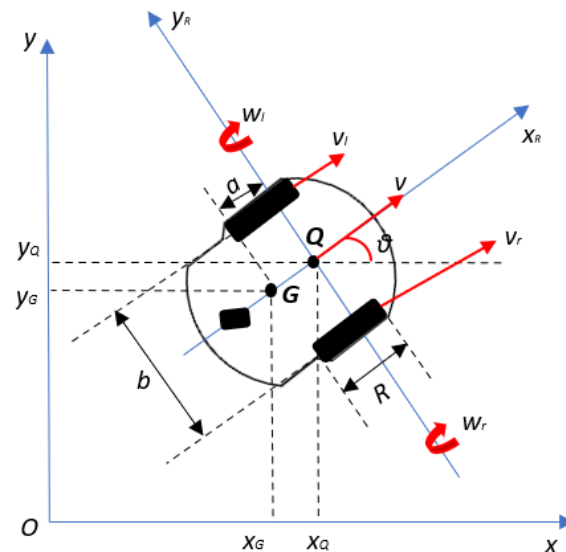


Figura 16. Diagrama cinemático del Pioneer P3DX

Donde los parámetros siguen siendo los mismos a diferencia de "R" que ahora representa el diámetro de las ruedas, Así se presentan el conjunto de ecuaciones cinemáticas en su forma normal y matricial.

$$\begin{aligned} \dot{x} &= v \cos \theta \\ \dot{y} &= v \operatorname{sen} \theta \\ \dot{\theta} &= \omega \end{aligned} \quad \rightarrow \quad \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \operatorname{sen} \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

Ecuación 39. Modelo cinemático de robot diferencial

Fuente (Stefek et al., 2020)

Básicamente lo que este conjunto de ecuaciones nos dice es que la primera derivada de la posición respecto de x , y , θ , es decir las velocidades de estas dependen de la velocidad tangencial y de la angular del robot.

Donde "v" estará dada por el promedio de las velocidades de ambas ruedas, "w" estará dada por la diferencia de la rueda derecha y la izquierda entre la distancia que hay entre estas representada por la letra "b" y que es una constante de la siguiente forma.

$$v = \frac{v_r + v_l}{2} \quad , \quad \omega = \frac{v_r - v_l}{b}$$

Ecuación 40. Velocidad lineal tangencial y velocidad angular

Fuente (Stefek et al., 2020)

Evaluación

Para la primera iteración que consta tan solo del 15% del análisis se logró realizar el diagrama cinemático del robot Pioneer P3DX a partir de su forma original y los parámetros propuestos en el marco teórico, se evaluaron los posibles riesgos en esta espiral y se toman en cuenta para las siguientes. Para la siguiente iteración se pretende la implementación del filtro de Kalman para trayectoria de robot, ecuaciones características, pasos a realizar y todo lo necesario para la aplicación de este al robot propuesto.

5.2.2 IMPLEMENTACIÓN DEL ALGORITMO DE FILTRO DE KALMAN

Planificación

El objetivo principal es desglosar los pasos a seguir para la implementación del filtro de Kalman, describiendo para el sistema un Filtro de Kalman extendido debido a la generación de trayectoria no lineales.

Análisis de riesgos

Debido a la no linealidad del sistema se debe utilizar una de las actualizaciones del filtro de Kalman, dentro de las 3 abordadas en el marco teórico se considera el filtro de Kalman extendido que, aunque se corre el riesgo de tener problemas por la linealización de las ecuaciones y que se produzcan medias y covarianzas diferentes a la realidad también puede brindar buenos resultados para lo que se quiere, en cuanto a los otros dos como el UKF y SRCKF se requiere mayor memoria y más pasos aunque mantienen el principio de las dos etapas de predicción y corrección.

Matriz de riesgo

Tabla 8. Matriz de riesgo en iteración II

Escenario de riesgo	Probabilidad de que ocurra	Impacto si ocurriera	Riesgo inicial
No seguir los pasos adecuadamente.	Baja	Alto	Bajo
Problemas por sistema altamente no lineal.	Baja	Alto	Bajo

Fuente elaboración propia

Desarrollo e implementación

El EKF trabaja con el principio del filtro tradicional con la diferencia que lo hace a partir de la linealización de las ecuaciones de estado con la expansión de Taylor, básicamente lo que sucede es que al tener un sistema lineal que causa perturbaciones de certidumbre no se puede ajustar este sistema a una distribución gaussiana, por esta razón se busca mediante la linealización ajustar o aproximar la función no lineal con un tipo de función que si es lineal y tangente a ella en la media de la distribución gaussiana, como se observa en la (figura 17).

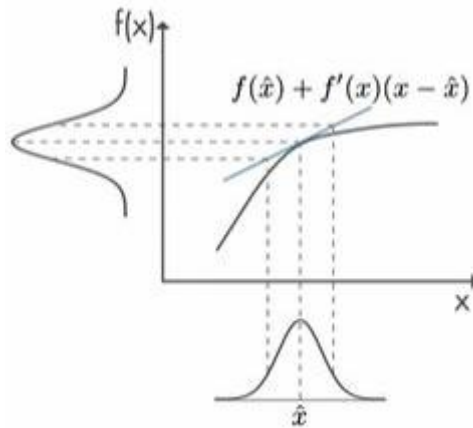


Figura 17. Linealización mediante serie de Taylor

Fuente (Ulusoy, 2017)

La expansión de Taylor está dada por:

$$x(t + \Delta t) = x(t) + \Delta t \dot{x}(t) + \frac{(\Delta t)^2}{2!} \ddot{x}(t) + \frac{(\Delta t)^3}{3!} \dddot{x}(t) + \dots$$

Ecuación 41. Expansión de Taylor

Fuente (Aranda R & Martínez H, 2017)

Pasos para aplicación de filtro de Kalman extendido

El proceso para la aplicación del EKF consta de las dos etapas características, pero en estas se incluye la linealización de la ecuación de estado y el desarrollo de las matrices jacobianas para ambas etapas.

Predicción

- 1) Se estima el estado a priori

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0)$$

Ecuación 42. Ecuaciones de estado

$$x_k = [x_k \ y_k \ \theta_k]^T$$

Ecuación 43. Vector de estado

$$u_k = [v_k \ \omega_k]^T = N(0, Q_k)$$

Ecuación 44. Señal de control

$$w_k = [w_{v,k} \ w_{\omega,k}]^T = N(0, R_k)$$

Ecuación 45. Ruido del sistema

Donde el vector de estado estará dado por la posición en x, y, así como por la orientación del robot, al mismo tiempo que el robot será controlado por la velocidad lineal y angular de este, para ser afectado por el ruido referentes a la velocidad lineal y angular.

2) Se proyecta la covarianza a priori

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

Ecuación 46. Covarianza a priori

3) Derivadas parciales de jacobianos de predicción

$$A_k = \frac{\partial f_x}{\partial x_k}$$

Ecuación 47. Jacobiano de predicción respecto del vector de estado

$$W_k = \frac{\partial f_x}{\partial w_k}$$

Ecuación 48. Jacobiano de predicción respecto de la matriz de control

Estos jacobianos son las derivadas parciales respecto a las ecuaciones del primer paso que a su vez están dadas por el modelo cinemático del robot propuesto en la iteración I.

Corrección

4) Calcular la ganancia de Kalman

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} = \frac{P_k^- H_k^T}{H_k P_k^- H_k^T + R_k}$$

Ecuación 49. Cálculo de ganancia del EKF

5) Derivadas parciales de jacobianos de corrección

$$H_k = \frac{\partial h_x}{\partial x_k}$$

Ecuación 50. Jacobiano de medición respecto del estado zk

6) Actualización del estado a posteriori o media real

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0))$$

Ecuación 51. Actualización del estado a posteriori EKF

7) Actualización de la covarianza a posteriori

$$P_k = P_k^- - K_k H P_k^-$$

Ecuación 52. Actualización de la covarianza EKF

Fuente (Balico et al., 2018; Fahmedha et al., 2015; Madhukar & Prasad, 2020; Simon, 2006; Toriz et al., 2010)

8) Repetir desde el paso 1

En la etapa de corrección se hace el cálculo de la ganancia de Kalman el cual es un valor que varía de 0 a 1, si K se aproxima a 0, se tendrán medidas inexactas con estimaciones estables y poco error, si K se aproxima a 1, se tienen medidas exactas con estimaciones inestables. Luego la actualización de la media se hace a partir del valor de z_k que representa la salida y ya no es una función f como en la predicción si no una función h .

Evaluación

Se logró desglosar todos los pasos del EKF a partir del modelo matemático de la primera iteración, siendo en total 7 pasos que al terminar se repiten, hasta este punto de logra alcanzar un 40% de la metodología en general, para la siguiente espiral se requiere el desarrollo de las matrices jacobianas, así como las demás matrices necesarias como las de covarianza tanto de estado a priori como a posteriori.

5.2.3 DESARROLLO DE MATRICES DEL SISTEMA

Planificación

El objetivo inicial es el desarrollo de matrices jacobianas y generales del EKF para mostrar el modelo matemático final y posteriormente en la siguiente iteración ser evaluadas mediante Matlab, al finalizar el cálculo de matrices se podrá alcanzar un 70% de la metodología total con el modelo matemático terminado teóricamente, para su posterior implementación en el simulador que corresponde al porcentaje restante.

Análisis de riesgos

Esta tal vez sea una de las iteraciones con más riesgos debido al cálculo de los jacobianos, ya que si estos no son correctos el EKF no podrá estimar bien la posición del robot y se tendrá un rango de error bastante elevado y nada parecido al real.

Tabla 9. Matriz de riesgos en iteración III.

Escenario de riesgo	Probabilidad de que ocurra	Impacto si ocurriera	Riesgo inicial
Calcular mal los jacobianos del sistema	Media	Alto	Medio

Fuente elaboración propia

Desarrollo e implementación

Para poder implementar el EKF en el robot Pioneer P3DX además de seguir los pasos de la iteración 2, es necesario calcular 3 matrices jacobianas, 2 para la etapa de predicción y 1 más para la etapa de corrección, así como las matrices de covarianza de ambas etapas a partir del modelo cinemático del robot propuesto en la espiral 1, a continuación, el cálculo de las matrices y posteriormente el modelo matemático general.

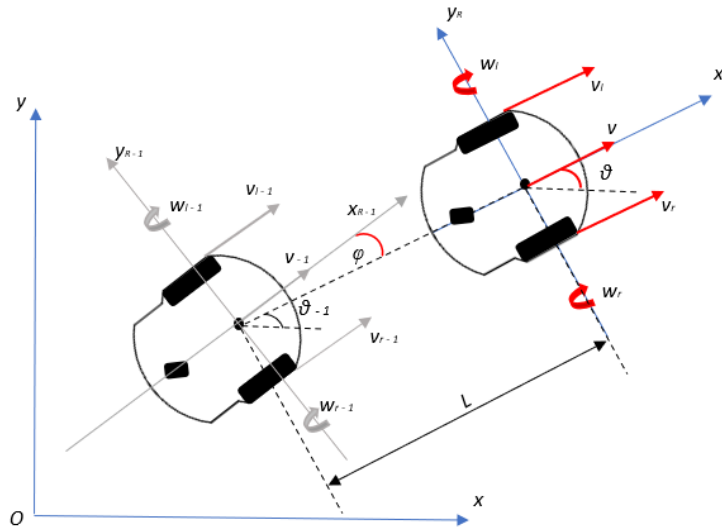


Figura 18. Nueva posición del robot con respecto a la anterior

Primero basado el modelo cinemático de la (ecuación 39) se puede disponer para encontrar la nueva posición a partir de la posición anterior, así como se muestra en la figura anterior que es una modificación del modelo cinemático tomando la posición inicial como una antes en el tiempo y que la nueva posición adopte las características de la primera, donde "L" es la distancia entre el punto de referencia y el siguiente donde estará el robot, " φ " es el ángulo al que las ruedas deben girar para la nueva posición, también se genera así un diferencial en las

velocidades y el tiempo, con estas características el vector de ecuaciones quedaría de la siguiente forma.

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + v \cdot \Delta t \cdot \cos\theta \\ y_{k-1} + v \cdot \Delta t \cdot \sen\theta \\ \theta_{k-1} + \Delta t \cdot \omega \end{bmatrix}$$

Ecuación 53. Modelo cinemático de robot diferencial en un tiempo actual

La señal de control que en la (ecuación 44) nos dice que está dada por la velocidad lineal y angular del robot, por nuestro modelo cinemático nos queda de la siguiente forma.

$$u_k = [v_k \ \omega_k]^T = \begin{bmatrix} v \cdot \cos\theta \\ v \cdot \sen\theta \\ \omega \end{bmatrix}$$

Ecuación 54. Señal de control del robot

La señal de medición de la parte correctiva está dada por la siguiente forma donde se toma la medida de distancia entre una posición y otra, y el ángulo requerido para la nueva posición, para visualizar mejor esto se puede ver la (figura 18).

$$z_k = h(\hat{x}_k) = \begin{bmatrix} L \\ \varphi \end{bmatrix} = \begin{bmatrix} (x - xa)\cos\varphi + (y - ya)\sen\varphi \\ \frac{\sen\varphi}{\cos\varphi} - \theta \end{bmatrix}$$

Ecuación 55. Señal de medición

Matrices Jacobianas

1)

$$A_k = \frac{\partial f_x}{\partial x_k} = \begin{bmatrix} 1 & 0 & -v \cdot \Delta t \cdot \sen\theta \\ 0 & 1 & v \cdot \Delta t \cdot \cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 56. Matriz jacobiana de predicción respecto del vector de estado

2)

$$W_k = \frac{\partial f_x}{\partial w_k} = \begin{bmatrix} \cos\theta & -v\sen\theta \\ \sen\theta & v\cos\theta \\ 0 & 1 \end{bmatrix}$$

Ecuación 57. Matriz Jacobiana de predicción respecto del ruido

3)

$$H_k = \frac{\partial h_x}{\partial x_k} = \begin{bmatrix} -\cos\theta & -\text{sen}\theta & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Ecuación 58. Matriz jacobiana de corrección respecto de la señal de medición

Matrices de covarianza

$$Q_{k-1} = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix}$$

Ecuación 59. Matriz diagonal de covarianza a priori

$$R_k = \begin{bmatrix} \sigma_L^2 & 0 \\ 0 & \sigma_\varphi^2 \end{bmatrix}$$

Ecuación 60. Matriz diagonal de covarianza a posteriori

Fuente (Aranda R & Martínez H, 2017; Balico et al., 2018; Bussi et al., 2017; Hernández Millán et al., 2016; Larco Chiriboga & Cortijo Leyva, 2019; Tzafestas, 2014; Zhao et al., 2016)

Las matrices de covarianza están dadas por una matriz diagonal de la desviación estándar al cuadrado o mejor conocido como varianza respecto de las variables de su respectiva etapa, por ejemplo, en el caso de Q_{k-1} la matriz diagonal de 2 x 2 estará dada por la varianza tanto de la velocidad lineal como la angular, en la programación será preciso obtener estos valores directamente de los encoders derecho e izquierdo multiplicando este valor por una constante de control para luego sumar el cuadrado de ellos y ser este el valor que se colocará en la matriz, para R_k se da el valor a la varianza a partir de la distancia entre posiciones del robot y del ángulo necesario al que las ruedas deben girar para la siguiente posición.

Evaluación

Una vez determinadas todas las matrices y tomados en cuenta los riesgos de la implementación y desarrollo de estas se procede a presentar el modelo matemático como uno solo desde la espiral 1 hasta la actual para luego implementar este en Matlab, con el modelo matemático de forma teórica en este punto la metodología está en un 70% terminada, faltando la validación y comparaciones ya en el entorno de simulación

Modelo matemático general

En la (figura 19) se muestra el modelo matemático inspirado a partir del diseño jerárquico de (Hehenberger et al., 2010). En el que se vincula la cinemática y control del robot con las etapas del EKF.

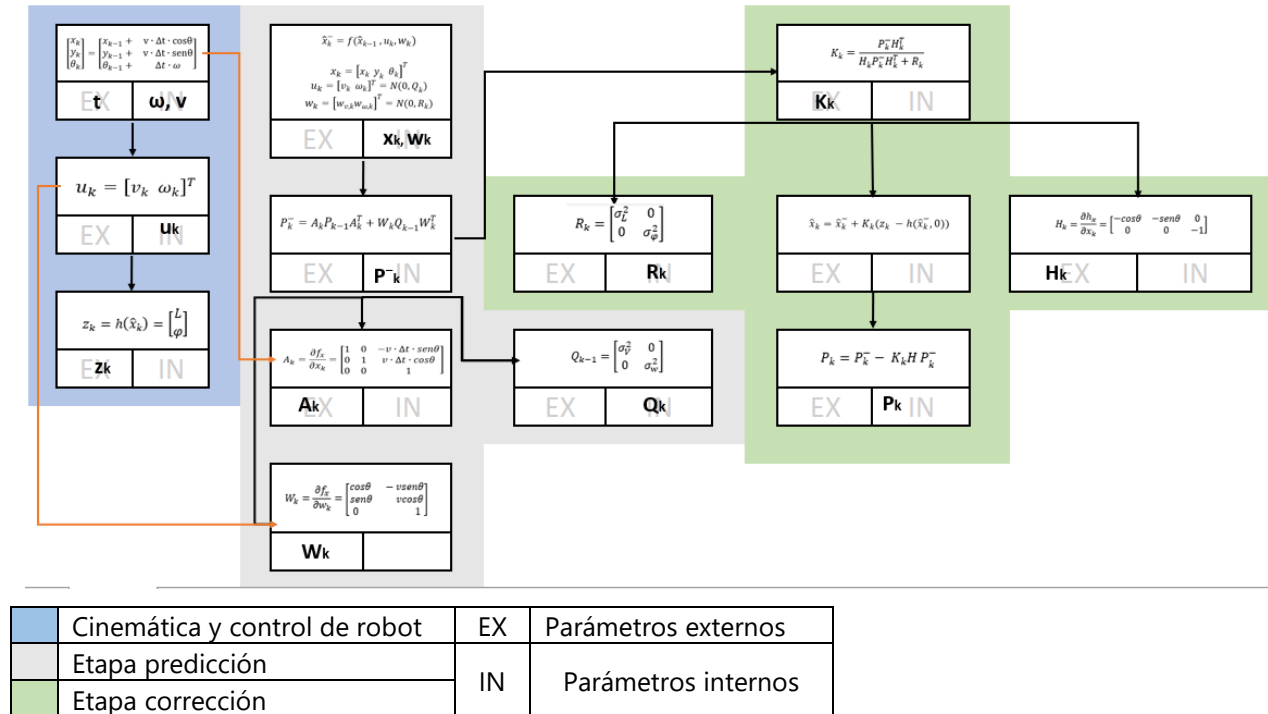


Figura 19. Modelo matemático de filtro de Kalman extendido para robot móvil diferencial

Fuente elaboración propia

5.2.4 ELABORACIÓN DE ESCENARIOS Y SIMULACIONES

Planificación

Uno de los objetivos iniciales de esta iteración es la creación de 5 escenarios en CoppeliaSim para luego implementar el modelo propuesto en la espiral 3, observar el comportamiento y hacer ajustes si es que se requiere, también se pretende implementar otro algoritmo de seguimiento de trayectoria en estos escenarios para comparar los resultados y encontrar la mejor estimación o el mejor uso de cada algoritmo, al finalizar esta espiral se validará totalmente el modelo matemático.

Para la implementación del modelo matemático de EKF propuesto se toma como referencia un robot de tipo diferencial con 2 ruedas independientes y una tercera de apoyo o

equilibrio más conocida como rueda loca, el cual se encuentra en CoppeliaSim (V-Rep) en versión educativa, con el modelo de Pioneer 3-DX, lo que se pretende es aplicar el modelo matemático propuesto en conjunto con la función de (Bijo, 2015) para el control del robot desde Matlab, en este se utiliza una heurística para trazar la trayectoria desde la posición inicial hasta la meta, para lograr la conexión entre estas dos se recurre al uso de la API de conexión entre V-Rep y Matlab detallada en el manual general de CoppeliaSim, inicialmente se pretendía solamente agregar el modelo matemático a la programación en general del robot para comprobar su funcionamiento, pero para la optimización de la misma se hicieron algunos cambios.

El algoritmo para comparar será el propuesto por Kevin Passino y modificado por (Ruiz Bravo, 2019) el que trabaja con odometría para la movilidad del robot, al cual se le hicieron las modificaciones en cuanto a los escenarios, dimensiones y los obstáculos representados por paredes y objetos.

Análisis de riesgos

Esta etapa es muy importante y al igual que las demás presenta riesgos los cuales se verán afectados por las espirales anteriores y del correcto modelado cinemático, seguimiento de pasos y cálculo de matrices, se espera no tener riesgos mayores y poder frenarlos y detectarlos a tiempo para el correcto uso del filtro de Kalman extendido en un robot móvil de tipo diferencial.

Matriz de riesgo

Tabla 10. Matriz de riesgo en iteración IV.

Escenario de riesgo	Probabilidad de que ocurra	Impacto si ocurriera	Riesgo inicial
Matlab y CoppeliaSim no se puedan conectar	Baja	Alto	Bajo
La estimación de posición sea muy diferente a la real.	Media	Alto	Bajo
No se pueda probar otro algoritmo	Medio	Bajo	Medio

Fuente elaboración propia

Desarrollo e implementación

Se cuenta con dos escenarios en los que se incluye tanto a las paredes que son las detectadas por los sensores y objetos que llenan los espacios, pero también se incluyen otros tres escenarios para observar el comportamiento del robot. Algunas consideraciones para tomar en cuenta serían, primero desasociar los scripts que viene por defecto con los objetos como mesas, sillas y personas pues en las primeras pruebas se presentaba una distorsión en los puntos de detección de objetos, también si se cuenta con más de un escenario en CoppeliaSim se recomienda abrir uno a la vez y no tener las ventanas en simultaneo abiertas pues también puede generar lentitud en la simulación, o errores en general.

5.2.4.1 SIMULACIÓN Y COMPARACIONES

Pruebas iniciales

Los escenarios son cuadrados de 5 x 5 m y donde el centro del cuadrado representa el punto cero, así que el robot saldrá de (-1.7, -1.7) m hasta su objetivo que se encuentra en el punto (1.7, 1.7) m, por su naturaleza probabilística donde cada espacio cuenta con la probabilidad de estar ocupado o no, en Matlab se irán trazando diferentes rutas dependiendo de los obstáculos que se detecten, en la gráfica el punto verde representa la meta, el punto rojo grande con una línea negrita de dirección representa el robot y donde se posiciona, la ruta está dada por la sucesión de puntos azules, las paredes por líneas negras y en puntos rojos más pequeños las uniones de paredes.

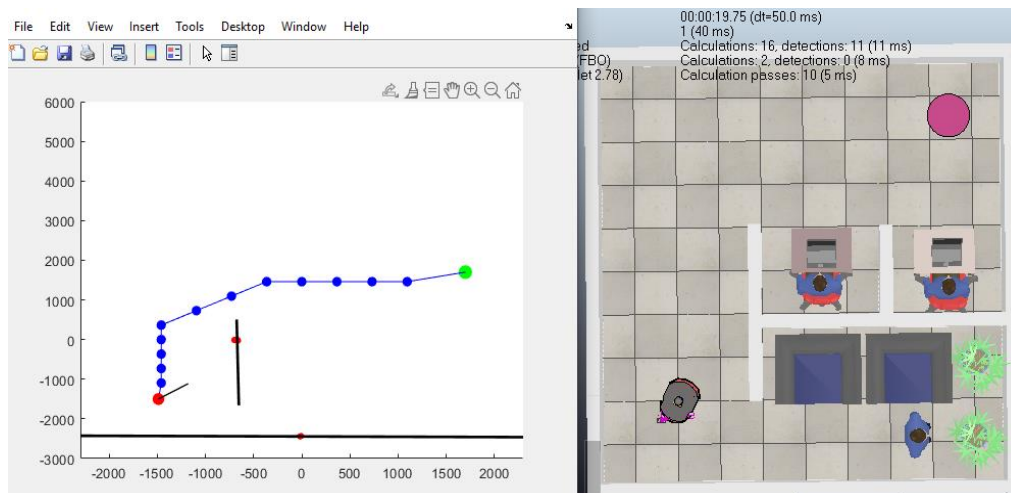


Figura 20. Prueba inicial escenario 1, trazado de ruta

En la (figura 20) para el escenario número 1 se cuenta con un espacio con cubículos, el robot al detectar una pared justo enfrente traza una ruta por donde considera es posible pasar, en CoppeliaSim se le agrego una gráfica al robot para que esta trace la posición en (x, y, z) de la ruta desde la salida hasta la meta, el objeto de estas pruebas iniciales no es el de comparar los resultados en ambos escenarios si no visualizar los errores para poder calibrar y ajustar los datos a los más óptimos.

En la (figura 21) se muestra cuando el robot llega a la meta y en Matlab solamente quedan los puntos donde estaban las paredes detectadas y en CoppeliaSim se muestra la ruta completa después de la planificación de las distintas trayectorias, a simple vista se ven algunas partes donde el robot tuvo cambios de posición (encerrados en rojo), por lo que para las siguientes pruebas se espera mejorar estas partes, así como otros parámetros en el controlador para mejorar la ruta.

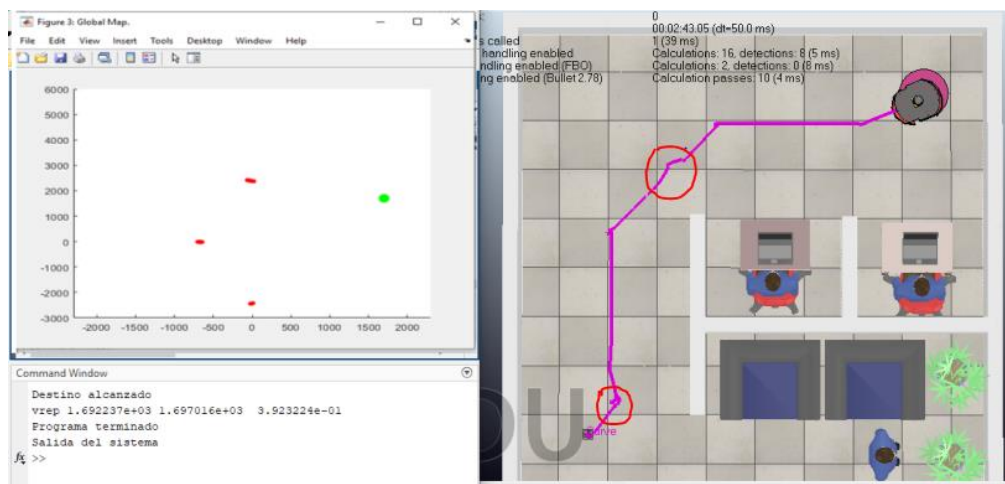


Figura 21. Prueba inicial escenario 1, llegada a la meta

El escenario 2 es una modificación del primero se quita uno de los cubículos y objetos de la parte de abajo para crear una nueva posible ruta y observar si el robot es capaz de seguir otra trayectoria o crear una parecida a la del primer escenario.

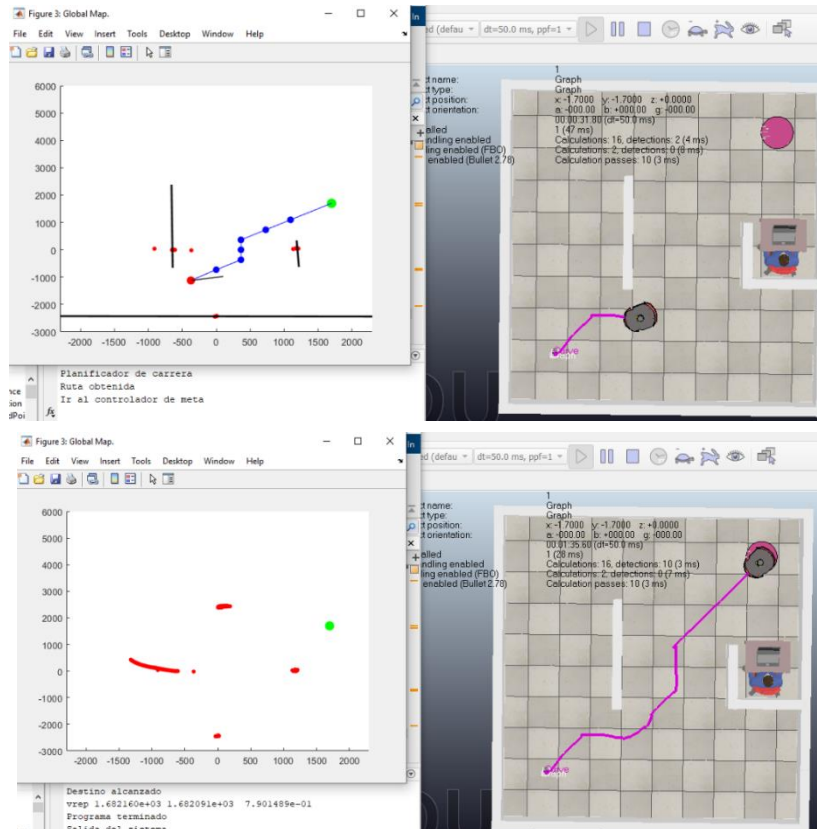


Figura 22. Simulación escenario 2, prueba inicial

En la (figura 22) se observa la nueva trayectoria que el robot toma en el escenario 2, esta se abre paso a través del cubículo despejado del primer escenario, al igual se observan algunos errores de puntos adicionales de pared, deben ajustarse los parámetros del controlador así como los del estimador de posición, ya que la (tabla 7) sacada de la hoja del fabricante nos dice que el Robot Pioneer P3Dx puede alcanzar velocidades de 1.2m/s y de 5.23rad/s, pero en las simulaciones anteriores no es posible trabajar con dichas velocidades así que antes de poder comparar el EKF con el de Kevin Passino, se harán una serie de pruebas para encontrar cuales son realmente las velocidades máximas que realmente soporta el robot bajo este modelo matemático, también es preciso mencionar la ganancia de Kalman valor que esta entre 0 y 1, cuando K tiende a 0 la estimación es más precisa aunque la predicción cambia y cuando K tiende a 1 las estimaciones son muy alejadas de la realidad aunque su predicción si presenta valores aproximados, también el escenario 2 presenta un tiempo de simulación menor al del escenario 1.

Tabla 11. Parámetros y resultados de escenarios 1 y 2 en pruebas iniciales

Escenario	Vmax-Vmin	Wmax-Wmin	Qk-1 Constantes v, w	Rk Sigma L, ϕ	Tiempo	Posición final del robot
1	700mm/s	0.75 rad/s	0.004, 0.007	150mm, 0.35rad	2min, 23 s, 05ms	X = 1.692237m Y= 1.687016m Θ= 0.3923224 rad / 22.478°
2	700mm/s	0.75 rad/s	0.004, 0.007	150mm, 0.35rad	1min, 35 s, 60ms	X = 1.682160m Y= 1.682091m Θ= 0.7901489 rad / 45.272°

Obteniendo para el escenario 1 un error de posición meta menor al del escenario 2, para el escenario 1 se tiene 0.007m para "x" y 0.012m para "y", para el escenario 2 un error de 0.017m para "x" y 0.018m para "y".

Para ambos escenarios se utilizaron los mismos parámetros, pero la simulación se realizó en distintos tiempos llevándose más en el primer escenario que según la estimación de posición está más cerca del objetivo situado a 1.7m respecto de cada eje, también se puede observar que el ángulo de orientación theta es diferente en ambas estimaciones en este punto se observa que en el escenario 1 de menor ángulo brinda una mejor estimación de posición aunque esto también podría cambiar en las siguientes pruebas.

A fin de perfeccionar la simulación se realizaron un total de 15 pruebas para cada escenario en cambios de parámetros de controladores, seguimiento de pared, evasión de obstáculos, tamaños de los gráficos, constantes y sigmas de las matrices de covarianza, velocidades entre otros, teniendo la tabla siguiente las 3 estimaciones más cercanas a la meta.

Tabla 12. 3 ejemplos de mejores resultados del EKF en pruebas iniciales

Prueba	Escenario	Tiempo	Posición final del robot
1	1	1min, 38 s, 01ms	X = 1.7025m Y= 1.70357m Θ= 0.382281 rad / 21.9°
	2	1min, 17 s, 11ms	X = 1.699144m Y= 1.6984m Θ= 0.788906 rad / 45.2°
2	1	1min, 47s, 95ms	X = 1.698992m Y= 1.699318m Θ= 0.3904122 rad / 22.37°
	2	1 min, 15s, 45ms	X = 1.7026m Y= 1.7027m Θ= 0.790704 / 45.3°
3	1	1min, 25s, 70ms	X = 1.702004m Y= 1.7011m Θ= 0.393973 rad / 22.57°
	2	1min, 8s, 91s	X = 1.7028m Y= 1.7003m Θ= 0.789811 rad / 45.25°

Los parámetros establecidos para proceder a comparar el algoritmo EKF con el de Kevin Passino están dados en la siguiente tabla, donde después de variar estos valores entre otros de la programación que no son específicos del filtro de Kalman extendido y por ello no se especifican, se considera que un máximo de 400mm/s se encuentra bien pues se probaron valores superiores y el sistema falla al encontrar las marcas, gira bruscamente entre otros por ello, se debe correr el programa en una velocidad lineal y angular baja, en tanto las constantes de la covarianza a priori presentaron buen resultado al igual que una distancia de 200 mm entre marcas y 15° para cambio de posición.

Tabla 13. Parámetros del EKF final

Vmax-Vmin	Wmax-Wmin	Qk-1 Constantes v, w	Rk Sigma L, φ
400mm/s	0.9 rad/s	0.0028, 0.0062	200mm, 0.26rad

El algoritmo para comparar es el formulado por Kevin Passino que es un algoritmo optimizado basado en el forrajeo de bacterias, básicamente lo que se hace es asociar un término o matrices a lo que son los obstáculos y la meta, generalmente bajo campos potenciales, asociando el peso del objeto, los puntos de sensores el radio de sensores entre otros.

Cabe decir que además de los dos escenarios previamente vistos se revisaron otros 3 donde 1 sería el cambio de posición de la meta a uno de los cubículos del escenario 1, otro sería la meta en otra posición y con 3 obstáculos o paredes y el último un escenario sin obstáculos para observar la trayectoria del robot en un ambiente sin nada que le impida avanzar.

Comparaciones

Escenario 1

Uno de los escenarios que se han venido desarrollando para la optimización de la estimación de la posición del robot, presenta un robot en el eje (x, y) partiendo del punto (-1.7m, -1.7) m hasta un punto al otro extremo (1.7,1.7) m a continuación los resultados de aplicar ambos algoritmos para el seguimiento de trayectoria de un robot tipo diferencial Pioneer P3DX.

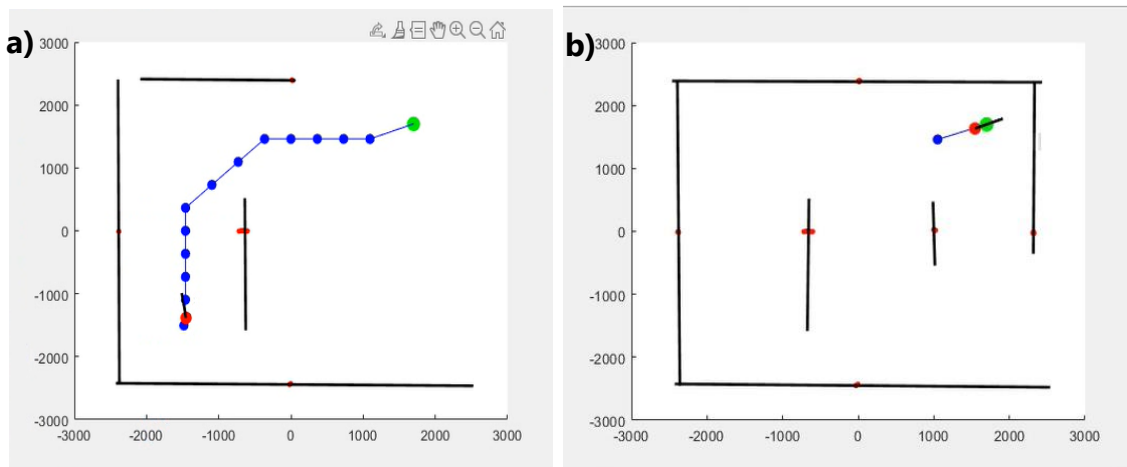


Figura 23. Mapeo en escenario 1 con EKF

Para la generación de la trayectoria en color azul se hace uso de una heurística, conforme el robot avanza y llega a cada punto esperado la trayectoria se reinventa de acuerdo a los obstáculos que se vayan detectando, en tanto la estimación de Kalman es bastante aproximada a la real y las paredes son censadas y evitadas correctamente trazando la trayectoria a una distancia considerada para que el robot no choque con las paredes y ejecute la evasión desde antes, en la (figura 23) la parte a) representa el inicio de la segunda generación de trayectoria cuando el robot ya ha detectado una pared enfrente de él y se dirige al segundo punto de referencia, en la parte b) se puede ver al robot casi llegando a la meta y detectando el resto de paredes, se observa también como el robot llega al punto que estimo como meta.

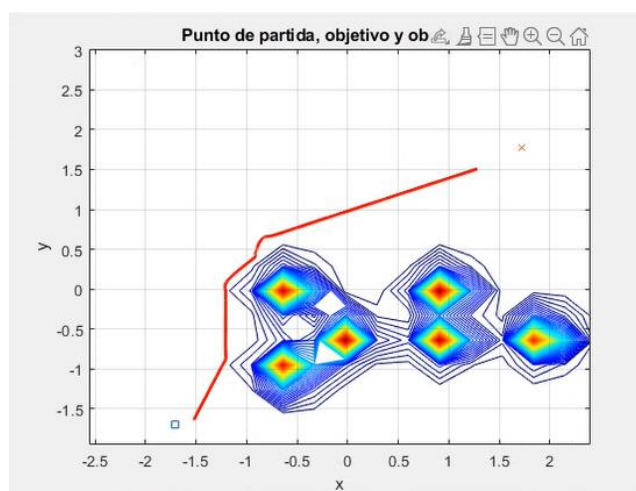


Figura 24. Mapeo en escenario 1 con algoritmo de Kevin Passino

En tanto que para el mapeo con el algoritmo de Passino se representa a los obstáculos en este caso las paredes, según el peso asignado por una especie de manchas, donde por las dimensiones de estas pareciese abarcan más espacio del que deberían así que no se asignó peso a las paredes de alrededor para que el robot pudiese hacer el recorrido, la posición inicial del robot se representa por un cuadrado azul y la meta por una equis en rojo, se observa el recorrido del robot bastante cerca de uno de los obstáculos y esto es debido a que el robot choco contra la esquina de la pared, aunque se reconoce que al poder salir del choque contra la pared el robot siguió su ruta sin desorientarse, también el algoritmo reconoce llegar al objetivo antes que en el EKF esto se observa mejor en las siguientes imágenes en CoppeliaSim.

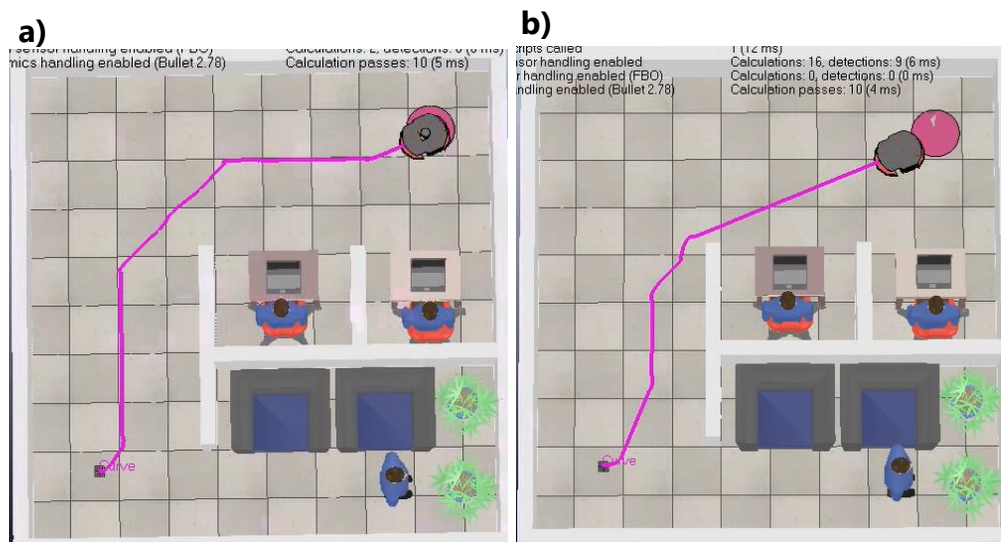


Figura 25. Meta alcanzada y trayectoria en escenario 1

En la (figura 25) se observa la trayectoria del robot en CoppeliaSim, para observar esto se agregó una graph que básicamente grafica la posición del robot en (x,y,z), se observa en la parte a) con el EKF que el robot se dirige a la meta alejada de los obstáculos y sin los problemas de la primera prueba, también se observa que el robot se coloca sobre el objetivo para llegar lo más cerca posible a la coordenada de meta, en la parte b) con el algoritmo de Kevin Passino se observa la trayectoria bastante cerca de la pared esto sucedió cuando este choca contra la pared y se reincorpora en su ruta, también en esta parte se detecta el objetivo antes de llegar a este pues el robot no se coloca sobre el sino, que reconoce haber llegado a las coordenadas dadas antes brindando un punto meta un poco distinto al solicitado.

Tabla 14. Comparación entre resultado de los algoritmos en escenario 1

Algoritmo	Tiempo	Posición final del robot	Error de posición
EKF	1min, 32 s, 15ms	X = 1.662896m Y= 1.683970m $\Theta = 0.4107626 \text{ rad} / 23.53^\circ$	X=0.037m Y=0.016m
Kevin Passino	1min, 15 s, 70ms	X = 1.301435m Y= 1.517161m $\Theta = 0.4038858 \text{ rad} / 23.14^\circ$	X=0.398m Y=0.182m

En resumen, para el escenario 1, en el EKF se brinda una estimación de posición más cercana a la real al comparar los datos de error de la tabla anterior, aunque el tiempo de simulado sea un poco mayor, también se evitan los obstáculos bastante bien y en el mapa se logra reconocer a las paredes en la posición que estas están, para Kevin Passino el estimado de posición es mucho menor al real aunque el tiempo de simulación también es menor, en este existen problema de choque en las paredes las que se representan por pesos y abarcan un espacio mayor al real.

Escenario 2

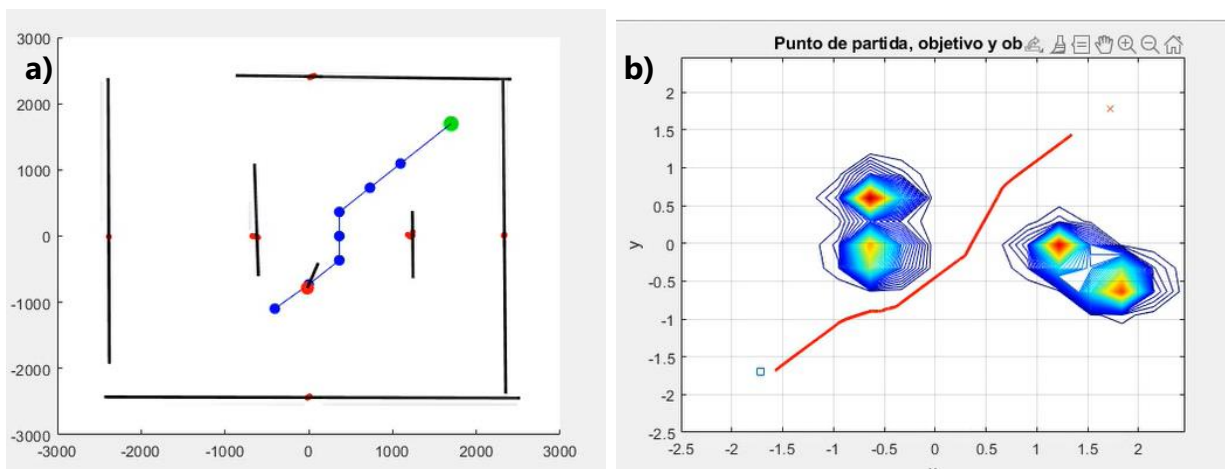


Figura 26. Mapeo en escenario 2

Para el escenario 2 se mantiene el punto de salida y de meta, pero se abre paso para la generación de otra trayectoria distinta a la primera, en la (figura 26) en la parte a) se observa el mapa con EKF y en la parte b) el mapa con Kevin Passino, al igual que en el primer escenario el robot con el algoritmo de Kevin Passino choca con la primera pared, pero se reincorpora bastante bien se observan las cuatro paredes de obstáculo en forma de peso y en EKF estas se representan mejor por líneas que no sobrepasan las dimensiones de estas.

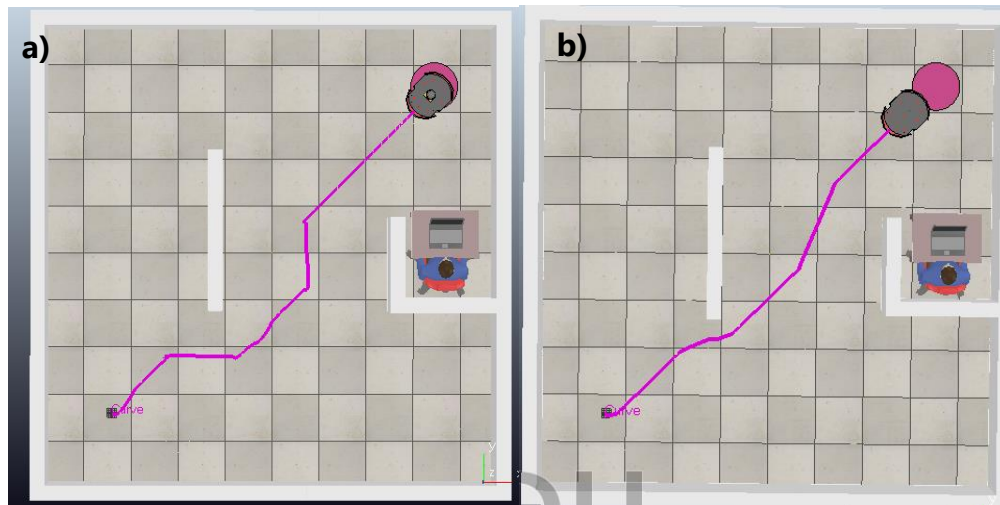


Figura 27. Meta alcanzada y trayectoria en escenario 2

En CoppeliaSim al finalizar la simulación se logra ver mejor la trayectoria completa para la parte a) con EKF se presenta de igual forma a la escena 1 una trayectoria que evita los obstáculos y posicionándose sobre la parte donde está la meta alcanzando una estimación de posición más cercana a la esperada, en la parte b) con Kevin Passino la trayectoria se observa más limpia, pero también choca con la primera pared y llegando a la meta en visualmente antes del punto esperado.

Tabla 15. Comparación entre resultado de los algoritmos en escenario 2

Algoritmo	Tiempo	Posición final del robot	Error de posición
EKF	1min, 14 s, 65ms	X =1.703606m Y= 1.703885m $\Theta = 0.784311 \text{ rad} / 44.93^\circ$	X=-0.003m Y=-0.038m
Kevin Passino	1min, 07 s, 44ms	X =1.376548m Y= 1.456758m $\Theta = 0.7856204\text{rad} / 45.01^\circ$	X=0.323m Y=0.243m

En el escenario 2 la estimación para EKF no es menor como en el escenario 1 si no que se pasa un poco del punto meta y por ello el error es negativo, pero siempre siendo un valor aceptable y dentro del objetivo en un buen tiempo y menor al presentado por este algoritmo en la prueba anterior, para el Algoritmo de KP la estimación es considerablemente menor al real y al de EKF, aunque el tiempo de simulación también es menor a todas las simulaciones anteriores de EKF y de él mismo en la prueba 1, lo parecido en los dos es el ángulo al que llegan a la meta.

Escenario 3

Inicialmente se pretendía comparar ambos algoritmos en los dos escenarios anteriores, pero para validar mejor las observaciones hechas a ambos resultados se añadieron 3 escenarios más, siendo el tercero una variante del primero con la diferencia de que en este la meta estará situada dentro de uno de los cubículos para probar como actúa el robot al tener que entrar a un espacio con las paredes más cerca del objetivo, el que se encuentra en el punto (0.400, 0.375) m.

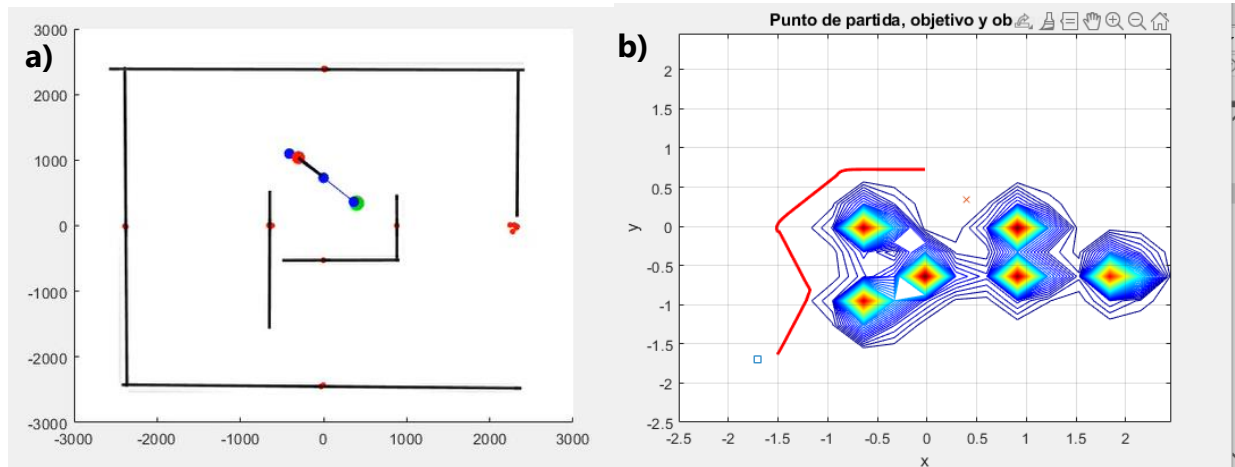


Figura 28. Mapeo en escenario 3

En la (figura 28) se presenta el mapeo del escenario 3 el cual es bastante similar al primero con la diferencia que el objetivo está dentro del primer cubículo, en la parte a) con EKF es aprecia la estructura del cubículo y la meta más cerca de una de las paredes a propósito, en la parte b) con KP la trayectoria inicial es diferente al primer escenario haciendo que el robot ya no choque con la pared, las que siguen representadas por un tipo de manchas bastante grandes, el problema en este escenario no es que al igual que los dos escenarios anteriores no se llegue directamente a la meta real si no que el robot no logra entrar al cubículo y pretende haber llegado a la meta con un poco más de diferencia a los anteriores, se observa mejor en las siguientes imágenes de la simulación en CoppeliaSim.

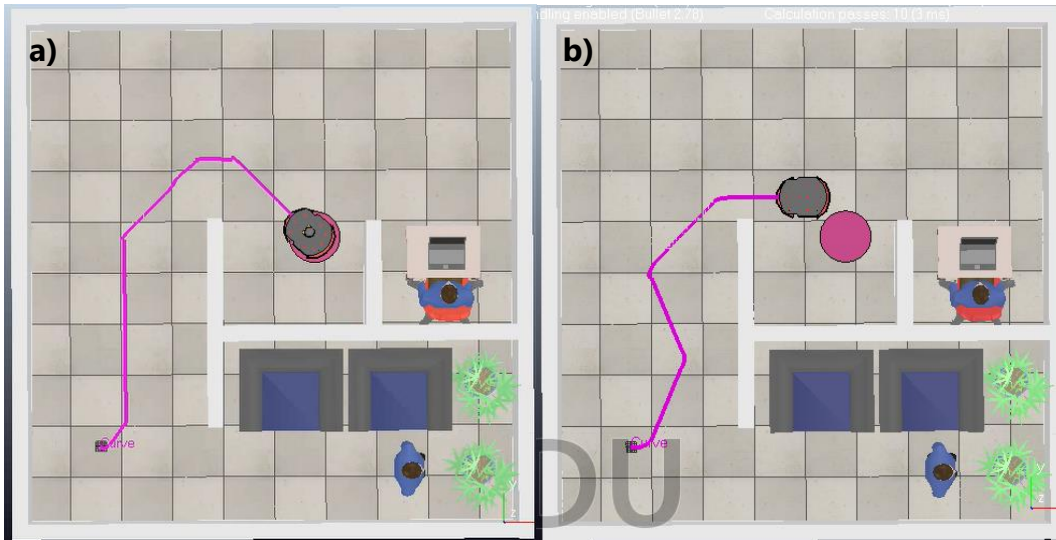


Figura 29. Meta alcanzada y trayectoria en escenario 3

En la (figura 29) en la parte a) con EKF se logra esquivar bastante bien los obstáculos y entrar hasta donde se encuentra el objetivo, cambio para la parte b) con KP, aunque bien ya no se choca en la pared si pasa bastante cerca de esta y no logra entrar al cubículo si no que el robot toma como terminada su ruta fuera de este y por ende el punto al que llega es considerablemente diferente al real como al estimado por el EKF.

Tabla 16. Comparación entre resultado de los algoritmos en escenario 3

Algoritmo	Tiempo	Posición final del robot	Error de posición
EKF	1min, 05 s, 45ms	$X = 0.3366m$ $Y = 0.3931m$ $\Theta = -0.77793 \text{ rad} / -44.57^\circ$	$X = 0.063m$ $Y = -0.018m$
Kevin Passino	56 s, 24ms	$X = 0.0113m$ $Y = 0.6588m$ $\Theta = -3.56^{-4} \text{ rad} / -0.020^\circ$	$X = 0.388m$ $Y = -0.283m$

En este escenario la diferencia entre ambos algoritmos se hace mucho más notable pues en las simulaciones anteriores, aunque en EKF la estimación es más cercana los ángulos si eran bastante similares, pero en el cambio de la posición del objetivo el EKF se aproxima bastante bien a la posición real con una ligera variación un poco más grande en el eje "y" que en el "x" y un ángulo negativo pues tuvo que girar para entrar al cubículo, en el de KP esto no sucede el robot no entra al cubículo por lo que sus valores en "x e y" son en el primero menores al real y en el segundo mayores, si se presenta un ángulo negativo que representaría que el robot entra al sitio pero es ligeramente pequeño.

Escenario 4

Para el escenario 4 no se hicieron variaciones de los escenarios previos si no que es totalmente diferente, en él se presentan 3 paredes como obstáculos principales, la meta se encuentra en el punto (0, 1.5) m, también en la ejecución del programa para ambos algoritmos se generaron 2 rutas.

Mapas

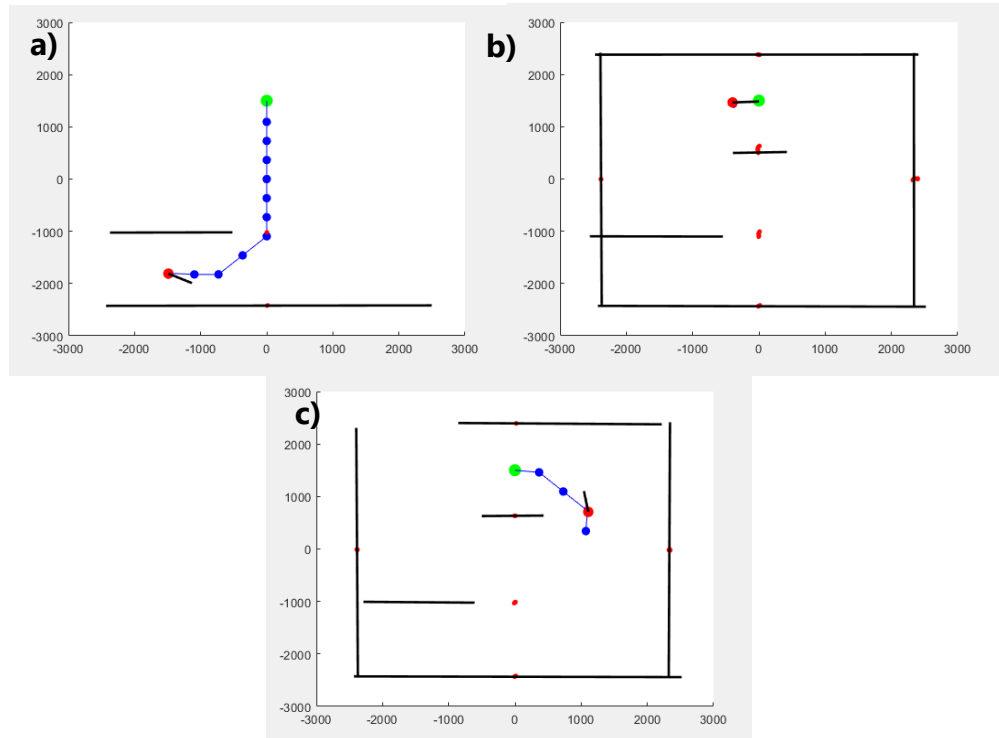


Figura 30. Mapeo en escenario 4 con EKF

Para el mapeo del escenario 4 se alcanza un punto que es cuando el robot detecta la pared frente al objetivo y se pueden generar dos rutas iguales por ambos lados, de la (figura 30) en la parte a) se muestra la ruta inicial sin haber detectado la pared de arriba, en la parte b) la ruta por la parte izquierda de la pantalla y en la c) por la parte derecha de la pantalla, al final ambas rutas estiman al mismo punto con distinta trayectoria una en forma de "s" y la otra de un "3" inclinado.

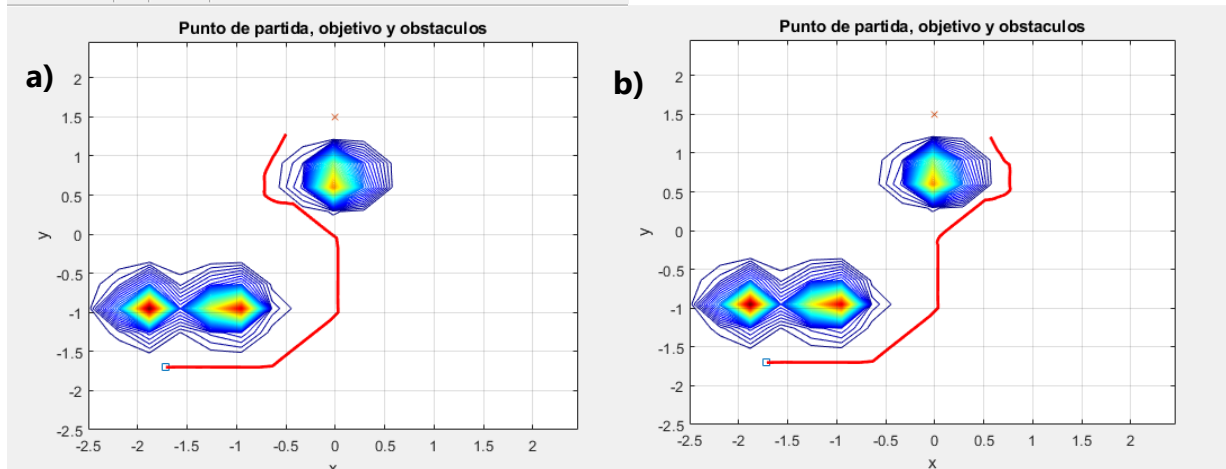


Figura 31. Mapeo en escenario 4 con algoritmo de Kevin Passino

Para el mapeo en el de KP también se presentan 2 rutas y en ambas el robot choca en la esquina de la pared, aunque para la ruta de la izquierda se ve más cerca del objetivo, lo que si cambia en este escenario respecto de los demás es que en la gráfica el robot sale exactamente del cuadrado azul, en la (figura 31) la parte a) es la ruta No.1 por la izquierda con el algoritmo de Kevin Passino y la parte b) es la ruta No. 2 por la derecha con Kevin Passino.

Escenarios

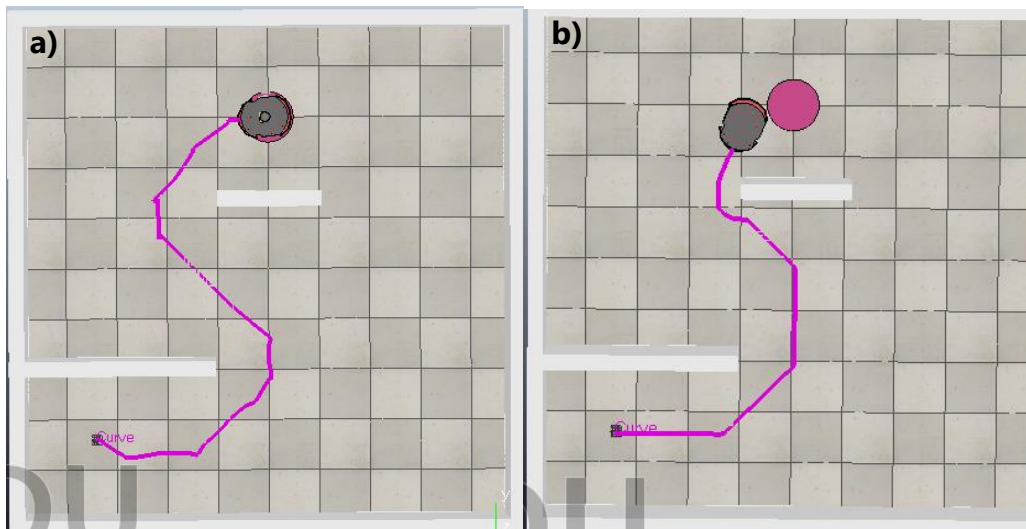


Figura 32. Meta alcanzada y trayectoria en escenario 4, ruta izquierda

Primero se comparan las rutas de la parte izquierda para ambos algoritmos, en la (figura 32) en la parte a) con el EKF se sigue observando la evasión que se hace de las paredes y como se traza una ruta alejada de estos además de que se posiciona sobre el objeto brindando una

estimación más cercana, en la parte b) con el de KP al inicio la ruta está alejada de las paredes, aunque con la última pared choca y llega a un extremo del objetivo.

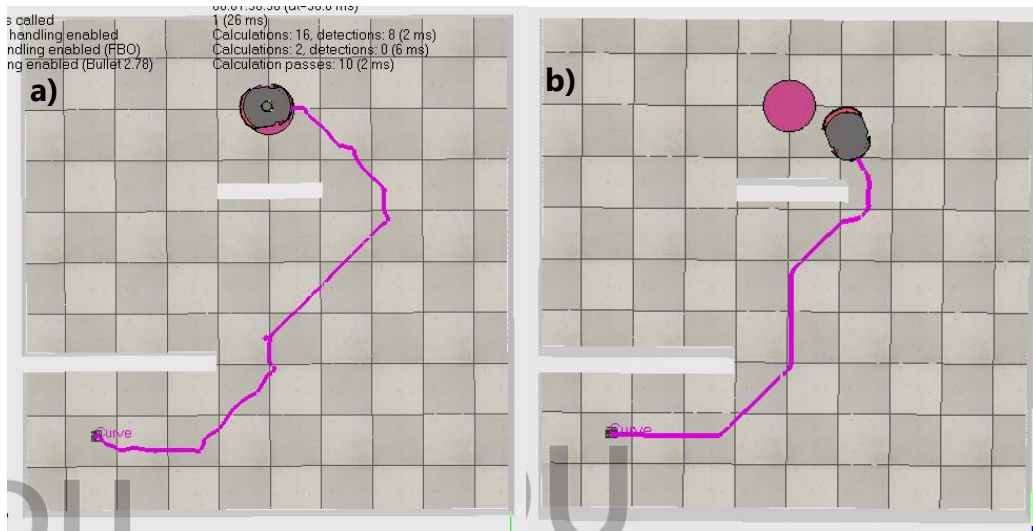


Figura 33. Meta alcanzada y trayectoria en escenario 4, ruta derecha

En la (figura 33) se muestra las rutas por la parte derecha de ambos algoritmos, en la parte a) con el EKF la trayectoria muestra al robot esquivando bien las paredes, pero con algunas turbulencias, aunque esto no afecta la estimación de la posición final del robot, en tanto para la parte b) con el de KP el robot esta vez no toca si quiera el borde del objetivo y da por hecho que llego a él después de salir de su choque con la pared.

Tabla 17. Comparación entre resultado de los algoritmos en escenario 4

Algoritmo	Ruta	Tiempo	Posición final del robot	Error de posición
EKF	Izquierda	1min, 34 s	X = -0.034m Y= 1.495m $\Theta = 0.1085 \text{ rad} / 6.22^\circ$	X=0.034m Y=0.005m
	Derecha	1 min, 56s, 30ms	X = -0.037m Y= 1.508m $\Theta = 3.0836 \text{ rad} / 176.68^\circ$	X=0.037m Y=-0.008m
Kevin Passino	Izquierda	1min, 19s, 65ms	X = -0.451m Y= 1.293m $\Theta = 1.1770 \text{ rad} / 67.44^\circ$	X=0.451m Y=0.207m
	Derecha	1min, 35s, 11ms	X = 0.483m Y= 1.3762m $\Theta = 1.9624 \text{ rad} / 112.43^\circ$	X=-0.483m Y=0.123m

En esta iteración es preciso observar el resultado dado en las rutas con su mismo algoritmo y en las rutas con el opuesto, por ejemplo, se observa en el EKF un tiempo de simulado mayor para la ruta por la derecha que es la que presento turbulencias respecto a su contraria, pero la posición es bastante similar, Para el de KP de igual forma se presenta un menor tiempo de simulado en la ruta izquierda respecto a la derecha y respecto a EKF, en lo que

si están un tanto diferente es en la posición que por lo que se menciona anteriormente en la ruta izquierda está más cerca desde "x" y en la ruta derecha está más cerca a la meta desde "y". En cuanto a ambos algoritmos al igual que los demás escenarios presenta mejor resultado de posición y evasión de obstáculos el EKF con un error de posición menor en tanto que el de KP presenta menor tiempo de simulación con un error mayor al de su contraparte.

Escenario 5

Después de 4 comparaciones es claro que algoritmo presenta mejores resultados y cuáles son las ventajas de uno respecto del otro, pero se tomó a bien una última comparación en un escenario totalmente vacío únicamente con las paredes de alrededor para determinar el comportamiento del robot si nada que le impidiera llegar a su objetivo, las coordenadas del punto meta son las mismas al escenario 1 y 2, (1.7, 1.7) m.

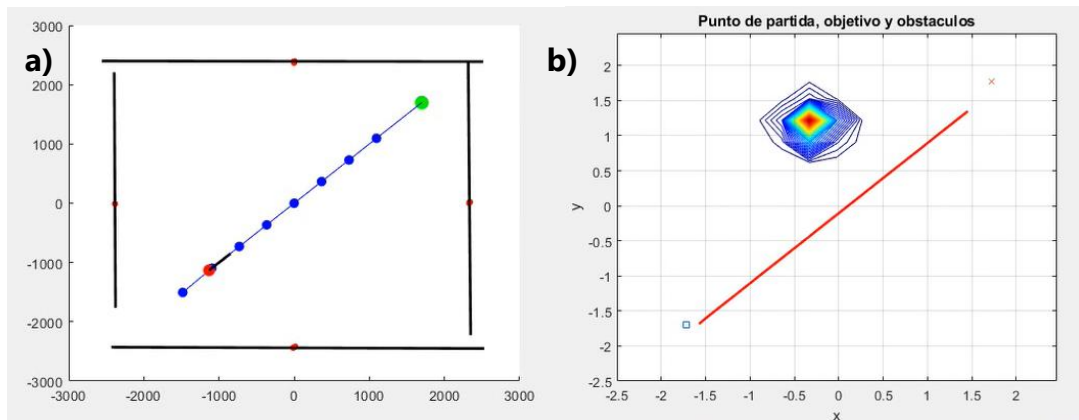


Figura 34. Mapeo en escenario 5

Para el mapeo de un escenario sin obstáculos que se muestra en la (figura 34 y 35) en la parte a) con EKF se traza una línea derecha hasta el objeto y se detectan las paredes, en la parte b) con el de KP debido a que no se agregaron las paredes como obstáculos por el tamaño en el que se estiman siempre se visualiza un peso de parte del programa donde realmente no hay nada, pero también se genera una línea derecha hasta poco antes de la meta, visualizándose mejor el resultado en las escenas de CoppeliaSim a continuación.

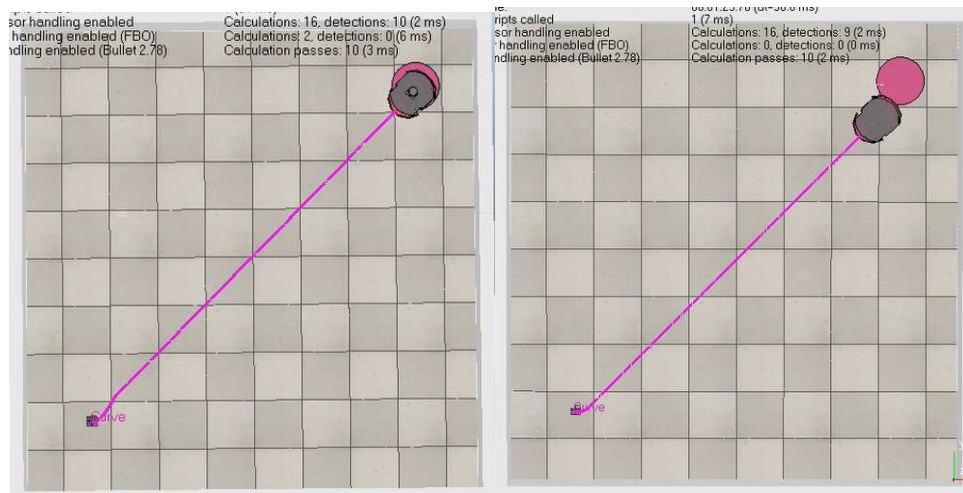


Figura 35. Meta alcanzada y trayectoria en escenario 5

Ambas simulaciones dan como resultado una línea sin turbulencias, pero al igual que en las anteriores con el EKF el robot llega hasta un punto bastante cerca al real, en tanto que con el de KP siempre el robot cree haber llegado a la meta mucho antes del punto real presentando también una ventaja en cuanto al tiempo de simulación respecto al EKF donde las velocidad lineal máxima es de 400mm/s o 0.4m/s debido a que velocidades mayores lo desorientan y le dificultan los giros y en el de KP la velocidad máxima es de 1.2m/s es decir la que el robot puede alcanzar según los datos de fabricación.

Tabla 18. Comparación entre resultado de los algoritmos en escenario 5

Algoritmo	Tiempo	Posición final del robot	Error de posición
EKF	1min, 05s, 45ms	X = 1.6788m Y= 1.6889m $\Theta = 0.7912 \text{ rad} / 44.57^\circ$	X=0.021m Y=0.011m
Kevin Passino	1min, 25s, 70ms	X = 1.4633m Y= 1.3551m $\Theta = 0.78403 \text{ rad} / 45.33^\circ$	X=0.236m Y=0.344m

Para un escenario sin obstáculos el error de posición disminuye respecto al error obtenido en la simulación del escenario 1.

Evaluación

Se logró la comprobación del modelo matemático basado en el EKF para un robot móvil de tipo diferencial propuesto en la iteración 3 de la metodología, mediante el uso de Matlab y CoppeliaSim, conectando estos dos programas mediante una API para su posterior comparación con el algoritmo de Kevin Passino para la generación de trayectoria basada en la asignación de matrices de costo, obteniendo mejores resultados de estimación de posición y evasión de obstáculos del EKF frente al de KP que presenta mejor tiempo de simulado. Llegando al final de

la metodología con un 100% de esta completada gracias al análisis minucioso de los riesgos presentes en cada etapa y a la corrección de estos.

5.3 DISCUSIÓN

En esta etapa se hace la comparación entre el funcionamiento del modelo matemático planteado a partir del EKF en un robot Pioneer P3DX con el algoritmo propuesto por Kevin Passino a partir de su ya conocido algoritmo para el forrajeo bacterial, generando un vaivén de ventajas y desventajas de cada uno.

Para la investigación se utilizaron una serie de recursos de estudios hechos previamente como en (Stefek et al., 2020) donde lo que se busca es seleccionar el mejor controlador para un robot diferencial a partir de la cinemática y la dinámica de este, en cambio en (Bussi et al., 2017) se proponen dos estrategias de control del robot diferencial a partir de su cinemática y utilizando el EKF como estimador de estado con una serie de obstáculos fijos, también en (Toriz et al., 2010) se presenta al robot Pioneer P3DX construyendo un mapa de forma abstracta mientras utiliza este mismo para su localización integrando al EKF junto un SRT o árbol aleatorio basado en sensores. En el caso del filtro de Kalman extendido uno de los miedos y cuidados que se tiene a la hora de aplicarlo es referente al cálculo de los jacobianos los cuales si no son correctos tampoco lo será la estimación de posición o el seguimiento de trayectoria, también se debe cuidar la aplicación de este en sistemas altamente lineales ya que el tal no converge y brinda estimaciones no reales tal como se encuentra en (Du et al., 2018), donde se aconseja utilizar este filtro en sistemas de baja intensidad no lineal y bajo ruido de observación. Por ello en este estudio se logró presentar el excelente funcionamiento del filtro en escenarios controlados para el seguimiento de distintas trayectorias y la estimación de la posición del robot con poco ruido en el proceso, la principal aportación de esta investigación ante la gran utilización del filtro es que con la presentación del modelo matemático final se pudo desglosar un conjunto de fórmulas matemáticas y matrices de una formas más fácil para su aplicación, demostrando que a partir de la cinemática de un robot diferencial se puede estimar la posición de este así como que el modelo matemático puede fácilmente ser utilizado en otros modelos de robot parecidos al Pioneer P3DX obteniendo resultados bastante satisfactorios que se ven evidenciados en las simulaciones de este, sin presentar problemas con el cálculo de jacobianos

o con el sistema en general, en el caso de las fórmulas de predicción y corrección de EKF coinciden con las utilizadas en la investigación de (Fahmedha et al., 2015) y de (Toriz et al., 2010), al final la aplicación del EKF obtiene mejores resultados que su contraparte el algoritmo de Kevin Passino el cual aunque es eficiente en cuestión de tiempo no obtiene los resultados que se buscaban de estimación de posición, evasión de obstáculos y seguimiento de trayectoria, por lo que si estos se utilizaran para trasportar algo en la utilización del algoritmo de Kevin Passino se corre el riesgo de dañar o dejar caer el producto cuando este choca con algunas de las paredes, esta comparación puede ayudar a las investigaciones futuras a determinar que algoritmo es más conveniente para lo que se está buscando, si lo que se desea es rapidez en el proceso o lo que se desea es que el proceso realmente se haga bien y de forma correcta, recalcando la buena estimación que en esta investigación se obtuvo en cada uno de los escenarios de parte del EKF.

CAPÍTULO VI. CONCLUSIONES

Se logró obtener un modelo matemático a partir de la cinemática del robot Pioneer P3DX de tipo diferencial utilizando el algoritmo de filtro de Kalman extendido para el seguimiento de trayectoria desde la posición inicial hasta un punto meta, así como la estimación de la posición del robot obteniendo su orientación dada por θ en radianes y sus coordenadas en (x, y) en metros, también se calcularon 3 jacobianos es decir, las derivadas parciales del vector de estado, señal de ruido y vector de medida respectivamente y dos matrices de covarianza característicos de la etapa de predicción y corrección del EKF.

Se realizó un diagrama cinemático en 2 dimensiones del robot Pioneer P3DX a partir de las medidas en la hoja del fabricante tanto en una posición inicial como en un punto después en el tiempo para la actualización de valores y estado general que se vinculan al filtro de Kalman extendido como el vector de estado que representa la posición y orientación del robot (x, y, θ) y los cuales dependen directamente de la corrección que se haga en este donde la distancia mínima entre una posición y otra es de 200mm y el ángulo de 0.26 rad, su matriz de control que está dada por la velocidad lineal máxima es de 400mm/s ya que con velocidades superiores la simulación presenta turbulencias y una velocidad angular máxima para el control del robot es de 0.9 rad/s.

Se comparó el rendimiento de ambos algoritmos encontrando al filtro de Kalman extendido superior al de Kevin Passino en seguimiento de ruta, evasión de obstáculos y estimación de la posición más aproximada a la real, el EKF obtuvo un error de 0.037m para la posición en "x" y 0.016m en "y" en cambio el algoritmo de Kevin Passino presento errores de 0.398m para "x" y 0.282m en "y" para el escenario 1, también en cuanto al mapa en EKF se aprecian mejor y de manera más limpia los obstáculos y la ruta que en el de Kevin Passino, en cambio en términos del menor tiempo de simulación el algoritmo de Kevin Passino supera al filtro de Kalman extendido con una diferencia de 16s : 45ms para el escenario 1 y 20s : 25ms en el escenario 5, ya que puede alcanzar la máxima velocidad permitida por el robot de 1200mm/s.

CAPÍTULO VII. RECOMENDACIONES

Para siguientes investigaciones para mejorar el problema que presenta el EKF en cuanto a velocidad baja se recomienda el poder utilizar un controlador de alto nivel o bien integrar el EKF con otro algoritmo de estimación que tome las ventajas de ambos para poder reducir el tiempo de simulación de este y por ende hacerlo más eficiente.

Se recomienda utilizar otros modelos de robots comerciales o industriales, ya sea de tipo diferencial o parecidos para comprobar el funcionamiento del modelo matemático y conocer los arreglos de cinemática, matrices y otros aspectos que se necesitarían cambiar para que el modelo matemático pueda ser adaptado a estos robots.

BIBLIOGRAFÍA

- Adept Technology Inc. (2021). *Pioneer P3-DX mobile robot*. Génération Robots.
<https://www.generationrobots.com/en/402395-robot-mobile-pioneer-3-dx.html>
- Al Malki, H. H., Moustafa, A. I., & Sinky, M. H. (2021). An Improving position method using Extended Kalman filter. *Procedia Computer Science*, 182, 28-37.
<https://doi.org/10.1016/j.procs.2021.02.005>
- Anbu, N. A., & Jayaprasanth, D. (2019). Integration of Inertial Navigation System with Global Positioning System using Extended Kalman Filter. *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 789-794.
<https://doi.org/10.1109/ICSSIT46314.2019.8987949>
- Antoulas, A., Georgiou, T. T., Khargonekar, P. P., Bulent Ozguler, A., Sontag, E. D., & Yamamoto, Y. (2017). Prof. Rudolf Emil Kalman [Obituary]. *IEEE Control Systems Magazine*, 37(1), 151-152. <https://doi.org/10.1109/MCS.2016.2621578>
- Aranda R, M. E., & Martínez H, J. M. (2017). *Estudio y aplicación del Filtro de Kalman en fusión de sensores en UAVs*. <https://idus.us.es/handle/11441/66071>
- Balico, L. N., Loureiro, A. A. F., Nakamura, E. F., Barreto, R. S., Pazzi, R. W., & Oliveira, H. A. B. F. (2018). Localization Prediction in Vehicular Ad Hoc Networks. *IEEE Communications Surveys Tutorials*, 20(4), 2784-2803. <https://doi.org/10.1109/COMST.2018.2841901>
- Bellar, A., & Mohammed, M. A. S. (2019). Satellite Inertia Parameters Estimation Based on Extended Kalman Filter. *Journal of Aerospace Technology and Management*.
<https://doi.org/10.5028/jatm.v11.1016>

- Benabdallah, M., & Bourza, M. (2018). A Convergence result for the Euler-Maruyama method of one-dimensional stochastic differential equations involving the local time of the unknown process at zero. *2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, 1-4.
<https://doi.org/10.1109/ICECOCS.2018.8610573>
- Bijo, S. (2015, agosto 25). Mobile Robotics. *Bijo Sebastian*.
<https://bijosebastian.wordpress.com/mobile-robotics/>
- Bussi, U., Mazzone, V., & Oliva, D. (2017). Control strategies analysis using EKF applied to a mobile robot. *2017 XVII Workshop on Information Processing and Control (RPIC)*, 1-6.
<https://doi.org/10.23919/RPIC.2017.8214361>
- Cardona, J. P., Leal, J. J., Ramírez, J. L., Cardona, J. P., Leal, J. J., & Ramírez, J. L. (2018). Modelado Matemático de la Posición del Centro de Masa de un Robot de Tracción Diferencial. Un Enfoque desde la Mecánica Lagrangiana. *Información tecnológica*, 29(6), 307-320.
<https://doi.org/10.4067/S0718-07642018000600307>
- Castañeda C., J. A., Nieto A., M. A., & Ortiz B., V. A. (2013). Análisis Y Aplicación Del Filtro De Kalman a Una Señal Con Ruido Aleatorio. *Scientia Et Technica*, 18(1), 267-274.
- Cevallos U., H. I., Intriago, G., Plaza, D., & Idrovo, R. (2018). The Extended Kalman Filter in the Dynamic State Estimation of Electrical Power Systems. *Enfoque UTE*, 9(4), 120-130.
<https://doi.org/10.29019/enfoqueute.v9n4.407>

- Chen, L., Shan, Y., Tian, W., Li, B., & Cao, D. (2018). A Fast and Efficient Double-Tree RRT^{*}-Like Sampling-Based Planner Applying on Mobile Robotic Systems. *IEEE/ASME Transactions on Mechatronics*, 23(6), 2568-2578. <https://doi.org/10.1109/TMECH.2018.2821767>
- Cüchango, H. E. E., & Esmeral, J. I. S. (2014). Algoritmo de optimización basado en enjambres de partículas con comportamiento de vorticidad y búsqueda individual y grupal. *Tecnura*, 18(42), 24-37. <https://doi.org/10.14483/udistrital.jour.tecnura.2014.4.a02>
- Darvishi, A., & Doroudi, A. (2019). Continuous Estimation of Speed and Torque of Induction Motors Using the Unscented Kalman Filter under Voltage Sag. *DYNA*, 86(208), 37-45. <https://doi.org/10.15446/dyna.v86n208a4.65567>
- Deng, X.-T. H. X.-F., Zeng, S.-Q. Z. Q. G. C.-X., & Zhou, K.-Q. (2020). Comparative Study on Application of Extended Kalman Filter and Unscented Kalman Filter in Target Tracking. *IOP Conference Series: Materials Science and Engineering*, 864, 012055. <https://doi.org/10.1088/1757-899X/864/1/012055>
- Du, Y., Yu, L., Gai, L., & Jiang, Y. (2018). Analysis of nonlinear processing ability of sequential Monte Carlo filter. *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 305-309. <https://doi.org/10.1109/ICCAR.2018.8384690>
- Elfatah, M. G. A., Zaky, H. N., & Gharib, M. (2019). Mobile robot position estimation using Euler-Maruyama algorithm. *IOP Conference Series: Materials Science and Engineering*, 610, 012074. <https://doi.org/10.1088/1757-899X/610/1/012074>

- Fahmedha, N., Prakash, P. C., Pooja, A., & Rachana, R. (2015). Estimation of System Parameters Using Kalman Filter and Extended Kalman Filter. *International Journal of Advanced Technology and Engineering Exploration*, 2(6), 84-89.
- Fernández, C., Pantano, N., Godoy, S., Serrano, E., & Scaglia, G. (2019). Optimización de Parámetros Utilizando los Métodos de Monte Carlo y Algoritmos Evolutivos. Aplicación a un Controlador de Seguimiento de Trayectoria en Sistemas no Lineales. *Revista Iberoamericana de Automática e Informática industrial*, 16(1), 89-99.
<https://doi.org/10.4995/riai.2018.8796>
- Freire C. (2009). Modelo matemático de un robot móvil. *Eídos*, 1, 3-7.
<https://doi.org/10.29019/eidos.v0i1.31>
- Georgiou, T. T. (2018). Rudolf E. Kalman's quest for algebraic characterizations of positivity. *Annual Reviews in Control*, 45, 205-206. <https://doi.org/10.1016/j.arcontrol.2018.01.001>
- Hacene, N., & Mendil, B. (2019). Motion Analysis and Control of Three-Wheeled Omnidirectional Mobile Robot. *Journal of Control, Automation and Electrical Systems*, 30(2), 194-213.
<https://doi.org/10.1007/s40313-019-00439-0>
- Hehenberger, P., Poltschak, F., Zeman, K., & Amrhein, W. (2010). Hierarchical design models in the mechatronic product development process of synchronous machines. *Mechatronics*, 20(8), 864-875. <https://doi.org/10.1016/j.mechatronics.2010.04.003>
- Hernández Millán, G., Ríos Gonzales, L. H., & Bueno López, M. (2016). Implementación de un controlador de posición y movimiento de un robot móvil diferencial. *Tecnura*, 20(48), 123-136.

- Jeong, H. Y., Song, B. D., & Lee, S. (2020). The Flying Warehouse Delivery System: A Quantitative Approach for the Optimal Operation Policy of Airborne Fulfillment Center. *IEEE Transactions on Intelligent Transportation Systems*, 1-10. <https://doi.org/10.1109/TITS.2020.3003900>
- Jiang, Y., & Nong, X. (2020). Trainable Kalman Filter Based on Recurrent Neural Network and its Application in Aviation Surveillance. *Journal of Physics: Conference Series*, 1642, 012010. <https://doi.org/10.1088/1742-6596/1642/1/012010>
- Jiang, Z., Yin, H., Luo, Y., Gong, J., Yang, Y., & Lin, M. (2019). Quantitative Analysis of Mobile Application User Interface Design. *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, 1-8. <https://doi.org/10.1109/IPCCC47392.2019.8958722>
- Larco Chiriboga, M. D., & Cortijo Leyva, R. E. (2019). IMPLEMENTACIÓN DE UNA PLATAFORMA ROBÓTICA MÓVIL DIFERENCIAL PARA SEGUIMIENTO DE TRAYECTORIA, MEDIANTE UN ALGORITMO DE CONTROL NO LINEAL. <http://repositorio.uisrael.edu.ec/handle/47000/1904>
- Lin, S.-L., & Wu, B.-H. (2021). Application of Kalman Filter to Improve 3D LiDAR Signals of Autonomous Vehicles in Adverse Weather. *Applied Sciences*, 11(7), 3018. <https://doi.org/10.3390/app11073018>
- Liu, C., Zhang, G., Guo, W., & He, R. (2020). Kalman Prediction-Based Neighbor Discovery and Its Effect on Routing Protocol in Vehicular Ad Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(1), 159-169. <https://doi.org/10.1109/TITS.2018.2889923>

- Madhukar, P. S., & Prasad, L. B. (2020). State Estimation using Extended Kalman Filter and Unscented Kalman Filter. *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, 1-4. <https://doi.org/10.1109/ICONC345789.2020.9117536>
- Mao, R., Gao, H., & Guo, L. (2019). Optimal motion planning of differential-drive mobile robots based on trapezoidal collocation method. *Journal of Physics: Conference Series*, 1341, 052007. <https://doi.org/10.1088/1742-6596/1341/5/052007>
- Mehta, B., Khurana, P., & Kumar, D. (2020). Deterministic and Stochastic Models-Population Growth Models Population Models. *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 212-216. <https://doi.org/10.1109/ICRITO48877.2020.9197769>
- Morales, A. (2021, febrero 18). Robots Móviles Autónomos (AMR): Ventajas, retos y aplicaciones. *CIO MX*. <https://cio.com.mx/robots-moviles-autonomos-amr-ventajas-retos-y-aplicaciones/>
- Noriega, G. M., Restrepo, J., Aller, J. M., Giménez, M. I., & Guzmán, V. M. (2010). Utilización del algoritmo de forraje bacterial para identificar en línea los parámetros de un sistema eléctrico. *Universidad, Ciencia y Tecnología*, 14(54), 045-054.
- Ortiz A., G., Andaluz, V. H., & Rosales, A. (2011). *Modelación, Identificación y Control de Robots Móviles*. <http://bibdigital.epn.edu.ec/handle/15000/4321>
- Passino, K. (2005). *Biomimicry for Optimization, Control, and Automation*. Springer-Verlag. <https://doi.org/10.1007/b138169>

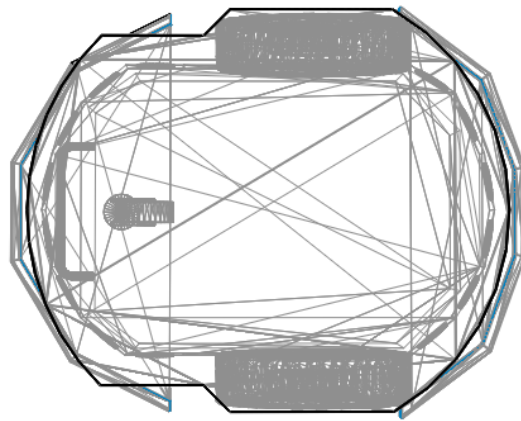
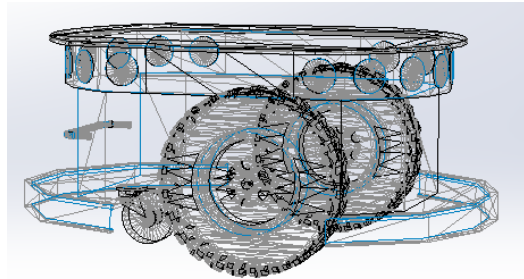
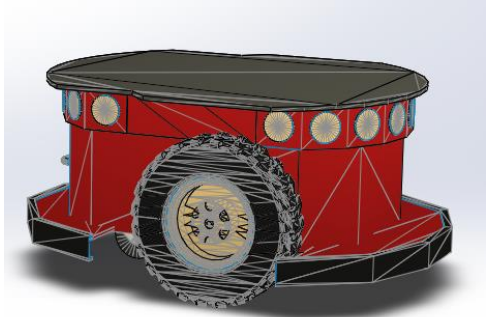
- Rachmawati, D., & Gustin, L. (2020). Analysis of Dijkstra's Algorithm and A* Algorithm in Shortest Path Problem. *Journal of Physics: Conference Series*, 1566, 012061. <https://doi.org/10.1088/1742-6596/1566/1/012061>
- Ren, X., Tan, L., Jiaqi, S., & Lian, X. (2021). Multi-target UAV path planning based on improved RRT algorithm. *Journal of Physics: Conference Series*, 1786(1), 012038. <https://doi.org/10.1088/1742-6596/1786/1/012038>
- Ríos G., L. H., & Bueno L., M. (2008). Modelo Matemático Para Un Robot Móvil. *Scientia Et Technica*, XIV (38), 13-18.
- Romero N., N. N., & Tuiran V., R. E. (2018). Optimization method of a six-bar steering mechanism formulated with natural coordinates. *DYNA*, 85(207), 168-173. <https://doi.org/10.15446/dyna.v85n207.71619>
- Ruiz Bravo, H. F. (2019). *Tutorial robot con tracción diferencial*. <https://www.youtube.com/watch?v=TTJT9G9dMOo>
- Shen, C., Zhang, Y., Guo, X., Chen, X., Cao, H., Tang, J., Li, J., & Liu, J. (2021). Seamless GPS/Inertial Navigation System Based on Self-Learning Square-Root Cubature Kalman Filter. *IEEE Transactions on Industrial Electronics*, 68(1), 499-508. <https://doi.org/10.1109/TIE.2020.2967671>
- Simon, D. (2006). Using nonlinear Kalman filtering to estimate signals. *Embedded Systems Design*, 19(7), 38-53. <https://doi.org/10.1115/1.1789153>
- Solaque, L. E., Velasco, A. E., & Riveros, A. (2018). Planificación de trayectorias por técnica de A* y suavizado por curvas de Bezier para la herramienta del sistema de remoción de maleza

- de un robot dedicado a labores de agricultura de precisión. *Entre Ciencia e Ingeniería*, 12(24), 43-51. <https://doi.org/10.31908/19098367.3814>
- Stefek, A., Pham, T. V., Krivanek, V., & Pham, K. L. (2020). Energy Comparison of Controllers Used for a Differential Drive Wheeled Mobile Robot. *IEEE Access*, 8, 170915-170927. <https://doi.org/10.1109/ACCESS.2020.3023345>
- Sun, Y., Fang, M., & Su, Y. (2021). AGV Path Planning based on Improved Dijkstra Algorithm. *Journal of Physics: Conference Series*, 1746, 012052. <https://doi.org/10.1088/1742-6596/1746/1/012052>
- Tan, A., Tan, Z., Tian, W., Yan, M., & Long, Y. (2019). Application of Kalman Filter in Deformation Data Processing of Ground-based Differential Interferometric Radar. *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, 1-5. <https://doi.org/10.1109/ICSIDP47821.2019.9173304>
- Toriz, A., Zapata, R., Sánchez, A., & Osorio, M. A. (2010). Integrated Exploration Based SRT-EKF. *2010 Ninth Mexican International Conference on Artificial Intelligence*, 171-176. <https://doi.org/10.1109/MICAI.2010.21>
- Tzafestas, S. G. (2014). *Introduction to Mobile Robot Control*. Elsevier. <https://doi.org/10.1016/C2013-0-01365-5>
- Tzafestas, S. G. (2018). Mobile Robot Control and Navigation: A Global Overview. *Journal of Intelligent & Robotic Systems*, 91(1), 35-58. <https://doi.org/10.1007/s10846-018-0805-9>

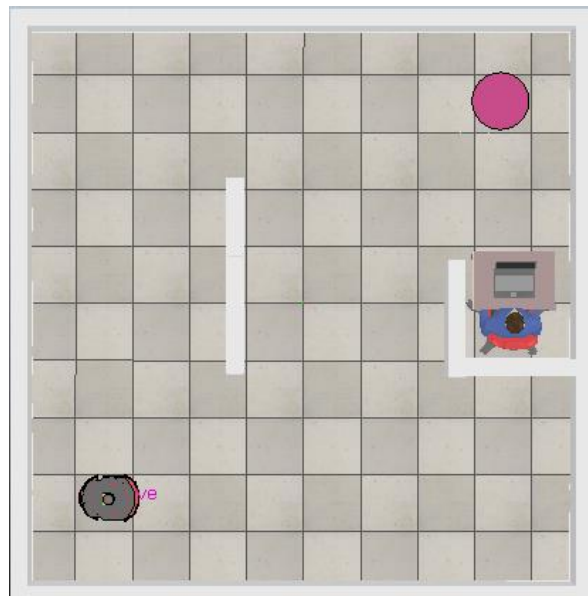
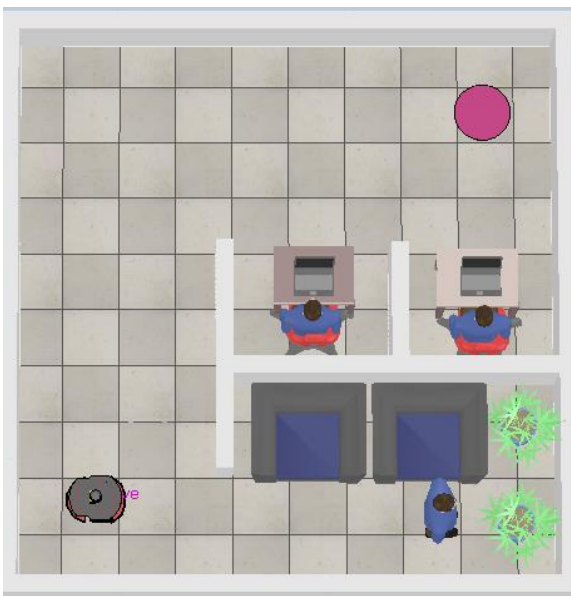
- Ulusoy, M. (2017). *Understanding Kalman Filters, part 5: Nonlinear State Estimators Video* [E]. MathWorks. <https://la.mathworks.com/videos/understanding-kalman-filters-part-5-nonlinear-state-estimators-1495052905460.html>
- Velásquez, S. M., Montoya, J. D. V., Adasme, M. E. G., Zapata, E. J. R., Pino, A. A., & Marín, S. L. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Revista CINTEX*, 24(2), 13-23. <https://doi.org/10.33131/24222208.334>
- Viveros, R. A., Yuz, J. I., & Pérez-Ibacache, R. R. (2014). Estimación simultánea de estado y parámetros para un sistema no lineal variante en el tiempo. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 11(3), 263-274. <https://doi.org/10.1016/j.riai.2014.05.001>
- Zhang, T. (2018). EM based estimation for general linear state space systems. *2018 Chinese Control and Decision Conference (CCDC)*, 3100-3105. <https://doi.org/10.1109/CCDC.2018.8407657>
- Zhao, S., Gu, J., Ou, Y., Zhang, W., Pu, J., & Peng, H. (2016). IRobot self-localization using EKF. *2016 IEEE International Conference on Information and Automation (ICIA)*, 801-806. <https://doi.org/10.1109/ICInfA.2016.7831929>
- Zhu, J., Hu, Y., Li, Y., Ming, A., Yang, Z., & Zhang, W. (2019). Application of adaptive square root cubature Kalman filter in turbofan engine gas path performance monitoring. *IOP Conference Series: Earth and Environmental Science*, 267, 042073. <https://doi.org/10.1088/1755-1315/267/4/042073>

ANEXOS

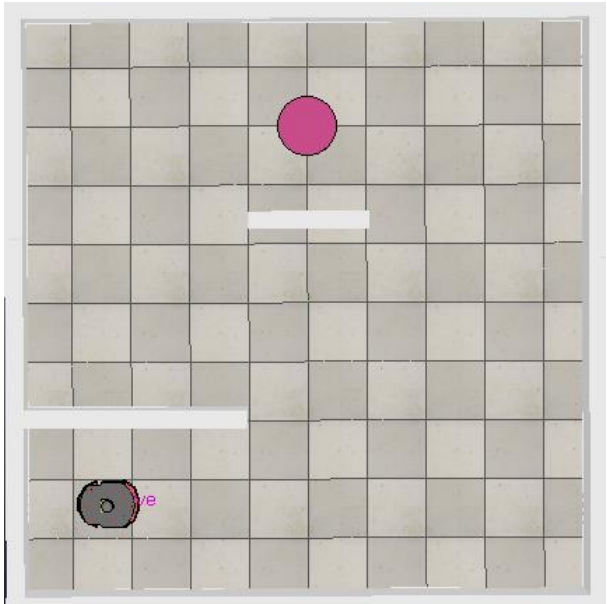
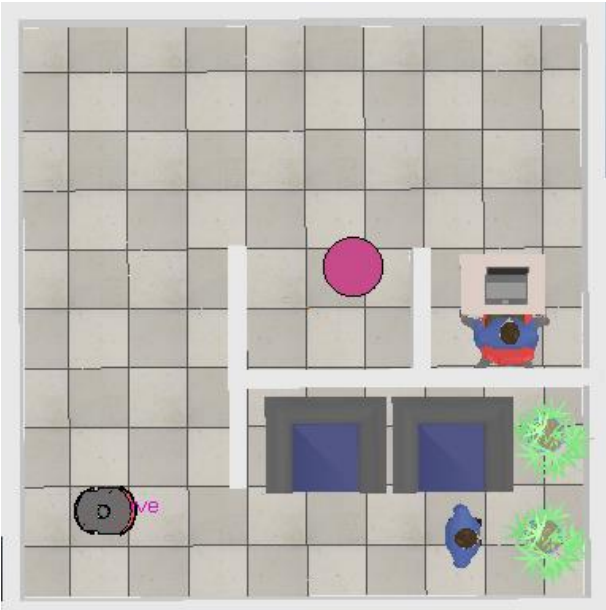
ROBOT PIONEER P3DX EN SOLIDWORKS



ESCENARIO 1 Y 2



ESCENARIO 3 Y 4



ESCENARIO 5

