Técnicas de slam con filtros probabilísticos; caracterización y resultados en robots móviles

Slam techniques with probabilistic filters; characterization and results in mobile robots

[a]Franklin Pineda-Torres

[a]Magister en Ingeniería Electrónica y de Computadores,
franklin.pineda@fuac.edu.co, Orcid: 0000-0003-1790-464X

Recibido: Enero 10 de 2019 Aceptado: Mayo 20 de 2019.

Forma de citar: F. Pineda-Torres,
"Técnicas de slam con filtros probabilísticos; caracterización y resultados en robots móviles",
Mundo Fesc, vol. 9, no.18, pp. 7-15, 2019.

Resumen

El desafío de la localización y mapeo SLAM (Simultaneous Localization And Mapping) en robots móviles consiste en descubrir si es posible navegar a través de un entorno desconocido y construir de manera incremental un mapa consistente del mismo, mientras que determina al mismo tiempo su posición dentro del mapa. La solución teórico-conceptual ha sido desarrollado por el SLAM especialmente con filtros de localización, tales como: el filtro de Kalman, el filtro de información, el filtro gráfico y el filtro de partículas entre otros. El estudio realizado al variar marcas establecidas en las trayectorias y analizar estos filtros desde el punto de vista probabilístico que se encuentran incorporados en dos prototipos de robots móviles que poseen sensores de ultrasonido y láser, muestra resultados de errores en odometría y tiempos necesarios que cada filtro SLAM tiene en promedio por iteración para la construcción del mapa bidimensional.
Palabras clave: Filtro de Kalman, localización, robot móvil, sensor, SLAM.

Abstract

Abstract: The challenge of locating and mapping SLAM in mobile robots is to discover if it is possible to navigate through an unknown environment and incrementally build a consistent map of it, while the robot at the same time determining its position within the map. The theoretical-conceptual solution has been developed by the SLAM especially with location filters, such as: the Kalman filter, the Information filter, the graphic filter and the particle filter among others. The research carried out by varying established marks in the trajectories and analyzing these filters from the probabilistic point of view, these filters are incorporated in two prototypes of mobile robots, which have ultrasound and laser sensors. The article shows the results of errors in odometry and necessary times that each SLAM filter has on average per iteration for the construction of the two-dimensional map.
Keywords: Kalman Filter, localization, mobile robot, sensor, SLAM.

Autor para correspondencia:

*Correo electrónico:franklin.pineda@fuac.edu.co

Introducción

Las preguntas fundamentales de un robot móvil son detalladas sobre tres procesos: planeación, localización y navegación. En la localización el robot se pregunta ¿dónde estoy? y ¿hacia dónde me dirijo?, en la planeación cuál es la mejor manera de llegar?, y en la navegación ¿cómo llego rápido? y ¿cómo evito obstáculos? La localización ha tenido gran investigación en la última década [1] debido a los avances tecnológicos que han resultado del desarrollo de microcontroladores más rápidos y de sistemas de sensado más precisos, allí diversos algoritmos son propuestos como formas de solución, que requieren etapas de percepción y sistemas de control ajustados a los diversos robots. La localización del robot se puede estudiar de varias formas.

Suponiendo que la postura inicial, antes de iniciar el movimiento, es conocida, la postura actual del robot se puede estimar usando información local del movimiento obtenido a través de distintos sensores de forma que se calcule la distancia recorrida desde el punto inicial; a esto se le conoce como odometría, navegación por estima (Dead Reckoning) o estimación de la posición local [1]. La odometría estima la postura del robot en un plano (x, y y la orientación 0 -ver figura 1) con el pose anterior, la velocidad del robot en los ejes (x,y) y la velocidad angular 0. Para un tiempo de muestreo Ts pequeño se tiene que el pose Lk está dado por la ecuación (1).

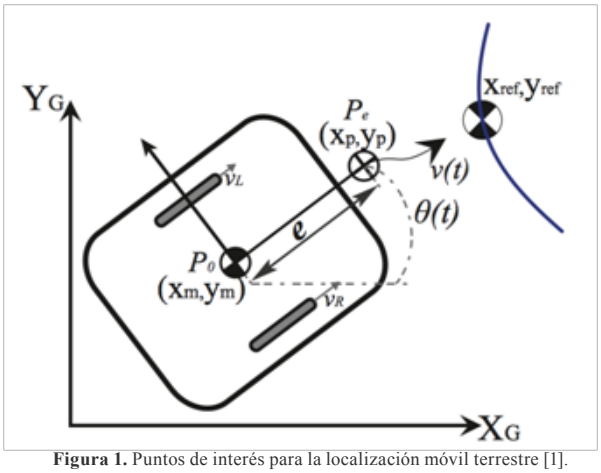

Figura 1. Puntos de interés para la localización móvil terrestre [1].

conjuntos difusos y se utiliza con éxito para manejar la incertidumbre en la toma de decisiones debido a que el uso de evaluaciones lingüísticas es más práctico que numérico.

Igualmente, Dubouis [9] discute algunas nuevas técnicas de análisis de decisión y sugiere el uso de la función de membresía, las variables lingüísticas y los intervalos difusos. Por lo que, utilizando esta última referencia, se extiende TOPSIS difuso para conjuntos de términos lingüísticos difusos imprecisos (HFLTS por sus siglas en inglés) con la opinión de los tomadores de decisiones sobre los criterios de las alternativas [7]. En el presente documento se propone una aplicación del método TOPSIS con la metodología términos conjunto de términos lingüísticos difusos imprecisos HFLTS para la toma de decisiones multi-criterio [MCDM] aplicado a un caso de selección de proveedores.

Conceptos básicos

En general, la Manufactura 4.0 es un nuevo exponente para organizar y controlar la cadena de valor durante la fabricación y el ciclo de vida del producto sustentado por las tecnologías de la información. Este término fue creado por el gobierno alemán como parte de un proyecto llamado: El futuro de la "industria 4.0" para contribuir a la llamada cuarta revolución industrial [10]. Donde el mundo físico-real y el mundo virtual se vinculan en un sistema llamado Cyber Physical-System CPS [11]. Dicha corriente de pensamiento ha aumentado la demanda de mantener satisfecho al cliente, lo que a su vez exige garantizar la realización sin problemas en la cadena de suministro industrial [12] .

De igual forma, los proveedores constituyen un eslabón importante en la cadena de suministro debido a sus servicios de provisión de materias.

$$L_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + T_s \begin{bmatrix} v_{k-1}\cos(\theta_{k-1}) \\ v_{k-1}\sin(\theta_{k-1}) \\ \omega_{k-1} \end{bmatrix} \quad (1)$$

Este procedimiento tiene la ventaja de que el tiempo de respuesta es pequeño (bajo consumo de recursos) y el inconveniente de que el error entre la posición real y la estimada se acumula a lo largo del tiempo (error no acotado). Debido a esto, tras recorrer una cierta distancia la estimación de la posición puede ser muy diferente de la posición real. En caso de que la postura inicial del robot sea desconocida, pero éste disponga de un sensor (barrido láser [2], sonar [3], infrarrojos o cámara omnidireccional [4]) que pueda determinar la posición *relativa* del robot a puntos de referencia *(Landmarks),* no preestablecidos, que observa y extrae del entorno, entonces el robot podrá construir un mapa del entorno y obtener su postura absoluta en este mapa de forma simultánea utilizando la fusión de la información proveniente de los sensores de movimiento y los sensores del entorno. Este procedimiento se conoce como *localización y mapeo simultáneo* (SLAM -1988) y ha sido estudiada desde hace unas décadas por diversos autores [1, 5, 6], entre otros).

La localización y su taxonomía entonces, se presenta de acuerdo al conocimiento de una posición inicial -*position tracking,* denominada también localización local [7] y también de la navegación sobre un entorno desconocido – localización global; más desafiante y requiere sistemas de percepción más completos [2]. Los trabajos de localización y mapeo simultáneo SLAM trabaja más sobre esta última con el uso de filtros estimativos, dentro de este estudio: el filtro de kalman extendido EKFSlam, el filtro de información SEIFSlam, el filtro de partículas FastSlam y el filtro Gráfico GraphSlam.

En la figura 2, se representa la esencia de la población objeto de estudio, en su orden: filtros de estimación en la parte de localización, percepción con sensores y *landmarks,* planeación con *waypoints* y trayectorias directas para la navegación. Físicamente se tienen en cuenta dos robots móviles terrestres y robot móvil aéreo dron que es sobre el cual se quiere tener más enfoque. Otro aspecto importante que se anexa en la implementación del objeto es sobre qué elementos tecnológicos

En la simulación se realizan los modelos cinemáticos y dinámicos de los robots y en base a ellos se programan los filtros para SLAM, en una interfaz gráfica en Matlab se presentan las respuestas y el mapeo de entornos previamente establecidos por el usuario. La parte de la implementación refiere una aplicación para Android que se comunica con protocolo XBee y Bluetooth con dos robots móviles terrestres, ellos a su vez con tarjetas Arduino ejecutan los filtros estimativos de localización y mapeo simultáneo.

Materiales y Métodos

Es indispensable que los robots puedan adquirir información acerca de su ambiente. De acuerdo a las tareas y necesidades existen cientos de sensores disponibles [8] que para localización y navegación se reducen y se concentran en gran medida a sensores exteroceptivos; entre los más utilizados: el GPS, los giroscopios, los sensores infrarrojos, de ultrasonido y láser, recientemente por supuesto los sensores de visión. Considerando que los sensores entregan solo información parcial, ninguno de ellos puede resolver todas las necesidades de percepción. Por ejemplo si encontramos un GPS lo bastante exacto y sin falla alguna, el problema de localización en el robot ¿dónde estoy? podría ser pasado por alto. Desafortunadamente para movimientos incrementales cortos, robots didácticos o de pequeña escala, el GPS no es tan preciso, dando como resultado una misma información dentro del ambiente. Otros problemas encontrados en los GPS son referidos a lugares donde prevalece la reflexión multicamino [9], interferencias y entornos cerrados o lugares donde no llega la señal.

Sensor de Ultrasonido

La mayoría de los sensores de ultrasonido de bajo coste se basan en la emisión de un pulso de ultrasonido cuyo lóbulo, o campo de acción, es de forma cónica. Midiendo el tiempo que transcurre entre la emisión del sonido y la percepción del eco, se puede establecer la distancia a la que se encuentra el obstáculo que ha producido

la reflexión de la onda sonora, mediante la ecuación (2), donde V es la velocidad del sonido en el aire y t es el tiempo transcurrido entre la emisión y recepción del pulso [10].

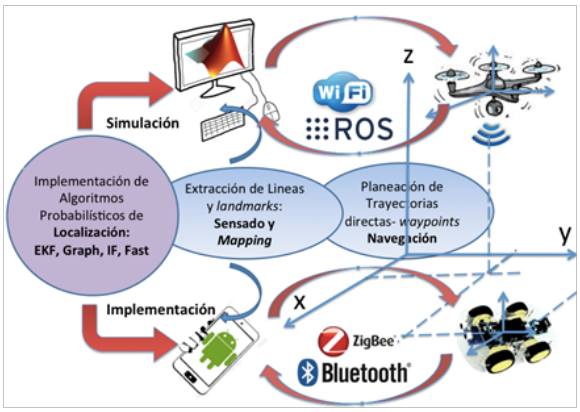

Figura 2. Objetos de Estudio [11].

El eco que se recibe como respuesta a la reflexión del sonido indica la presencia del objeto más cercano que se encuentra dentro del cono acústico y no especifica en ningún momento la localización angular del mismo. Este es el problema más común al usar sensores de ultrasonido. Otro problema sucede cuando la onda reflejada llega varias veces sobre la superficie y se producen "falsos ecos", éste problema se encuentra asociado al denominado *"crosstalk"* que se produce cuando se utiliza un cinturón de ultrasonidos y la onda reflejada llega a otro sensor diferente al que la envió.

Sensor Láser

La palabra láser responde a las siglas *Light Amplification by Stimulated Emission of Radiation* (amplificación de luz mediante la emisión inducida de radiación). Al igual que el ultrasonido, utiliza el principio de tiempo de vuelo para medir distancias (ver ecuación 2). Entre el ultrasonido y el láser existe una diferencia fundamental: la velocidad de propagación. Para el sonido es 0,3 m/ms, mientras que para las señales electromagnéticas es 0,3 m/ns, es decir, 1 millón de veces más rápido [12]. Como principales desventajas se pueden contar su elevado precio y el hecho de mediciones erróneas que algunos materiales como el cristal son invisibles al sensor.

Problema y Técnicas de SLAM

La localización es el problema de estimar la posición del robot y su camino dado un mapa conocido de su ambiente, contrario a ello, el mapping (mapeo)

Para solucionar el problema del SLAM se utiliza un mapa estocástico, que a diferencia de la localización a partir de un mapa a priori, se debe mantener y actualizar a medida que el robot está realizando el mapa automáticamente [13]. En la figura 3, se muestran las variables involucradas dentro del proceso de construcción y localización dentro del mapa. Suponiendo un robot móvil que usa un sensor para el instante de tiempo ***k,*** se define como $\boldsymbol{X_k}$ el vector de estado que describe la posición y orientación del robot, $\mathbf{U_k}$ el vector de control necesario para llevar el robot al estado xk en el instante k, mi definido como el i-ésimo termino del vector de referencia landmark, y zk como la observación tomada por el robot del punto de referencia i-ésimo en el instante ***k***. Para dar solución al problema del SLAM de forma probabilística, se requiere que la distribución de probabilidad sea calculada en todos los instantes k. Haciendo uso del teorema de Bayes el problema puede ser solucionado de manera recursiva, haciendo uso de datos proporcionados por los sensores (con su probabilidad) en cada instante ***k,*** y la función de probabilidad del estado del instante ***k−1***, con ello aparecen los modelos de observación (3) y el modelo de movimiento (4). Estas dos funciones son generalmente invariantes en el tiempo por lo que no dependen de la variable k, ya que las observaciones están condicionadas a el mapa dado y el estado actual del robot, y el próximo estado del robot depende de su estado pasado y de la entrada de control aplicada al mismo.

$$p(z_k / x_k, m) \quad (3)$$
$$p(x_k / x_{k-1}, \mu_k) \quad (4)$$

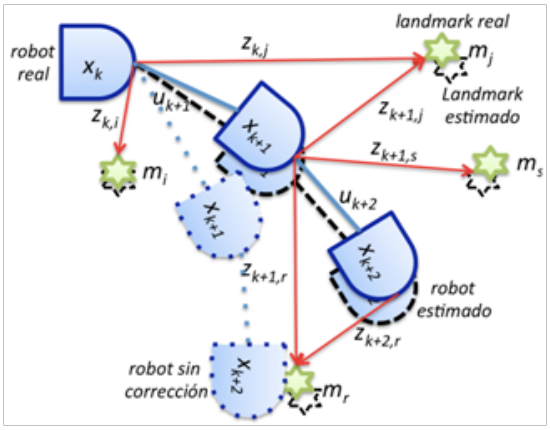

Figura 3. Esquema SLAM (Modificado de [14])

SLAM con Filtro Extendido de Kalman

Aquí los modelos de movimiento y percepción asumen ruido gaussiano como incertidumbre que deben ser linealizados para aplicar el filtro de Kalman extendido EKF como algoritmo de localización –ver tabla I. El ciclo del filtro se basa en dos etapas: la predicción y la actualización [18]. EKF sufre enormemente en la complejidad de actualización –cuyo costo computacional es $O(n_2)$, porque en la mayoría de casos debe revisar muchos *landmarks* y limitaciones en el rastreo del mapa para realizar la asociación de datos, por ello usualmente el número de *landmarks* debe ser pequeño (menor a 1000) [14].

TABLA I.
ALGORITMO EKF

Filtro_Extendido_Kalman (u_{k-1}, Σ_{k-1}, u_k, z_k):
1: $u_k^* = g(u_k, u_{k-1})$
2: $\Sigma_k^* = G_k \Sigma_{k-1} G_k^T + R_k$
3: $K_k = \Sigma_k^* H_k^T (H_k \Sigma_k^* H_k^T + Q_k)^{-1}$
4: $u_k = u_k^* + K_k(z_k - h(u_k^*))$
5: $\Sigma_k = (I - K_k H_k)\ \Sigma_k^*$
return u_k, Σ_k

Fuente: [14].

Filtro Extendido de Información EIF SLAM

El filtro de información estándar está sujeto a las mismas suposiciones subyacentes del filtro de Kalman. Mientras que en el filtro de Kalman, los gaussianos están representados por sus momentos (media, covarianza), los filtros de información representan a los gaussianos con una representación canónica, que comprende una matriz de información Ω (5) y un vector de información ξ (6) [15]. La diferencia en la representación conduce a diferentes ecuaciones de actualización. En particular, lo que es computacionalmente complejo en una representación, pasa a ser simple en la otra (y viceversa).

$$\Omega = \Sigma^{-1} \rightarrow \Sigma = \Omega^{-1} \quad (5)$$
$$\xi = \Sigma^{-1}\mu \rightarrow \mu = \Omega^{-1}\xi \quad (6)$$

FastSLAM con Filtro de Partículas

El filtro de Partículas (PF) es una alternativa no-paramétrica de la implementación del filtro de Bayes [7]. La función de distribución de probabilidad se aproxima utilizando métodos de Montecarlo, para ello mantiene un número de partículas que son N muestras de la distribución a actualizar [16] –cuyo costo computacional es O(N). Cada una de estas partículas mantiene un estado concreto del sistema, es decir una ubicación del robot y un mapa.

La dinámica es similar al filtro de Kalman con los pasos de predicción y actualización. El paso de predicción busca modificar la función de densidad para reflejar un movimiento realizado por el robot –línea 3 de la tabla II. El paso de actualización busca ajustar la función de densidad a la última información de sensado recibida por medio del peso de cada partícula wk –línea 4 de la tabla II.

TABLA II.
ALGORITMO DEL FILTRO DE PARTÍCULAS

Filtro_Particulas (χ_{k-1}, u_k, z_k):
1: $\chi_k^* = \chi_k = 0$
2: for $m=1$ to N do
3: sample $x_k^{[m]} \sim p(x_k / u_k, x_{k-1}^{[m]})$
4: $\omega_k^{[m]} = p(z_k / x_k^{[m]})$
5: $\chi_k^* = \chi_k^* + \langle x_k^{[m]}, \omega_k^{[m]} \rangle$
6: end
return χ_k

Fuente: [14].

Filtro Gráfico GraphSLAM

La solución del filtro modela el problema de SLAM como un grafo [16], donde los nodos representan posiciones del robot xi y landmarks que se optimizan en base a la función de costo (7) obtenida del grafo. GraphSLAM extrae de los datos un conjunto de restricciones generalmente no lineales (8)(9) [16], representadas por un gráfico disperso –ejemplo, figura 4. Obtiene el mapa y la ruta del robot resolviendo estas restricciones en una estimación globalmente consistente, teniendo en cuenta, al igual que EKF y SEIF el ruido generado en la medición y en el movimiento, representados por las matrices de covarianza Q y R respectivamente.

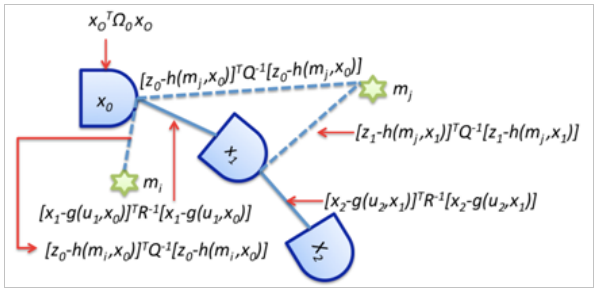

Figura 4. Ilustración de GraphSLAM (Modificado de [14])

$$J = x_0^T \Omega_0 x_0 + \sum_k y_k^T R^{-1} y_k + \sum_k \sum_c w_k^T Q^{-1} w_k \quad (7)$$
$$y_k = x_k - g(u_k, x_{k-1}) \quad (8)$$
$$w_k = z_k - h(m_c, x_{k-1}) \quad (9)$$

Confrontación de Filtros

Una de las primeras características en la implementación, es si la técnica de SLAM se va a ejecutar online u offline; allí la probabilidad posterior (10), (11) para los pasos de medición y actualización difieren considerablemente en la cantidad de información que por ejecución se va a lograr. EKF, SEIF y Fast resuelven piezas de información por ciclo, Graph es más perezoso porque acumula toda la información en un grafo sin resolver, sin embargo puede adquirir mapas de gran magnitud de los que los otros filtros pueden manejar.

$$p(x_k, m \,/\, z_k, u_k) \quad (10)$$
$$p(x_{1:k}, m \,/\, z_{1:k}, u_{1:k}) \quad (11)$$

Una segunda consideración se encuentra constituida con el grado de complejidad que proporciona cada filtro, O(x) de la tabla III. Los modelos paramétricos poseen una complejidad cuadrática supeditados a trabajar con un conjunto reducido de parámetros y contadas funciones de densidad que ellos pueden representar. En contraste, los filtros no paramétricos pueden representar cualquier función de estimación manejando mejor las no linealidades que se presentan en los modelos -de ruido, de movimiento y del mismo robot, así en estos, el coste computacional en la actualización de las distribuciones sea mayor.

TABLA III.

CARACTERISTICAS DE FILTROS PARA SLAM. (AUTORES)

Filtro	*$p(x_k/z_k,u_k)$*	*O(x)*	*Modelo*
EKF:	*online*	*$O(n^2)$*	Paramétrico
SEIF	*online*	*$O(n^2)$*	Paramétrico
FAST	*online*	*O(N)*	No Paramétrico
GRAPH	*offline*	*O(J)*	Paramétrico

Resultados y discusión

Los resultados son tomados de acuerdo al modelo de odometría para los robots móviles (1); se programan los algoritmos para SLAM sobre la plataforma Matlab [17] plantea una trayectoria dentro del proceso de planeación variando los *landmarks* –ejemplo figura 5. Para los resultados se toman en cuenta, los errores de odometría y los tiempos de procesamiento [18].

EKFSLAM

Cuando la odometría cambia se produce incertidumbre debido a los movimientos del robot, entre más movimientos más incertidumbre. El robot extrae del ambiente los landmarks en la medida que se va moviendo.

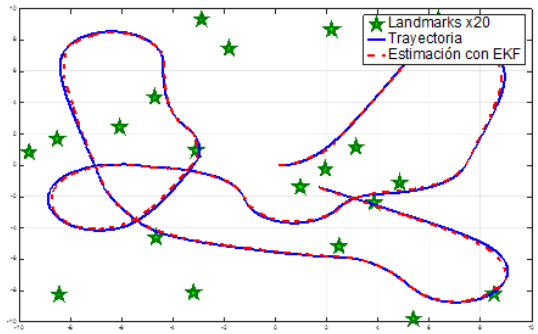

Figura 5. Trayectoria Estimada con EKF y 20 landmarks (Autores)

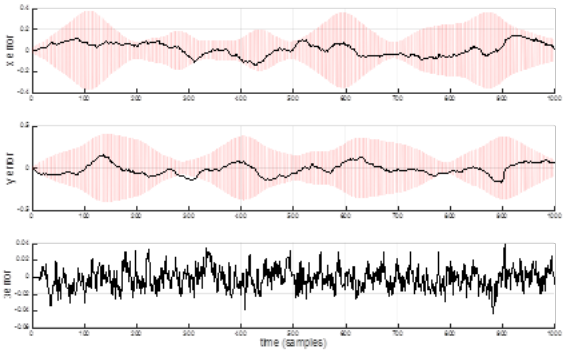

Figura 6. Errores de Estimación de la Odometría con EKFSlam, 20 landmarks y 1000 muestras (Autores)

TABLA IV.

ERROR MEDIO CUADRÁTICO DE ESTIMACIÓN EN ODOMETRÍA PARA CADA FILTRO CON 1000 MUESTRAS Y 20 LANDMARKS.

Filtro	*x_error*	*y_error*	*Θ_error*
EKF	*0.3612*	*0.1401*	0.0063
SEIF	*0.4307*	*0.1561*	0.0071
FAST	*0.2805*	*0.3606*	0.0008
GRAPH	*0.3223*	*0.3467*	0.0008

FastSLAM

Debido a que la generación de partículas es de manera aleatoria, la distribución inicial de ellas es muy dispersa, esto conyeva a errores altos de estimación en sus etapas inciales –ver figura 7. Se encuentra también que a mayor cantidad de landmarks, el error medio cuadrático por odometría disminuye.

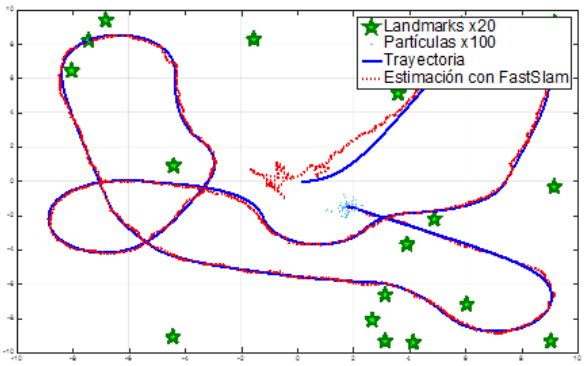

Figura 7. Trayectoria Estimada con FastSlam (Autores)

GraphSLAM

En la primera etapa se establecen las relaciones espaciales de los datos extraídos del sensor, para luego optimizar el grafo y crear la trayectoria global del robot. La representación de movimiento aunque parecido a FAST y más rápida que EKF –ver figura 10, compite con la presencia de altos errores en las estimaciones por odometría representados en la figura 8, anexando picos en la estimación de la orientación.

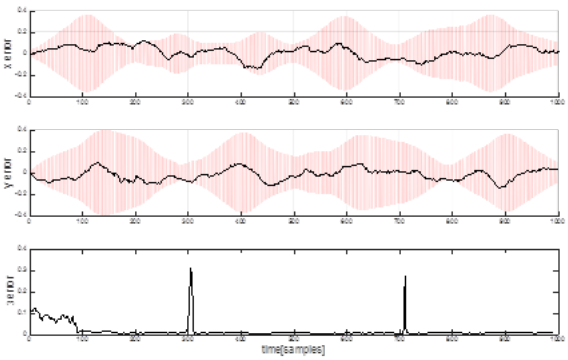

Figura 8. Errores de Estimación de la Odometría con GraphSlam, 20 landmarks y 1000 muestras (Autores)

Generación de Mapa

Las pruebas realizadas de generacion de mapa sobre trayectorias bidimensionales en las instalaciones de la Universidad Autónoma de Colombia B6 -tercer piso, figura 9, aportan algunas tendencias importantes en la implementación con cada filtro. Los mapas generados son muy similares al representado en la figura 9 por EKFSlam donde las diferencias se reflejan en tiempos de procesamiento, errores de odometría y número de landmarks necesarios para generar mapas acordes al camino trazado por los waypoints; estos a su vez como puntos por donde el robot debe pasar no deben verse bloqueados por obstáculos y deben ser fácilmente alcanzables, de igual forma los sensores cuyas distancias de sensado son limitadas deben sincronisarsen para la captura de información sobre el ambiente estático referido.

No obstante EKFSlam posee el menor error medio cuadrático en la generación del mapa, es el que más recurso adquiere de la CPU –ver figura 10; FastSlam administra mejor el tiempo de la CPU, pero SEIF es más rápido dentro de cada iteración.

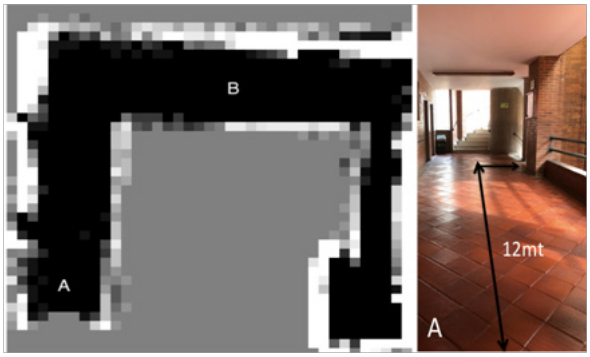

Figura 9. Lugar de trayectoria y mapa creado con EKFSlam -20 *landmarks.*

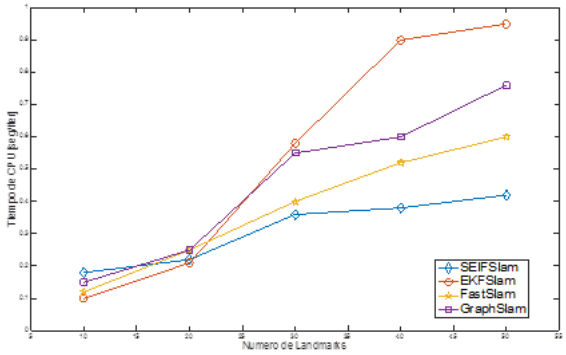

Figura 10. Comparación de filtros SLAM, del uso promedio de tiempo de la CPU.

Conclusiones

En este artículo, como resultado parcial del proyecto de investigación denominado *"Localización Probabilística en Drones"* aprobado por el SUI en la convocatoria No. 26, caracteriza cuatro filtros utilizados para la localización y mapeo simultáneo de robots, teniendo en cuenta resultados de simulación y de implementación. En SLAM cada filtro es soporte de anexo a la caracterización del mapa en base a la información de sensores, l*andmarks y waypoints,* esto es: EKFSlam se encuentra soportado por el filtro de Kalman extendido, FastSlam por el filtro de partículas, GraphSlam por un mapa de grafos y SEIFSlam por el filtro de información.

Observamos que para la implementación de cada filtro es necesario tener la mayoria de elementos en sincronía de comunicación; la mayoría de errores de odometría son generados por estos problemas. No obstante la diferencia de errores que cada filtro provee no es muy grande, EKF proporciona un buen desempeño con la menor tasa de error en odometría y el filtro Gráfico que funciona offline proporcionó la mayor tasa de errores por iteración, esto se debe a que el filtro acumula toda la informacion, la que se envía sin posibilidad de corrección en la localización. El filtro de información es el más rápido en la generación de mapa por tiempo de CPU, utiliza un almacenamiento lineal y no cuadrático como lo hace EKF, los otros dos compiten casi muy similar en cuanto a recurso de CPU se refiere.

Agradecimientos

El autor agradece a todos los integrantes del semillero de investigación S&C por su paciencia, recursos y disciplina para los aportes a la investigación realizada.

Referencias

[1] R. Siegwart, I. Nourbakhsh, D. Scaramuzza "Introduction to Autonomous Mobile Robots". France: MIT, Second Edition, 2011.

[2] L. Armesto, "Técnicas de control y fusión sensorial multifrecuenciales y su aplicación a la robótica móvil". Tesis doctoral, Universidad Politécnica de Valencia, España, 2005.

[3] S. Kim, y B.K. Kim, "Dynamic Ultrasonic Hybrid Localization System for Indoor Mobile Robots". *IEEE Transactions on Industrial Electronics,* vol. 60, pp 4562-4573, 2013.

[4] G. N. DeSouza y A. C. Kak. "Vision for Mobile Robot Navigation: A Survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 2, 2002.

[5] J. M. Armingol-Moreno, Localización Geométrica de Robots Móviles Autónomos. Tesis Doctoral Universidad Carlos III de Madrid, España, 1997.

[6] H. Durrant-Whyte, y T.Bailey. "Simultaneous localization and mapping".: Robotics *Automation Magazine,* vol. 13 no.2, pp. 1-9, 2006.

[7] S. Thrun, W. Burgard, D. Fox, "*Probabilistics Robotics"*. Massachusetts: The MIT Press, 2005

[8] H. R. Everettt, *"Sensors for Mobile Robots, theory and application"*. Massachusetts: A.K. Peters Ltd., 1995.

[9] P. Corke, "*Robotics, Vision and Contro"*. Unite Stated: Springer Publishing, 1 Ed, 2011.

[10] D. Pérez, Sensores de distancia por ultrasonido, 2017. [En linea]. Disponible en: http://www.alcabot.com/alcabot/seminario2006/Trabajos/DiegoPerezDeDiego.pdf [Accedido: 13-marzo-2019].

[11] F. E. Pineda F. "Localización Probabilística en Drones, para aprendizajes cooperativos". Informe de Avance. SUI. Universidad Autónoma de Colombia. Bogotá, Colombia, 2017.

[12] C. Fernández. "Técnicas de Navegación de Robots basadas en medición por láser". Tesis de Pregrado. Universidad de Salamanca, España, 2007.

[13] J. Berrío J. "Mapeo y Localización Simultánea de un Robot Móvil en Ambientes Estructurados Basado En Integración Sensorial". Tesis de Maestría. Universidad del Valle, Cali, Colombia, 2012.

[14] *S. Thrun, "Robotic mapping: A survey. In Exploring Artificial Intelligence in the New Millenium"*. Massachusetts: Morgan Kaufmann Publishers2003.,

[15] C. Stachniss. (2017). "*Robot Mapping - WS 2013/14"*. Germany: UniFreiburg AIS, 2017.

[16] F. Andrade, y M. Llofriu. *"Estudio del estado del arte del SLAM e implementación de una plataforma flexible"*. Tesis de pregrado. Universidad de la República. Montevideo Uruguay, 2017.

[17] Corke Peter (2017). "*Robotics Toolbox for MATLAB"*. European: Press Realease 10.

[18] F. E. Pineda F. "*Localización Probabilística en Drones, para aprendizajes cooperativos"*. Segundo Informe de Avance. SUI. Universidad Autónoma de Colombia. Bogotá, Colombia, 2017.