

## Diseño e Implementación de un Sistema de Localización y Mapeo Simultáneos (SLAM) para la Plataforma Robótica ROBOTINO®

Narváez V. \*, Yandún F. \*, Pozo D. \*, Morales L. \*, Rosero J. \*, Rosales A. \*, Auat F. \*\*

\*Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica, Quito, Ecuador  
e-mail: victor\_narvaez\_t@hotmail.com; fynsk8@yahoo.es; andres.rosales@epn.edu.ec;  
david.pozo@epn.edu.ec; luis.moralesec@epn.edu.ec; jorge.rosero@epn.edu.ec;

\*\* Universidad Técnica Federico Santa María, Departamento de Electrónica, Valparaíso, Chile  
e-mail: fernando.auat@usm.cl

---

**Resumen:** En este documento se presenta el desarrollo e implementación en MATLAB de un algoritmo de localización y mapeo simultáneo (SLAM) para un robot móvil comercial, operando en entornos estructurados estáticos, cuya localización está dada por la odometría del robot y el modelo del mismo. Para la realización del mapeo se utiliza el sensor de rango laser Hokuyo URG-04LX, una vez obtenido el mapa se implementa un algoritmo capaz de detectar esquinas dentro del entorno, con el fin de que puedan ser utilizadas en futuras aplicaciones de SLAM donde sea necesario la inclusión de "landmarks" o características de entorno. Para atenuar los errores de posicionamiento inherentes a la odometría propia del robot se implementa un Filtro Extendido de Kalman entre la odometría y el modelo omnidireccional del robot móvil, en base a lo cual se obtienen mejores resultados para la localización. La detección de esquinas se realiza usando la Transformada de Hough, algoritmo mediante el cual se identifica líneas presentes en el mapa y posteriormente se halla esquinas como un cruce entre dos líneas. Se presentan los resultados como un análisis de los mapas obtenidos contrastados con los entornos reales, además del análisis de errores sobre la localización del robot.

**Palabras Clave:** ROBOTINO, SLAM, Odometría, Transformada de Hough.

**Abstract:** This paper presents the development and implementation of a Simultaneous Localization And Mapping (SLAM) algorithm for a commercial mobile robot, operating in structured and static environments. The location is given by the odometry of the robot and its model. To perform the mapping a sensor laser range Hokuyo URG - 04LX is used. Later, an algorithm capable of detecting corners in the environment is implemented, they can be used in future applications of SLAM where being necessary the inclusion of landmarks or characteristics of environment. To mitigate the position errors inherent to odometry of the robot is implemented an Extended Kalman Filter between the odometry and omnidirectional mobile robot model. This technique improves the results of localization. Corner's detection is performed using the Hough transform algorithm, which identifies lines on the map and subsequently corners like a cross between two lines. We present the results as an analysis of the obtained maps that include contrasts with the real environments. In addition, we analyze the location error of the robot.

**Keywords:** ROBOTINO, SLAM, Odometry, Hough Transform.

---

### 1. INTRODUCCIÓN

Uno de los principales objetivos de la robótica móvil es dotar a los robots de la autonomía necesaria para ser capaces de navegar a través de ambientes desconocidos brindando información acerca del entorno sin la ayuda del ser humano. Siendo uno de los principales inconvenientes la localización del robot móvil cuando se desplaza por un

entorno desconocido [1-3], el presente trabajo se enfoca en solucionar el problema de localización y mapeo mediante una reconstrucción del entorno en dos dimensiones utilizando la información entregada por un sensor laser [4], el cual entrega lecturas de distancias, que procesadas mediante un algoritmo se convierten en un mapa del entorno. En el presente trabajo se utiliza el sensor laser Hokuyo URG-04LX [5].

Adicionalmente, la plataforma robótica Robotino® [6] cuenta con encoders que entregan información de la posición del móvil con respecto a un sistema de coordenadas fijo. La estimación del desplazamiento de la plataforma móvil está

sujeta a un margen de error considerable debido principalmente a la resolución de los encoders y al deslizamiento de las llantas sobre la superficie. Por esta razón, es necesario el desarrollo de un algoritmo que permita obtener una posición fiable del móvil a lo largo del tiempo, dicho algoritmo es el Filtro Extendido de Kalman, el cual será estudiado en las siguientes secciones.

Para la implementación del Filtro Extendido de Kalman [7],[8], se ha considerado estimar la posición del robot móvil en base a su modelo cinemático omnidireccional y como observador se tendrán los datos por odometría.

La detección de características (esquinas o *landmarks*) del entorno, es realizada tomando al mapa como una imagen binarizada y usando los diferentes métodos de detección de bordes que posee MATLAB y posteriormente aplicando la Transformada de Hough [9].

## 2. DESCRIPCIÓN DEL ROBOT MÓVIL

Robotino® ha sido diseñado por *FestoDidactic*, siendo una herramienta didáctica que permite manejar un robot a nivel académico con una diversa gama de sensores que facilitan la realización de diferentes tareas de control.



Figura 1. Robotino®, tomado de [6]

El software provisto por el fabricante para controlar al robot es ROBOTINOView, pero existe la posibilidad de controlarlo mediante los programas MATLAB o LABView con el uso de las respectivas librerías desarrolladas. Nuestro trabajo ha sido desarrollado utilizando MATLAB.

### 2.1 Accionamiento

Robotino® es una plataforma móvil con accionamiento omnidireccional, puede desplazarse en cualquier dirección sin necesidad de girar mediante el cambio de orientación. Cuenta con tres unidades de accionamiento omnidireccional permitiéndole movimientos en todas las direcciones: adelante, atrás, y lateralmente, considerando que además puede girar sobre su propio eje (manteniendo o no su orientación), ello significa que dispone de tres grados de libertad: dos de traslación y uno de rotación.

### 2.2 Modelo Cinemático

El esquema para el modelo omnidireccional del robot móvil se presenta en la Fig. 2.

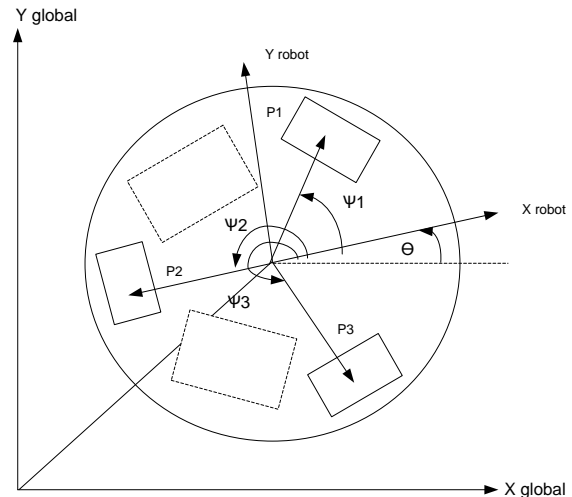


Figura 2. Representación gráfica de un robot con modelo omnidireccional

La expresión (1) representa la relación matemática entre las variables descritas en la Fig. 2 y el movimiento del robot.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = r * \begin{bmatrix} -\sin(\theta + \psi_1) & \cos(\theta + \psi_1) & R \\ -\sin(\theta + \psi_2) & \cos(\theta + \psi_2) & R \\ -\sin(\theta + \psi_3) & \cos(\theta + \psi_3) & R \end{bmatrix}^{-1} * \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (1)$$

En la ecuación (1),  $\omega_1$ ,  $\omega_2$ ,  $\omega_3$  son las velocidades angulares de cada uno de los motores del robot,  $r$  es el radio de las ruedas,  $R$  es la distancia entre una rueda y el centro del robot,  $\theta$  es el ángulo de orientación del robot respecto a los ejes de coordenadas globales,  $\psi_1$ ,  $\psi_2$ ,  $\psi_3$  corresponden a los ángulos en los cuales se disponen las ruedas del robot, finalmente  $\dot{x}$ ,  $\dot{y}$  y  $\dot{\theta}$  son las velocidades lineales y angular del robot. Todas las variables están referidas al sistema de coordenadas global. Además, la ecuación (1) describe el movimiento en tiempo continuo, aunque es necesario discretizar el modelo obtenido para realizar la implementación en base a un algoritmo de tipo recursivo, tal como se muestra en la ecuación (2).

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \Delta T * r * F(\theta_{k-1}) * \begin{bmatrix} \omega_{1k} \\ \omega_{2k} \\ \omega_{3k} \end{bmatrix} \quad (2)$$

Donde el subíndice  $k$  indica el tiempo,  $\Delta T$  es el tiempo de muestreo y  $F(\theta)$  está dado por la siguiente expresión:

$$F(\theta) = \begin{bmatrix} -\sin(\theta + \psi_1) & \cos(\theta + \psi_1) & R \\ -\sin(\theta + \psi_2) & \cos(\theta + \psi_2) & R \\ -\sin(\theta + \psi_3) & \cos(\theta + \psi_3) & R \end{bmatrix}^{-1} \quad (3)$$

Más detalles de la obtención del modelo pueden ser encontrados en [10].

### 3. ALGORITMO DE CONTROL

El algoritmo de control está constituido por tres secciones: el algoritmo del Filtro Extendido de Kalman (EKF) para filtrar los datos de odometría provenientes del robot, el algoritmo que permite la construcción del mapa en base a los datos del sensor laser y un algoritmo basado en la Transformada de Hough para la detección de características (esquinas).

#### 3.1 Implementación del Filtro de Kalman

El filtro toma como base para la etapa de predicción el modelo de un robot omnidireccional, el cual se describió previamente en (2), y como observador para la etapa de actualización los datos de la odometría ( $x, y, \theta$ ) brindados por el software del robot.

##### 3.1.1 Etapa de Predicción

En esta etapa se usa el modelo cinemático del robot (2) para obtener una estimación previa de la posición ( $x, y, \theta$ ) del robot.

El cálculo de la matriz de covarianza estimada, considera el valor de la matriz de covarianzas en un instante anterior  $P_{k-1}$ , además se tiene  $W_k$  que corresponde al Jacobiano del modelo con respecto al ruido, con lo cual se obtiene la siguiente expresión [8]:

$$P_k^- = A_{x,k} P_{k-1} A'_{x,k} + W_k Q_{k-1} W'_k \quad (4)$$

Donde  $A_{x,k}$  es el Jacobiano del modelo con respecto al vector  $x$ . Esta ecuación es completamente válida, pero si se desea tomar en cuenta las perturbaciones en las entradas de control, esta expresión puede extenderse y añadir un término que asocie también la dinámica de tales incertidumbres, quedando la predicción de la matriz de covarianza como:

$$P_k^- = A_{x,k} P_{k-1} A'_{x,k} + A_{u,k} U_k A'_{u,k} \quad (5)$$

Donde  $A_{u,k}$  es el Jacobiano del modelo con respecto a las entradas de control y  $U_k$  es la matriz de covarianzas asociada al proceso, cuyo valor inicial es determinante para que el filtro funcione adecuadamente. Este valor inicial fue

encontrado experimentalmente y se encuentra expresado como se muestra en la expresión (6).

$$U_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \quad (6)$$

donde  $\sigma_x^2$ ,  $\sigma_y^2$  y  $\sigma_\theta^2$  representan las varianzas correspondientes al proceso en los ejes  $x, y, \theta$ , respectivamente.

##### 3.1.2 Etapa de Actualización

En primer lugar se calcula la ganancia de Kalman [8]:

$$K_k = P_k^- H'_k (H_k P_k^- H'_k + R_k)^{-1} \quad (7)$$

Donde  $H_k$  es la matriz identidad de 3x3 y se obtiene directamente de la posición adquirida por odometría considerando al observador como un modelo de tipo lineal. Además,  $R_k$  es la matriz de covarianza asociada a la observación (odometría del robot), cuyo valor inicial es determinante para que el filtro funcione adecuadamente, este valor inicial fue encontrado experimentalmente.

$$R_k = \begin{bmatrix} \sigma_{xr}^2 & 0 & 0 \\ 0 & \sigma_{yr}^2 & 0 \\ 0 & 0 & \sigma_{\theta r}^2 \end{bmatrix} \quad (8)$$

donde  $\sigma_{xr}^2$ ,  $\sigma_{yr}^2$  y  $\sigma_{\theta r}^2$  representan las varianzas correspondientes al observador (odometría del robot) en los ejes  $x, y, \theta$ , respectivamente.

Posteriormente se calcula la posición corregida que servirá para realizar el mapeo. La expresión (9) muestra el cálculo de la posición corregida [8].

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - \hat{x}_k^-) \quad (9)$$

Finalmente, se calcula la nueva matriz de covarianza [8].

$$P_k = (I - K_k H_k) P_k^- \quad (10)$$

Al final de todo el proceso de filtrado, los valores de la matriz de covarianzas deberían ser bajos, puesto que el objetivo del filtro es disminuir los errores y con esto la covarianza de los elementos del vector de estados (diagonal principal de P). El vector de localización filtrado va a tener datos almacenados cada 0.5 segundos y con los cuales se va a realizar el mapeo del entorno.

#### 3.2 Mapeo

Una vez que se han obtenido los datos por parte del sensor

láser y se han filtrado los datos de la odometría del robot, se aplica el algoritmo de mapeo para obtener la imagen del entorno en dos dimensiones.

Los datos que entrega el sensor láser están en coordenadas polares, es por ello que se los debe convertir a coordenadas rectangulares para posteriormente construir el mapa. Con esta transformación es preciso también establecer los ejes de coordenadas para el sistema de referencia local del robot, esto se lo realiza con las ecuaciones 11 y 12:

$$x = r * \cos\left(\theta - \frac{\pi}{2}\right) \quad (11)$$

$$y = r * \sin\left(\theta - \frac{\pi}{2}\right) \quad (12)$$

Quedando el sistema de referencia local del robot, como se muestra en la Fig. 3.

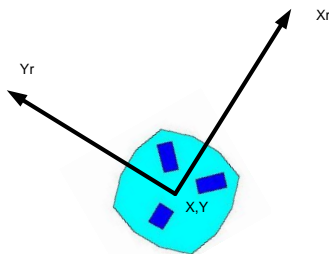


Figura 3. Sistema de referencia local al robot

Es así que los datos del sensor láser quedan relacionados al sistema de referencia local del robot, siendo necesario realizar transformaciones tanto de rotación como de traslación teniendo en cuenta el movimiento del robot, para que los datos obtenidos estén referidos al sistema global de coordenadas, como se muestra en las ecuaciones 13 y 14.

$$x_{mapa} = x_r * \cos(\theta_a) - y_r * \sin(\theta_a) + x \quad (13)$$

$$y_{mapa} = x_r * \sin(\theta_a) + y_r * \cos(\theta_a) + y \quad (14)$$

donde  $x_r$  e  $y_r$  son las medidas del láser en coordenadas cartesianas,  $x$ ,  $y$  y  $\theta_a$  representan la posición actual del robot (luego de ser filtradas).

### 3.3 Detección de Características

Para esta etapa se utiliza la Transformada de Hough con el fin de detectar las líneas del mapa y posteriormente encontrar las esquinas como una intersección de dos líneas.

La Transformada de Hough[11] es una técnica que brinda la posibilidad de detectar figuras dentro de una imagen previamente binarizada, su uso se limitaba a la detección de rectas en una imagen, pero con el transcurrir del tiempo, se extendió hasta llegar a la identificación de cualquier Fig. que se pueda describir con parámetros, tales como circunferencias

y elipses. En el presente artículo se hace énfasis en el estudio del reconocimiento de líneas.

#### 3.3.1 Binarización de la imagen

En primera instancia se crea una cuadrícula con una separación de 6 cm entre cada línea (entre menor sea esta separación se obtiene mayor exactitud, pero el software trabajaría con un mayor número de datos). El objetivo es colocar los puntos del mapa obtenido en la intersección de líneas de la grilla más cercana a ellos, y posteriormente colocar unos (1L) en estas intersecciones y ceros (0L) en las cuales no lo hacen, teniendo así el mapa como una imagen binaria.

#### 3.3.2 Detección de Bordes

Luego de obtener la imagen binarizada, se detectan los bordes de la misma, con este fin se usaron dos de los métodos que proporciona MATLAB para este efecto: Roberts y Prewitt [12]. Es importante destacar que estos métodos se aplicaron por separado a la misma imagen binarizada ya que los resultados obtenidos se complementaban entre sí. Se observó que ciertas líneas existentes en la imagen se detectaban por Roberts, y otras se detectaban con Prewitt, razón fundamental por la que se optó por someter a las imágenes a los dos métodos, recogiendo todos los datos de líneas al final del tratamiento de la imagen.

#### 3.3.3 Detección de Líneas

Una vez detectados los bordes de la imagen, se aplica la Transformada de Hough, encontrando primero los parámetros de las posibles líneas, para posteriormente detectar los picos donde es más probable encontrar una línea en la imagen.

#### 3.3.4 Localización de esquinas

Una vez detectadas las líneas, se promedian las que tengan un rango de distancia entre puntos extremos iniciales y finales, menor a un valor determinado. Posteriormente se encuentran las esquinas como un cruce entre líneas promediadas, teniendo en cuenta que una esquina va a estar en el cruce de dos rectas y el ángulo entre ellas debe ser aproximadamente 90°.

## 4. PRUEBAS Y RESULTADOS

Observamos la implementación en entornos reales<sup>1</sup>, filtrando la odometría y variando diferentes parámetros del filtro o parámetros de las funciones que permiten detectar las características del entorno (esquinas). Para presentar las

<sup>1</sup>Las pruebas fueron realizadas en espacios físicos de libre acceso pertenecientes a la Facultad de Ingeniería en Electrónica y Control de la Escuela Politécnica Nacional (Quito- Ecuador)

pruebas y resultados se han tomado datos en dos ambientes diferentes, ambos estructurados y estáticos, tal como se muestra en la Fig. 4.



Figura 4. Pasillo

#### 4.1 Mapeo

Se comparan los resultados del procesamiento usando únicamente los datos de la odometría, y los datos del procesamiento usando el Filtro de Kalman, se puede apreciar que la odometría entrega datos relativamente aceptables hasta que en cierto punto los errores acumulativos entorpecen el procesamiento y el mapa comienza a desviarse de su correcta construcción. Sin embargo, cuando se aplica el Filtro de Kalman el error disminuye y la elaboración del mapa mejora.

Además, se muestran los resultados al cambiar las constantes del Filtro de Kalman (Figuras 7 y 8) para la inicialización de las matrices de covarianza tanto de la observación (R) como en el proceso (Q). Estas variables contienen la ponderación de los errores presentes en cada caso. Los valores con los cuales se obtuvieron resultados más satisfactorios fueron encontrados de manera heurística, considerando el criterio de que al asignar valores pequeños, las perturbaciones se asumirán como pequeñas; mientras que si se inicializa con valores grandes, la perturbación será considerable.

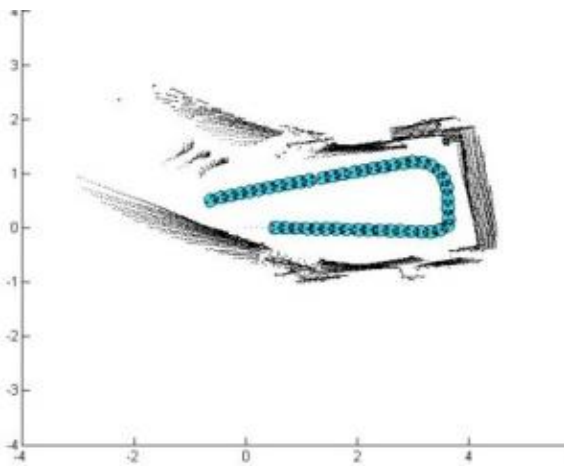


Figura 5. Mapa obtenido con odometría

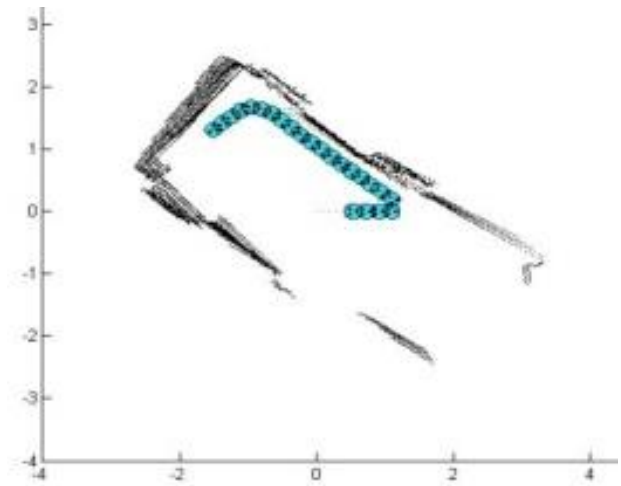


Figura 6. Mapa obtenido aplicando el Filtro Extendido de Kalman

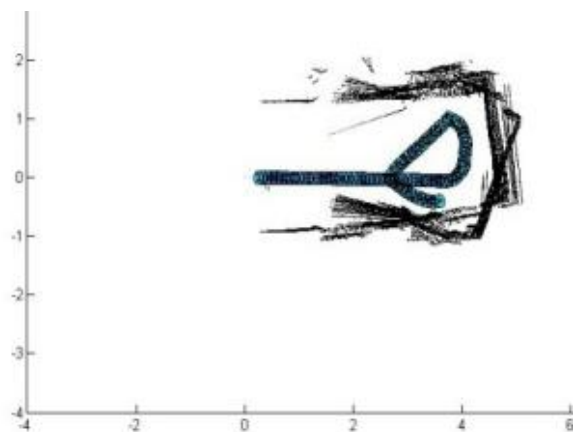


Figura 7. Mapa obtenido con constantes incorrectas en el Filtro Extendido de Kalman

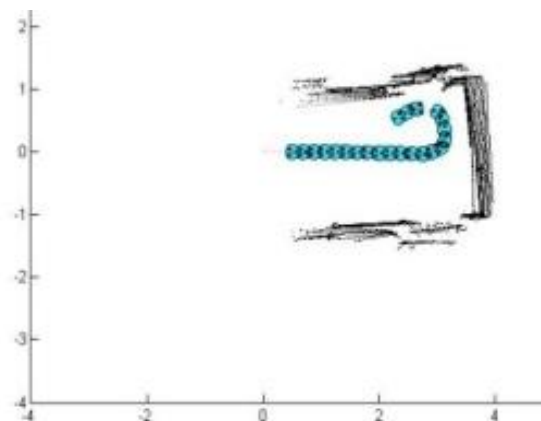


Figura 8. Mapa obtenido con constantes correctas para el Filtro Extendido de Kalman

Además de la experimentación en entornos estructurados y estáticos, se realizaron pruebas cuando algunas personas se encontraban circulando en el ambiente de prueba, esto se

puede notar en las marcas existentes dentro del mapa mostrado en la Fig. 9. La existencia de objetos móviles en el área de mapeo es algo crucial para el algoritmo, puesto que las personas pueden ser confundidas con características, lo que llevaría a un error en la detección de las mismas. Estos errores están descartados en esta aplicación puesto que el presente proyecto está orientado a entornos estáticos y estructurados.

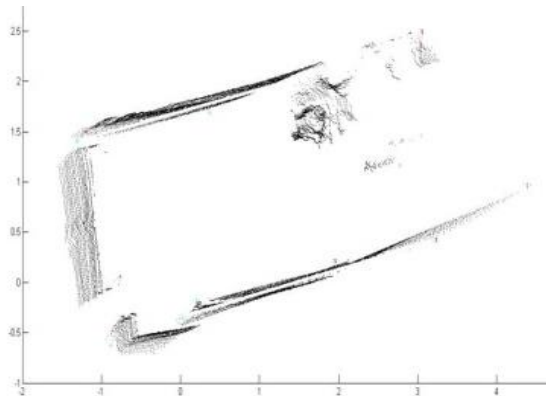


Figura 9. Mapa obtenido con personas caminando en el entorno de prueba

A continuación, se muestra uno de los mapas finales, construido en el entorno de trabajo (Fig. 4) luego de afinar las constantes del EKF (*Extended Kalman Filter*) y realizar la detección de esquinas, las cuales se muestran como círculos en el gráfico.

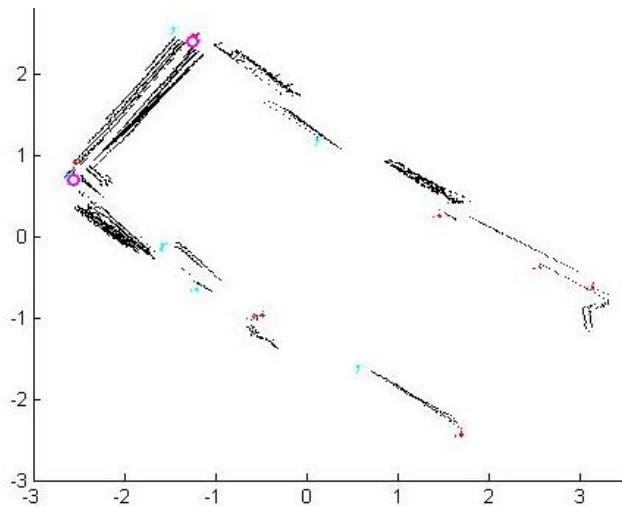


Figura 10. Mapa final

#### 4.2 Localización

Se analiza la posición final del robot y la precisión en la detección de esquinas luego de un recorrido, también se compara la posición del robot en cada iteración del algoritmo durante un mismo recorrido, obteniendo los resultados que se

muestran en la Tabla 1.

Tabla 1. Valores obtenidos en cuatro pruebas en entornos reales

Entornos	Variables	Odometría	Modelo	Kalman	Real
1	X(m)	2,887	2,45	2,648	2,66
	Y(m)	0,6214	0,449	0,611	0,61
	$\theta$ [°]	42,68	42,69	42,7	45
2	X(m)	2,978	2,572	2,748	2,78
	Y(m)	0,32	0,137	0,3	0,38
	$\theta$ [°]	45,97	46,32	45,980	45
3	X(m)	3,155	2,615	2,893	2,92
	Y(m)	-0,121	-0,089	-0,107	-0,2
	$\theta$ [°]	-3,85	-3,71	-3,83	-10
4	X(m)	3,162	2,616	2,9	2,88
	Y(m)	-0,095	-0,066	-0,081	-0,15
	$\theta$ [°]	-3,08	-3,29	-3,06	-5

En la Tabla 2 se muestran los errores obtenidos tanto con la odometría (A), con el modelo (B) así como con el Filtro de Kalman (C) y con los valores reales medidos directamente. Se puede apreciar que luego del filtrado, los datos tienden a aproximarse al valor real, a pesar que los datos de odometría difieren considerablemente de los valores reales.

Tabla 2. Cálculo de errores en la pose del robot

Entornos	Variables	% Error A	% Error B	% Error C
1	X(m)	8,53	7,89	0,45
	Y(m)	1,87	26,39	0,16
	$\theta$ [°]	5,16	5,13	5,11
2	X(m)	7,12	7,48	1,15
	Y(m)	15,79	63,95	21,05
	$\theta$ [°]	2,16	2,93	2,18
3	X(m)	8,05	10,45	0,92
	Y(m)	39,50	55,50	46,56
	$\theta$ [°]	61,50	62,90	61,70
4	X(m)	9,79	9,17	0,69
	Y(m)	36,67	56,00	46,00
	$\theta$ [°]	38,40	34,20	38,80

La Fig. 11. muestra la posición del robot en cada iteración del algoritmo (5 segundos), detallando tanto los valores de desplazamiento obtenidos directamente por la odometría, como por el Filtro de Kalman. Se grafican también los valores reales medidos in situ.

Se puede observar que si el número de iteraciones continúa creciendo, la línea de tendencia de la odometría difiere demasiado de la línea de tendencia de los valores reales, mientras que con el Filtro de Kalman se tiene un mejor resultado. Basados en la gráfica se puede afirmar que luego de procesar los valores dados por el Filtro de Kalman, el algoritmo implementado corrige la odometría y mejora la localización, pero debido a condiciones propias del robot estos resultados no son los ideales. Este comportamiento es justificado debido a que existen variables que no han

sidoconsideradas y que influyen en el sistema como por ejemplo: el deslizamiento de las ruedas, el tipo de piso, nivel de baterías del robot, deterioro de sensores del robot, etc.

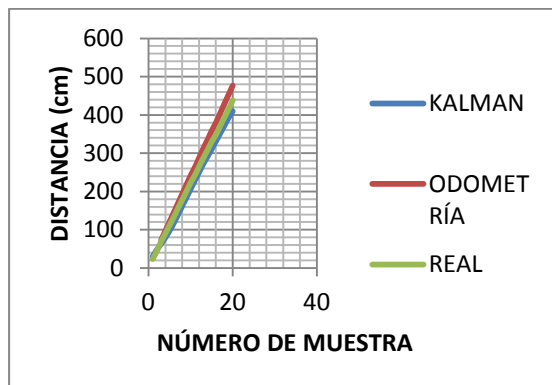


Figura 11. Comparación entre odometría, datos filtrados y datos reales; tomados cada 5 segundos

## 5. CONCLUSIONES

La calibración de las matrices de covarianza en el algoritmo del Filtro de Kalman es un factor que genera una sensibilidad muy alta en el comportamiento del Filtro, razón por la cual, al realizar un mínimo cambio en alguna de ellas los resultados se ven afectados drásticamente.

Si bien el sensor utilizado brinda buenos resultados, este presenta la limitación de verse afectado por la cantidad de luz del ambiente y los colores de los objetos circundantes. Para eliminar o reducir su grado de vulnerabilidad se pueden usar otros sensores como sonares, cámaras, etc.

La posibilidad de establecer una comunicación inalámbrica a través de la red *Wireless* propia de Robotino® es una gran ventaja ya que elimina el hecho de tener un medio físico tal como un cable. Sin embargo, esto incurre en una desventaja por el hecho de que esta comunicación se ve afectada por la contaminación WiFi, que significa que en presencia de otras redes inalámbricas el desempeño del robot no es óptimo, debido a que se puede ver afectada la emisión y recepción de datos.

Si se desea obtener datos de odometría más precisos es posible añadir sensores exteroceptivos, como el sistema de navegación NorthStar [13]. Sin embargo, se debe considerar que si bien los datos de la posición y orientación del robot pueden mejorar, los ambientes de trabajo se verían limitados debido a las características de funcionamiento del sensor, especialmente relacionado con el rango del alcance que es de 2.5m de altura y hasta 4m de distancia.

## 6. REFERENCIAS

- [1] H. Durrant-Whyte, T. Bailey. "Simultaneous Localization and Mapping (SLAM) Part I", IEEE Robotics&Automation Magazine, September 2006.
- [2] H. Durrant-Whyte, T. Bailey. "Simultaneous Localization and Mapping (SLAM) Part II", IEEE Robotics&Automation Magazine, September 2006.
- [3] F. Auat. "Localización y Reconstrucción Simultánea de Entornos por un Robot Móvil basada en la navegación Orientada a las Zonas de Máxima Incertidumbre", Instituto de Automática, Universidad Nacional de San Juan, San Juan Argentina. Diciembre, 2008.
- [4] F. Caramés, "Técnicas de navegación de Robots basadas en Sistemas de medición por Láser", Departamento de Informática y Automática, Universidad de Salamanca, Salamanca, España. Septiembre, 2007.
- [5] Scanning Laser Range Finder URG-04LX-UG01 (Simple-URG) Specification, (2009, Agu). Available: [http://www.hokuyo-aut.jp/02/sensor/07scanner/download/products/urg-04lx-ug01/data/URG-04LX\\_UG01\\_spec\\_en.pdf](http://www.hokuyo-aut.jp/02/sensor/07scanner/download/products/urg-04lx-ug01/data/URG-04LX_UG01_spec_en.pdf).
- [6] FESTO, © Festo Didactic GmbH & Co. KG • 544305 Manual, Denkendorf, 2007.
- [7] Begoña F. Aguado, Janelcy A. Castaño, Eduardo Z. Casanova, Jaime G. García-Bermejo., "Diseño y Simulación de un Filtro Kalman para un Robot Móvil", XXV Jornadas de Automática, Ciudad Real., Septiembre, 2004.
- [8] G. Welch, G. Bishop., "An Introduction to the Kalman Filter", Department of computer Science, University of North Carolina, Chapel Hill, EEUU., July, 2006.
- [9] Universidad Nacional de Quilmes, "Procesamiento Global empleando la Transformada de Hough", Ingeniería en Automatización y Control Industrial, Septiembre, 2005.
- [10] AalborgUniversity, "Modelo Cinemático Omnidireccional", Denmark, 2012. Disponible: <http://www.control.aau.dk/~tb/wiki/index.php/Kinematics>
- [11] Wikipedia, "Transformada de Hough, Teoría e Introducción", 2013. Disponible: [http://es.wikipedia.org/wiki/Transformada\\_de\\_Hough](http://es.wikipedia.org/wiki/Transformada_de_Hough); teoría e introducción de la transformada de Hough.
- [12] Universidad de Jaén, "Detección de bordes en una imagen", Práctica 3, Departamento de Ingeniería, Telecomunicación y Automática.
- [13] FESTO, "NorthstarSensor 2". Manual, Denkendorf, 2009. Disponible: [www.festo-didactic.com](http://www.festo-didactic.com)