

MAPEO Y LOCALIZACIÓN SIMULTÁNEA DE UN ROBOT PARA LEVANTAMIENTO DE MAPAS EN INTERIORES

Julie Stephany Berrío Perez

Universidad Autonoma del Caribe, Barranquilla, Atlantico, Colombia, julie.berrio@uac.edu.co

Eduardo Francisco Caicedo Bravo

Universidad del Valle, Cali, Valle del Cauca, Colombia, eduardo.caicedo@correounivalle.edu.co

Lina Maria Paz Perez

Universidad de Zaragoza, Zaragoza, Zaragoza, España, linapaz@unizar.es

ABSTRACT

This paper presents the development of an algorithm to build maps of the environment based on features of it, which have been extracted of information from a laser scanner embarked on a mobile robot, as vehicle odometry is often wrong, is performed in parallel the location of it, this process is called SLAM (simultaneous localization and Mapping), which is based on a succession of steps, where the objective for this case is to use the straight lines that describe the environment where the robot moves to restore his position. To solve the SLAM problem a stochastic map is used, which is maintained and updated through an Extended Kalman Filter (EKF) while the robot is constructing the map automatically. The results presented were generated through simulation of explorations carried out by a mobile robot PIONNER 3-DX (equipped with a SICK LMS laser sensor-200) in semi-structured environment.

Keywords: error correction, laser scanner, simultaneous localization and mapping, navigation, mobile robot.

RESUMEN

Este documento presenta el desarrollo de un algoritmo que permite construir mapas del entorno basado en características del medio, estas, han sido extraídas a partir de información proveniente de un escáner embarcado en un robot móvil. Como la odometría del vehículo es frecuentemente errónea, se realiza paralelamente la localización del mismo, a este proceso se le denomina SLAM (*Simultaneous Localization and Mapping*). El SLAM se fundamenta en una sucesión de pasos, donde el objetivo para este caso, es utilizar las líneas rectas que describen el ambiente donde se mueve el robot para restablecer su posición. Para solucionar el problema del SLAM se utiliza un mapa estocástico, que se mantiene y actualiza mediante un Filtro de Kalman Extendido (EKF) a medida que el robot está realizando el mapa automáticamente. Los resultados expuestos fueron generados a través de simulaciones de exploraciones realizadas por un robot móvil PIONNER 3-DX (equipado con un sensor laser SICK LMS-200) en entornos semi-estructurados.

Palabras clave: corrección de error, escáner laser, mapeo y localización simultánea, navegación, robot móvil.

1. INTRODUCTION

Entre las muchas formas geométricas, el segmento de línea es la más sencilla. Es fácil describir la mayoría de entornos semi-estructurados con segmentos de línea. Muchos algoritmos de SLAM han sido propuestos para utilizar como características, las líneas extraídas de datos en escáner 2D, por ejemplo, Vandorpe en (Vandorpe,

1996) presenta un algoritmo para la construcción de mapas dinámicos basados en las características geométricas (líneas y círculos) utilizando un escáner láser. Arras en (Arras, 1997) utiliza un escáner 2D con un método de segmentación basado en la regresión lineal basado en los mapas de ubicación. Jensfelt en (Jensfelt, 1998) presenta una técnica para la adquisición y el seguimiento de la posición de un robot móvil con un escáner láser trazando líneas ortogonales (paredes) en un entorno de oficina. Por último, Pfister en (Pfister, 2003) sugiere un algoritmo para la extracción de la línea de conexión ponderada para la construcción de mapas basados en líneas (Nguyen, 2007).

El presente artículo muestra el desarrollo teórico y práctico de un algoritmo que da solución al problema del SLAM basado en la extracción de características del medio ambiente, las cuales son representadas por líneas rectas. En el primer apartado se muestra cómo se procesan los datos del sensor laser para obtener los parámetros de las líneas que describen dichas lecturas. Seguido por el modelado y desarrollo matemático de las ecuaciones necesarias en el Filtro de Kalman Extendido, que se encargan del ajuste del mapa y localización del vehículo. Los resultados obtenidos a través de la implementación del algoritmo son presentados y analizados, para finalmente concluir el documento con anotaciones de los resultados y trabajo futuro.

2. EXTRACCIÓN DE CARACTERÍSTICAS

Para el presente trabajo las características extraídas del entorno son representadas por líneas rectas que se adquieren de los datos obtenidos a través de un escáner laser, estos datos son procesados inicialmente el método RANSAC (*Random Sample Consensus*) para eliminar datos espurios y obtener solo aquellos que describan una línea recta. La división y parametrización de rectas se realiza con el método SEF (*Successive Edge Following*) y la técnica TLS (*Total Least Square*).

El espacio de trabajo es descrito por características estáticas rectilíneas (tales como partes de las paredes, puertas, estantes) que son detectadas por el sistema sensorial del robot. El robot está equipado con sensores de proximidad que proporcionan una serie de pasos s y medidas, $A^s = [r_R^A, \theta_R^A]$ que son parametrizados por su distancia r_R^A desde el origen hasta el punto en la dirección de detección θ_R^A . La pareja $A^s = [r_R^A, \theta_R^A]$ puede considerarse como las coordenadas polares de la lectura s en un robot centrado en un sistema de referencia. Las coordenadas cartesianas se denotan por:

$$A^s = [x_R^A, y_R^A] = [r_R^A \cos(\theta_R^A), r_R^A \sin(\theta_R^A)] \quad (1)$$

Para extraer líneas de los barridos laser, se utiliza del algoritmo RANSAC *Random Sample And Consensus*, el cual es un método para estimar los parámetros de un modelo determinado a partir de un conjunto de datos contaminados por grandes cantidades de valores atípicos. El ajuste es robusto, ya que reconoce la presencia de valores desalineados en los datos. El procedimiento RANSAC se compone de dos pasos que se repiten de forma iterativa (hipótesis y el marco de la prueba) (Zuliani, 2009):

- **Hipótesis:** En primer lugar establece un mínimo de muestra (MSS del inglés *minimum simple set*) seleccionadas al azar del conjunto de datos de entrada y los parámetros del modelo se calculan utilizando sólo los elementos del MSS.
- **Prueba:** En el segundo paso del RANSAC se evalúa que los elementos del conjunto de datos sean coherentes con el modelo instanciado con los parámetros estimados en el primer paso.

Si el número de puntos del conjunto de consenso es mayor que un cierto umbral, a continuación, se vuelve a reajustar la línea para el conjunto de puntos de consenso (con los mínimos cuadrados totales, para este caso). De lo contrario el algoritmo se repite hasta que se alcanza un número máximo de iteraciones establecido de antemano. Cuando todas las iteraciones se terminan, se obtiene la línea con una regresión de los puntos clasificados como alineados con el TLS (*Total Least Squares*). El TLS minimiza la suma del cuadrado de las distancias de los puntos a la recta (medidas en perpendicular). En el caso del problema del TLS, se toma en cuenta las perturbaciones en la matriz de sensibilidad, para ello el TLS se define como la minimización sobre

el rango y perturbaciones de la matriz de sensibilidad de tal forma que la distancia normal euclidiana entre la recta y los puntos que la componen sea la menor posible.

3. ASOCIACIÓN DE CARACTERÍSTICAS

La asociación de las características relaciona la información observada en la iteración actual, que se obtiene respecto a la posición del robot, con una característica observada en iteraciones anteriores y que se encuentra almacenada en el vector de estado (mapa). La disponibilidad de un modelo estocástico, tanto para el mapa como para las mediciones, nos permite calcular cada correspondencia de las características observadas, mediante la predicción de la ubicación de los ya almacenados puntos de referencia en relación a la posición del robot y el sensor, y determinar la compatibilidad con una prueba de hipótesis sobre la innovación y la covarianza de cada pareja posible.

Las características se asocian con una ventana de validación que define el error máximo admisible entre la observación y la predicción. La distancia de Mahalanobis o la distancia estadística, es una distancia que para cada uno de sus componentes (las variables) toma su variabilidad en cuenta al determinar su distancia al centro correspondiente. Esto se logra ajustando la base de los componentes considerados, es decir, para dos puntos $X_i = (\rho_{11i}, \theta_{12i})$ y $Y_i = (\rho_{12i}, \theta_{22i})$ la distancia de Mahalanobis está dada por la ecuación (2).

$$d_M = \sqrt{(\rho_{11i} - \rho_{12i})^T C^{-1} (\theta_{22i} - \theta_{22i})} \quad (2)$$

donde C es una matriz de covarianza no singular de dimensión $n \times n$. En un principio, hay mn hipótesis de emparejamiento, la validación se realiza mediante una prueba estadística basada en la distancia de Mahalanobis y su aproximación por la distribución χ^2 (distribución de probabilidad continua).

4. FILTRO DE KALMAN EXTENDIDO

Inicialmente en la implementación del Filtro de Kalman Extendido se ha determinado la ubicación del robot como parte del vector de estado $x \in R$ el cual está conformado por sus coordenadas cartesianas y orientación:

$$x = (x_{robot}, y_{robot}, \phi_{robot}) \quad (3)$$

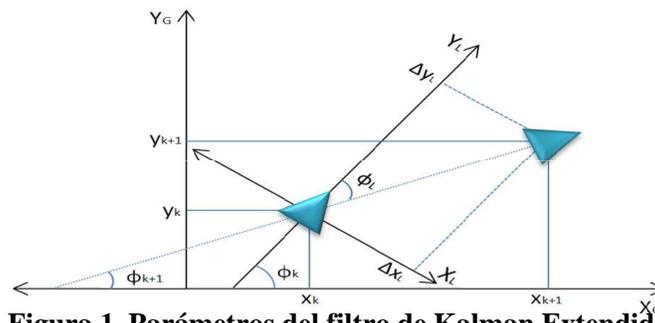


Figura 1. Parámetros del filtro de Kalman Extendido.

y el vector de entradas de control como $u = (\nabla x_r, \nabla y_r, \nabla \phi_r)$ donde ∇x_r , ∇y_r y $\nabla \phi_r$ representan las lecturas de odometría. El modelo dinámico del sistema (robot móvil) se define en forma generalizada como:

$$x_r = f(x_k, u_k) + q_k \quad (4)$$

donde f es un modelo no lineal de la dinámica del robot móvil, cuyas variables corresponden al estado actual x_k y al de control u_k , q_k representa el ruido del proceso de media cero y covarianza Q_k con $k = 1, 2, 3, \dots$. Con estas especificaciones se define la función $f = [f_x, f_y, f_\phi]^T$ como:

$$f_x = x_{k+1} = x_k + \nabla x_l \cos(\phi_k) - \nabla y_l \sin(\phi_k) \quad (5)$$

$$f_y = y_{k+1} = y_k + \nabla x_l \sin(\phi_k) + \nabla y_l \cos(\phi_k) \quad (6)$$

$$f_\phi = \phi_{k+1} = \phi_k + \nabla \phi_l \quad (7)$$

donde: $\nabla \phi_l = \arctan \frac{\nabla x_l}{\nabla y_l}$ (8)

Para el desarrollo de las ecuaciones del filtro de Kalman se han de tener en cuenta las siguientes notaciones:

$x_{k|k}$ representa la estimación del estado x en el instante k con las medidas disponibles estimación).

$x_{k+1|k}$ representa la estimación de x en el instante $k + 1$ dada las primeras k medidas (predicción).

El vector de estado corresponde a un vector columna $x = [x_r^T, x_1^T, x_2^T, x_3^T, \dots, x_n^T]^T$ que contiene la posición estimada actual del robot en coordenadas cartesianas $x_r = [x_r, y_r, \phi_r]^T$, junto con las posiciones de todos los hitos observados $x_i = [\theta_i, \rho_i]^T$. Por tanto, este vector tendrá dimensiones $(3 + 2n) \times 1$, siendo n el número de hitos. La matriz de covarianza $P_{k|k} = E[\eta_k \eta_k^T]$ contiene las relaciones de la posición del robot consigo mismo, con las posiciones de los hitos, las relaciones entre las posiciones de los hitos y las relaciones cruzadas entre hitos.

La primera submatriz P^{rr} corresponde a la covarianza de la posición del robot por tanto será una matriz 3×3 . Las tres primeras filas de la matriz, representadas por P^{ri} , representan a las correlaciones entre el robot y los diferentes hitos, al igual que las tres primeras columnas que son las correlaciones de los hitos con el estado del robot.

$$P = \begin{bmatrix} P^{rr} & P^{ri} & \dots & P^{rn} \\ P^{ir} & P^{ii} & \dots & P^{in} \\ \vdots & \vdots & \ddots & \vdots \\ P^{nr} & P^{ni} & \dots & P^{nn} \end{bmatrix} \quad (9)$$

La primera será una matriz de dimensión $3 \times 2n$ mientras que la segunda será de dimensión $2n \times 3$, donde n es el número de hitos del vector de estados. Las matrices restantes P^{ii} , P^{in} y P^{ni} que constituyen una matriz de dimensión $2n \times 2n$ simétrica, se conforma de la covarianza de los hitos consigo mismos, y la covarianza cruzada entre hitos.

5. PROGRAMACIÓN DEL FILTRO DE KALMAN EXTENDIDO.

El robot y el mapa se expresan en un único vector de estados X con su correspondiente estimación del error de covarianza P en cada paso, un EKF se utiliza para estimar el estado X y la covarianza P dada la medición Z , así, el progreso de creación del mapa se fracciona en tres partes: desplazamiento del vehículo (reajuste de odometría), composición de los hitos y re-observación de los mismos. La figura 2 muestra el diagrama de pasos general que fue desarrollado.

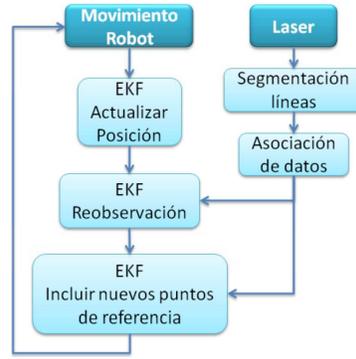


Figura 2. Proceso de SLAM implementado.

Cada que se produzca un movimiento del robot (cambio en la odometría), se realiza el cálculo de la nueva posición del mismo, actualizando las coordenadas x , y y θ . Con las lecturas proporcionadas por el sensor laser, se extraen las características del entorno, las cuales son procesadas junto con las características anteriormente detectadas, aquellas que son asociadas a una característica pasada, corregirán la posición actual del robot. Conociendo la estimación de las coordenadas de la posición anterior y el movimiento producido, se realiza el cálculo de la discrepancia entre la posición observada de los hitos y la estimada, presumiendo que el robot se halle efectivamente en el lugar que enseña la odometría, para realizar el ajuste el estado actual del robot. Las características que se han observado por primera vez se incluyen en el conjunto de hitos reconocidos para futuras iteraciones y las reobservadas se ajustan con los nuevos parámetros.

5.1 PREDICCIÓN

5.1.1 DESPLAZAMIENTO DEL VEHÍCULO

Cuando el robot móvil se mueve, el estado estimado del mismo en el instante $k + 1$, está determinado por la expectativa en el modelo del estado de transición:

$$x_{r_{k+1}|k} = E[f(X_k, u_k)] \approx f(x_{r_{k|k}}, u_k) \quad (10)$$

Como suponemos que el ruido que afecta el proceso tiene una media igual a cero, el valor se hace cero al evaluarlo con el valor estimado x_r para hallar $x_{r_{k|k}}$. La covarianza $P_{k|k}^{rr}$ se propaga por el modelo lineal del estado de la transición del EKF proporcionando $P_{k+1|k}^{rr}$:

$$P_{k+1|k}^{rr} = J_x P_{k|k}^{rr} J_x^T + Q_k \quad (11)$$

donde $Q_k = E[q_k q_k^T]$, J_x es el jacobiano de f con respecto a X . Este procedimiento brinda una posición estimada del vehículo basada únicamente en el modelo del robot y los valores de odometría. El jacobiano del modelo de predicción está estrictamente relacionado con el modelo de predicción. Este precisa como predecir la posición del robot desde de la posición anterior

$$f = \begin{bmatrix} x_R + \nabla x_l \cos(\theta_R) - \nabla y_l \sin(\theta_R) \\ y_R + \nabla x_l \sin(\theta_R) + \nabla y_l \cos(\theta_R) \\ \theta_R + \nabla \theta_l \end{bmatrix} \quad (12)$$

donde x_R y y_R son las coordenadas del robot, θ_R es la rotación, ∇x_l ∇y $\nabla \theta_l$ es el cambio de las coordenadas respectivamente (ver figura 2). Asumiendo que se puede acceder directamente a los incrementos a través de señales del robot y que se va a trabajar con un modelo linealizado, el jacobiano del modelo de predicción quedará como:

$$J_x = \begin{bmatrix} 1 & 0 & -\nabla x_l \sin(\theta_R) - \nabla y_l \cos(\theta_R) \\ 0 & 1 & \nabla x_l \cos(\theta_R) - \nabla y_l \sin(\theta_R) \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Se supone que el proceso tiene un ruido gaussiano proporcional a los controles dados al robot. La matriz Q se calcula como $Q = J_{\nabla x} C J_{\nabla x}^T$ donde $J_{\nabla x}$ es la matriz jacobiana de derivadas parciales de f con respecto a ∇x :

$$J_{\nabla x} = \begin{bmatrix} \cos(\theta_R) & -\sin(\theta_R) & 0 \\ \sin(\theta_R) & \cos(\theta_R) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

La matriz C es una aproximación de la exactitud de la odometría. El valor suele estimarse realizando experimentos y comprobando cual es el valor que mejor se ajusta a los resultados obtenidos.

También es menester actualizar las correlaciones cruzadas robot-hito como:

$$P_{(k+1|k)ri} = J_x P_{(k|k)ri} \quad (15)$$

Sin embargo no es necesario renovar las correlaciones cruzadas entre los hitos, debido que la ubicación del hito en el plano no es afectada por cambios en el estado del robot.

5.1.2 PREDICCIÓN DE OBSERVACIONES

En este punto queremos saber cómo serían las rectas si se observan desde la posición actual del robot, para luego compararlas con las observaciones actuales. En el instante k se tiene en vector de estado de la forma $X = [x_r^T, x_1^T, x_2^T, x_3^T, \dots, x_n^T]^T$, donde x_r es la posición actual del robot y x_i las posiciones de todos los hitos observados en la referencia global. Esta transformación proporciona una estimación para la posición del punto de referencia visto desde la nueva posición del robot, dando como resultado una función de predicción h de la siguiente forma:

$$h = \begin{bmatrix} \rho - x_R \cos(\theta) - y_R \sin(\theta) \\ \theta - \theta_R \end{bmatrix} \quad (16)$$

Dado que la predicción se realiza a través de una función no lineal debido a las transformaciones presentadas, y dado que también se debe obtener la matriz de covarianza, se realiza una linealización alrededor del punto x_k , para propagar de esta manera la matriz P a través de la transformación.

La matriz Jacobiana H_x de la predicción es compuesta por H_r y H_f donde:

$$H_r = \frac{\partial h}{\partial x_R} = \begin{bmatrix} -\cos(\theta) & -\sin(\theta) & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (17)$$

$$H_f = \frac{\partial h}{\partial x_f} = \begin{bmatrix} 1 & x_R \sin(\theta) - y_R \cos(\theta) \\ 0 & 1 \end{bmatrix} \quad (18)$$

Para todas las predicciones, tenemos una matriz jacobiana H de la forma:

$$H = \begin{bmatrix} H_{r1} & H_{f1} & 0 & \dots & 0 \\ H_{r2} & 0 & H_{f2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{2n} & 0 & 0 & \dots & H_{fn} \end{bmatrix} \quad (19)$$

Propagando la matriz de jacobianos en la predicción así:

$$S = H P_k H^T \quad (20)$$

5.1.3 RE-OBSERVACIÓN DE LOS HITOS (ACTUALIZACIÓN)

Cuando un hito presente en el vector de estados es observado, se actualiza por medio del EKF el estado del mapa incluyendo la posición del robot. La ecuación del modelo de observación para el punto de referencia i (en forma de línea) tiene la forma

$$z_{ik+1} = \begin{bmatrix} \rho_{ik+1} \\ \theta_{ik+1} \end{bmatrix} = \quad (21)$$

$$\begin{bmatrix} x_R \cos(\theta_R + \theta_i) + y_R \sin(\theta_R + \theta_i) + \rho_i \\ \theta_R + \theta_i \end{bmatrix} + n_{zi} \quad (22)$$

$$= h_i(x_{k+1|k}) + n_{zi} \quad (23)$$

Se considera que el ruido de todo el proceso n_{zi} es un ruido blanco gaussiano de covarianza R_i . Si los n hitos son observados el modelo de la observación estará dado por las siguientes ecuaciones:

$$z_{k+1} = \begin{bmatrix} z_{ik+1} \\ \vdots \\ z_{nk+1} \end{bmatrix} \quad (24)$$

$$h = \begin{bmatrix} h_1 \\ \vdots \\ h_n \end{bmatrix} \quad (25)$$

$$R_{k+1} = \begin{bmatrix} R_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_n \end{bmatrix} \quad (26)$$

Denotando al jacobiano de h como H , las ecuaciones del proceso de actualización son:

$$x_{k+1|k+1} = x_{k+1|k} + K_{k+1} (z_{k+1} - h(x_{k+1|k})) \quad (30)$$

$$P_{k+1|k+1} = (I - K_{k+1}H_x)P_{k+1|k} \quad (27)$$

donde K_{k+1} es la ganancia de Kalman que se calcula como:

$$K_{k+1} = P_{k+1|k+1}H^T(H_xP_{k+1|k+1}H_x^T + R_{k+1})^{-1} \quad (28)$$

La ganancia de Kalman o factor de mezcla establece la cantidad de influencia del error entre nuestra estimación y la medida, por lo que le proporcionará un mayor o menor peso a la odometría en relación con los hitos observados.

5.1.4 INTEGRACIÓN DEL NUEVO HITO

Cuando un nuevo hito $L_{new} = [\rho, \theta]$ es visto y validado el estado del mismo X_{n+1} se incorpora al vector de estado del sistema

$$X_{n+1} = m(x_{k+1|k+1}, L_{new}) \quad (29)$$

$$X_{n+1} = \begin{bmatrix} x_R \cos(\theta_R + \theta) + y_R \sin(\theta_R + \theta) + \rho \\ \theta_R + \theta \end{bmatrix} \quad (30)$$

$$X_{k+1|k+1} \leftarrow \begin{bmatrix} X_{k+1|k+1} \\ X_{n+1} \end{bmatrix} \quad (31)$$

Seguidamente se completa la matriz de covarianza P para que comprenda al nuevo hito, para su realización se tendrá que calcular siguientes sub-matrices que son la covarianza del hito $n + 1$, la covarianza cruzada hito-robot y la covarianza cruzada hito-hito (Dominguez, 2008)

$$P^{n+1|n+1} = J_{X_r} P_{k+1|k+1}^{rr} J_{X_r}^T + J_z R J_z^T \quad (32)$$

$$P^{rn+1} = P^{n+1rT} = P_{k+1|k+1}^{rr} J_{X_r}^T \quad (33)$$

$$P_{k+1|k+1}^{n+1i} = P_{k+1|k+1}^{in+1}{}^T = J_{X_r} P_{k+1|k+1}^{ri} \quad (34)$$

Las matrices J_{X_r} y J_z son las matrices jacobianas de m con respecto al estado del robot X_r y al punto de referencia L_{new} .

$$c_j = \cos(\theta + \theta_R) \quad (35)$$

$$s_j = \sin(\theta + \theta_R) \quad (36)$$

$$J_{X_r} = \begin{bmatrix} c_j & s_j & -x_R s + y_R c \\ 0 & 0 & 1 \end{bmatrix} \quad (37)$$

J_{X_r} muestra cuánto cambia el rango y la orientación de las líneas (hitos) cuando varían x_R , y_R y θ_R .

El primer elemento de la primera fila es el cambio del rango con respecto a x_R ; el segundo, respecto a y_R y el tercer valor es con respecto a θ_R . La segunda fila proporciona la misma información pero con relación a la orientación:

$$J_z = \begin{bmatrix} 1 & -x_R \sin(\theta_R + \theta) + y_R \cos(\theta_R + \theta) \\ 0 & 1 \end{bmatrix} \quad (38)$$

6. RESULTADOS

Para comparar los resultados con la realidad del medio, es esencial que toda la información se encuentre en el mismo marco de referencia. Sin embargo, en experimentos de laboratorio el marco de referencia del SLAM, la odometría y el mapa verdadero no están por lo general en el mismo marco. Es por esto que se realiza una simulación ideal donde los valores de odometría y laser son exactos, datos que luego son alterados con la adición de ruido, obteniendo de esta forma dos conjuntos de datos (real y aquel con datos alterados con ruido) con el mismo marco de referencia. Cabe anotar que el conjunto de datos con presencia de ruido es procesado por el algoritmo de EKF-SLAM, cuyo resultado puede ser comparado con la realidad del mapa.

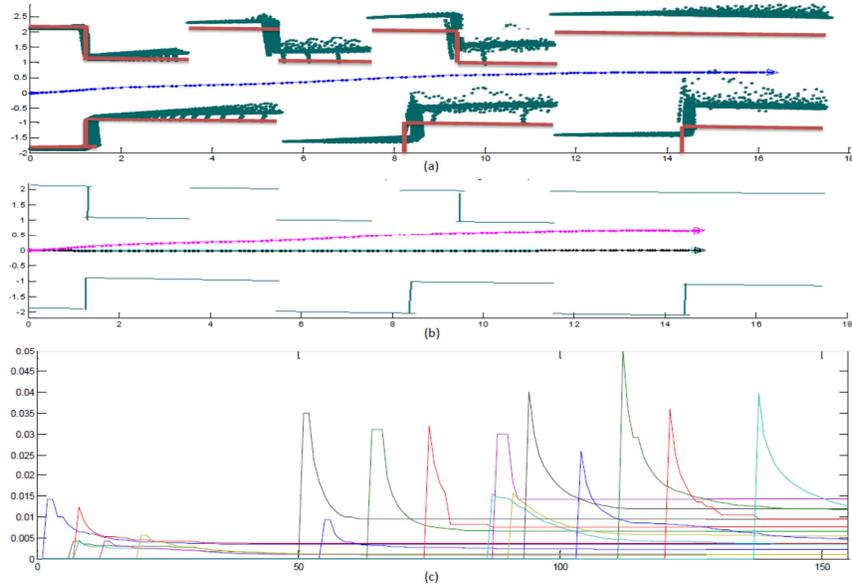


Figura 3. Resultado de la ejecución del EKF-SLAM para el mapa modelado.

La Figura 3 con tres gráficas es el resultado del EKF-SLAM para el mapa diseñado, donde robot ha realizando un recorrido en línea recta, iniciando con una desviación de 2.7° respecto a la horizontal. La gráfica 3(a) muestra el mapa real superpuesto con los datos de odometría y laser afectados por ruido (ejes de coordenadas en metros). La segunda gráfica 3(b) se ha obtenido después de 160 iteraciones del algoritmo, donde se muestra el estado del robot real (azul), estimado (negro) y según odometría (magenta), junto con el mapa construido. Por su parte la tercera gráfica 3(c) muestra la evolución de la incertidumbre de las características con el pasar de las iteraciones (eje x). Con el fin de comparar numéricamente, tanto el mapa creado como la corrección de odometría realizados a través del SLAM-EKF, se definieron algunas medidas de distancia que describen la disposición del mapa, las cuales son confrontadas con aquellas obtenidas tras la ejecución del proceso de SLAM.

La Tabla 1 registra el valor las longitudes especificadas en metros del entorno real registrado por el robot y el resultante de la ejecución del algoritmo de EKF-SLAM, además de una columna del error porcentual existente entre ellas.

Tabla 1. Comparación entre distancias características de un mapa.

| Distancia | SLAM [m] | Mapa Real [m] | Error % |
|-----------|----------|---------------|---------|
| d1 | 1.001 | 1 | 0.12% |
| d2 | 1.961 | 2 | 1.93% |
| d3 | 0.994 | 1 | 0.52% |
| d4 | 2.064 | 2 | 3.22% |
| d5 | 1.933 | 2 | 3.35% |
| d6 | 3.077 | 3 | 2.59% |
| d7 | 3.002 | 3 | 0.07% |
| d8 | 3.054 | 3 | 1.82% |
| d9 | 2.986 | 3 | 0.44% |

Con el fin de ratificar la importancia y potencialidad del estudio del problema de SLAM, se ha realizado cambios en el código original de mapeo, buscando que este genere un archivo externo que contenga el mapa procesado. El mapa final se encuentra compuesto de puntos extremos de líneas rectas que lo componen, que son escritos al final del proceso en un archivo ".txt". El archivo del mapa es cargado y dibujado en el ambiente AutoCad (Bousfield (1999)) por medio de un programa desarrollado en Autolisp (lenguaje de programación que permite desarrollar

programas y funciones para el manejo de entidades del tipo gráfico orientadas al uso específico de AutoCAD). La figura 9 muestra un ejemplo de los dibujos de los mapas en el ambiente de diseño Autocad.

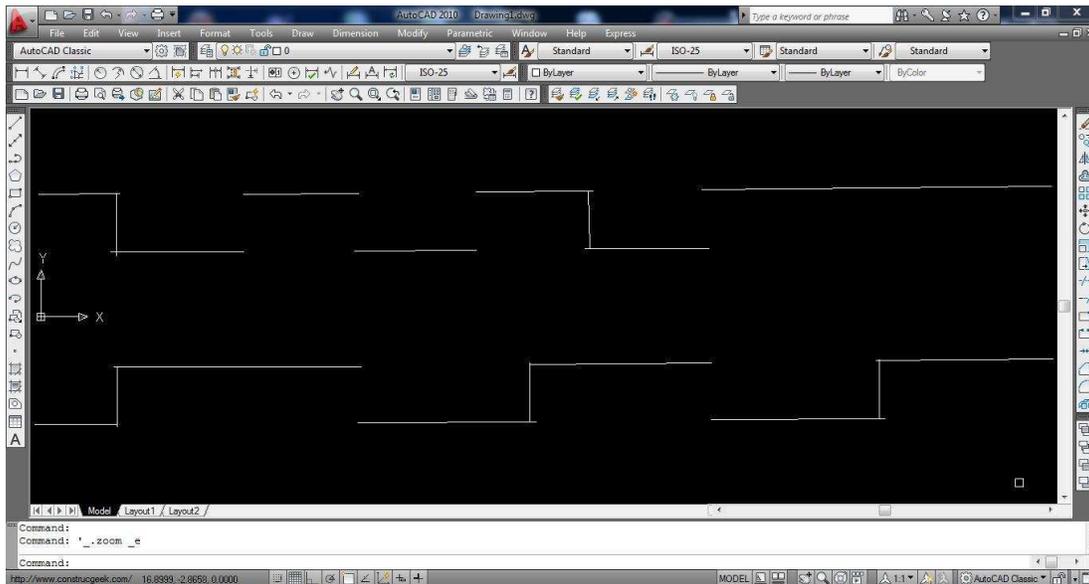


Figura 4. Mapa en autocad.

CONCLUSIONES

La técnica de SLAM implementada basada en líneas rectas como puntos de referencia, resulta adecuada para realizar el mapeo de ambientes estructurados, tanto por la descripción del ambiente como por la compacidad de los datos. Esto en virtud de que una sola característica detectada y actualizada resulta ser muy representativa (en longitud) para la construcción del mapa, haciendo que el crecimiento del tamaño del mismo sea reducido, y por lo tanto el tiempo de computo menor.

Un desarrollo eficiente del SLAM requiere mantener el error de la ubicación del robot examinada mediante la observación de las características con regularidad. El tema de la función de correspondencia basada en la cercanía a menudo conduce a la adecuación de las características cercanas en el espacio, pero no topológicamente, con lo que se realiza un mapa erróneo, debido a esto es necesario realizar otras pruebas para indicar compatibilidad.

Los mapas desarrollados en este trabajo asumen el medio ambiente semi-estático y por lo tanto, el algoritmo está diseñado para manejar entornos medianamente dinámicos. Sin embargo, teniendo en cuenta el trabajo reciente de localización en entornos dinámicos, el sistema de SLAM podrá funcionar en entornos dinámicos, realizando el seguimiento de rasgos estáticos y descartando los no estáticos.

El dibujo obtenido a través del programa desarrollado en AutoLisp brinda una aproximación a un levantamiento de mapas de interiores para determinar la configuración del ambiente y la posición sobre la superficie de elementos o instalaciones, que podría ser utilizado en las fases de los proyectos arquitectónicos y de ingeniería civil.

REFERENCIAS

H. X. J. Vandorpe, H. Van Brussel (1996), "Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2d range finder," in In Proceedings of the IEEE International Conference n Robotics and Automation, Minneapolis, Minnesota, pp. 901–908, 1996.

K. O. Arras and R. Siegwart (1997), "Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building," in Symposium on Intelligent Systems and Advanced Manufacturing, 1997.

P. Jensfelt and H. Christensen (1998), "Laser Based Position Acquisition and Tracking in an Indoor Environment," in In Proceedings of the IEEE International, 1998.

S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick (2003), "Weighted line fitting algorithms for mobile robot map building and efficient data representation," in In ICRA, pp. 14–19, 2003.

V. Nguyen, S. Gachter, A. Martinelli, N. Tomatis, and R. Siegwart (2007), "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics," Auton. Robots, vol. 23, pp. 97–111, August 2007. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1286035.1286041>

M. Zuliani (2009), RANSAC for Dummies. University of California, Santa Barbara, Octubre 2009.

J. V. Domínguez (2008), Tesis: Implementación del FKE para SLAM. España: Escuela de ingenieros de Sevilla, 2008.

T Bousfield (1999), "A practical guide to Autocad Autolisp". Addison-Wesley Longman Publishing co., Inc., Boston, Ma, USA. 1999

AUTHORIZATION AND DISCLAIMER

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.