

LOCALIZACIÓN Y MAPEO DE UN ROBOT MÓVIL UTILIZANDO EL FILTRO EXTENDIDO DE KALMAN Y FILTRO DE PARTÍCULAS

AUTORES:

Cárdenas Acosta, Ronald[†]
Huanay Quispe, Jaspers W.[†]
{roncardenasacosta, hwilliamnq}@gmail.com

ASESOR:

Ivan Calle Flores[†]
{icallef@uni.pe}

[†]Universidad Nacional de Ingeniería
Facultad de Ingeniería Mecánica

Escuela Profesional de Ingeniería Mecatrónica
Grupo de Investigación en Sistemas de Control e Inteligencia Artificial – GISCIA
Av. Tpac Amaru 210

Resumen—La localización de un robot autónomo en ambientes con presencia de agentes extraños que puede considerarse como incertidumbre, es crítica en muchas aplicaciones donde el robot debe conocer su posición para cada instante de tiempo ya que de esta depende el éxito o fracaso de la tarea asignada. En este trabajo se utilizarán técnicas probabilísticas para una mejor aproximación de la localización mediante el filtro extendido de kalman y el filtro de partículas para un vehículo de configuración Ackerman. Esta configuración es la mas utilizada en la navegación autónoma.

Palabras Clave—Localización, Filtro extendido de Kalman, Robot Móvil, Ackerman, Filtro de partículas, Robótica probabilística.

I. INTRODUCCIÓN

LA esencia de la robótica probabilística trata de estimar los estados a partir de la información obtenida de los sensores. En muchas aplicaciones de robótica es fácil decidir la acción a realizar si el entorno donde se moverá y los obstáculos son conocidos. Desafortunadamente estas variables no son medibles directamente, en vez de eso el robot tiene que confiar en sus sensores para la obtención de esta información, pero los sensores solo pueden obtener información parcial de estas variables y en la mayoría de veces son distorsionadas por el ruido generado por agentes extraños. En consecuencia el objetivo de los algoritmos probabilísticos de estimación de estados son calcular la distribución belief sobre un posible estado del mundo real.

Existen 2 tipos fundamentales de interacción del robot y su entorno: El robot puede influir los estados de su entorno a traves de sus actuadores y puede obtener información del estado a traves de sus sensores. Los conceptos de interacción son los siguientes:

A. Medida de los sensores

Percepción es el proceso por el cual el robot usa sus sensores para obtener información acerca del estado de su entorno, un ejemplo puede ser la imagen tomada por la cámara de un robot. El resultado de dicha interacción perceptual lo llamaremos medición. Normalmente la medición obtenida llega con un retraso lo que significa que corresponde a los estados de un tiempo anterior.

B. Acción de Control

Cambian los estados del mundo (entorno) los ejemplos de la acción de control del robot incluyen el movimiento del robot y la manipulación de objetos. Aun cuando el robot no realiza ninguna acción por si misma el estado usualmente cambia.

En este trabajo se realizará la localización de un vehículo tipo Ackerman y estará dividido en las siguientes partes: Primero se desarrollará el modelamiento del robot (II), luego se desarrollará el modelamiento de los sensores (III), y la localización para el modelo elegido basado en velocidades usando el filtro extendido de Kalman (IV), es decir: $u = \begin{bmatrix} v \\ \varphi \end{bmatrix}$, donde u representa a la variable de control que en el caso de la configuración Ackerman es la velocidad (v) y ángulo de dirección principal (φ). Finalmente se concluye presentando la simulación y resultados (V) obtenidos, adicionalmente se realiza la comparación del filtro extendido de kalman con el filtro de partículas.

La nomenclatura que se empleará es la siguiente:

II. MODELAMIENTO DEL ROBOT

La Arquitectura Ackerman (ver Fig. 1) es una de las más usadas actualmente en los vehículos motorizados, la ventaja de este modelo se encuentra en que evita el deslizamiento de las llantas bajo ciertas condiciones. Este diseño cuenta con tracción

Estado del robot:	
(x, y)	coordenadas del robot (CG)
θ	orientación global del robot
Dimensiones del robot:	
L	longitud del robot
b	distancia entre ruedas motrices
D	diámetro de las ruedas
R	radio de curvatura
(x_c, y_c)	posición del CIR del robot
Estado del sensor:	
r	distancia del robot a la marca
ϕ	ángulo entre el robot y la marca
s	Identificador de la referencia (landmark)
Controles del robot:	
v	velocidad lineal del robot
$\varphi_s = \varphi$	ángulo de dirección (steering angle)
Dimensiones del mapa en número de celdas:	
M	número de celdas en x
N	número de celdas en y

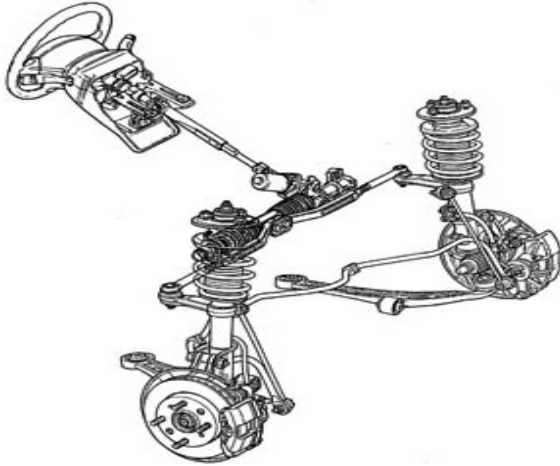


Fig. 1: Vehículo tipo Ackerman

trasera (dos ruedas motrices) y dos ruedas directrices (gobernadas por una volante).

Una simplificación de este modelo comúnmente usado en seguimiento de trayectorias es el modelo de bicicleta (ver Fig. 2).

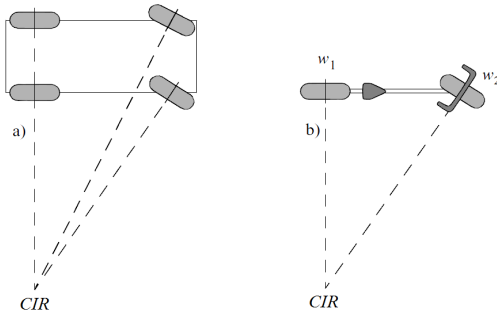


Fig. 2: a) Configuración Ackerman, b) Modelo bicicleta

Para el modelamiento tomamos en cuenta las siguientes consideraciones:

- 1) Se reemplazan las cuatro ruedas por solo dos (una delantera directriz y una trasera motriz).
- 2) El vehículo se mueve sobre un plano.
- 3) El tratar a este problema como el de una bicicleta aproxima el modelo del carro razonablemente a bajas velocidades y ángulos de dirección moderados.
- 4) Las llantas no presentan deslizamiento lateral.

- 5) Posee como centro CIR a la intersección de las rectas perpendiculares a todas las llantas (ver Fig. 3).

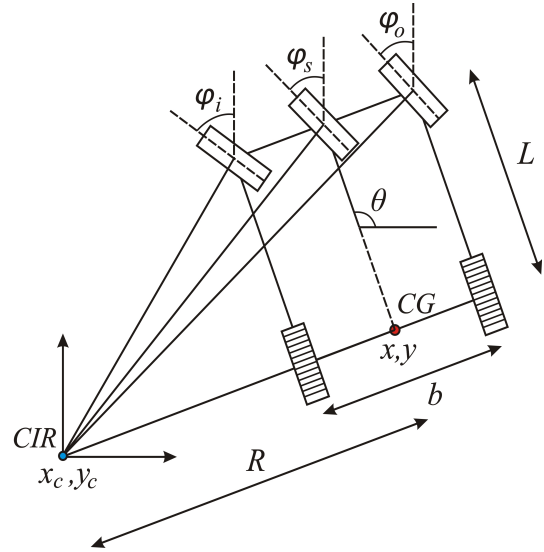


Fig. 3: El CIR es generado por la intersección de las perpendiculares a las ruedas

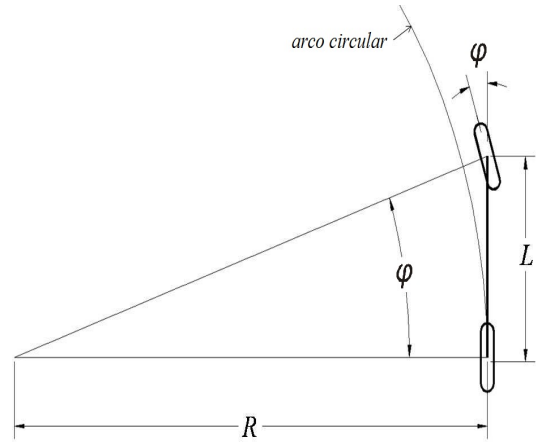


Fig. 4: Geometría básica del modelo bicicleta

Se obtienen las siguientes relaciones geométricas (ver Fig. 3 y Fig. 4):

$$\cot(\varphi_o) = \frac{R + b/2}{L} \quad (1)$$

$$\cot(\varphi_i) = \frac{R - b/2}{L} \quad (2)$$

$$\cot(\varphi_o) - \cot(\varphi_i) = \frac{b}{L} \quad (3)$$

$$\cot(\varphi_s) = \frac{b}{L} \quad (4)$$

$$\tan(\varphi_s) = \frac{L}{R} \quad (5)$$

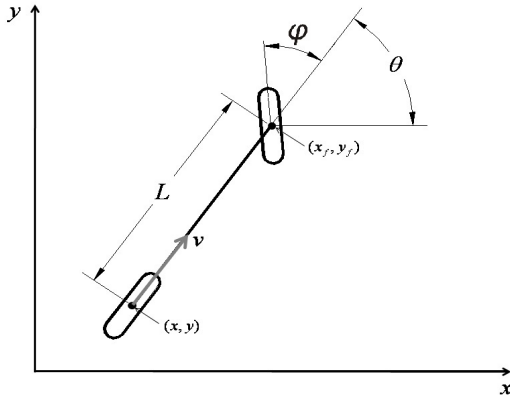


Fig. 5: Modelo cinemático de la bicicleta

Consideramos que la velocidad constante, las ecuaciones de restricción no holonómica para las ruedas delanteras y traseras son (ver Fig. 5):

$$\dot{x}_f \sin(\theta + \varphi) - \dot{y}_f \cos(\theta + \varphi) = 0 \quad (6)$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0 \quad (7)$$

(x_f, y_f) es la coordenada global de la rueda directriz principal (imaginaria), pudiendo ser expresadas como:

$$\begin{aligned} x_f &= x + L \cos(\theta) \\ y_f &= y + L \sin(\theta) \end{aligned} \quad (8)$$

Eliminando (x_f, y_f) de la Eq. 6:

$$\begin{aligned} 0 &= \frac{d(x + L \cos(\theta))}{dt} \sin(\theta + \varphi) - \frac{d(y + L \sin(\theta))}{dt} \cos(\theta + \varphi) \\ &= (\dot{x} - \dot{\theta}L \sin(\theta) \sin(\theta + \varphi)) - (\dot{y} - \dot{\theta}L \cos(\theta) \cos(\theta + \varphi)) \\ &= \dot{x} \sin(\theta + \varphi) - \dot{y} \cos(\theta + \varphi) \\ &\quad - \dot{\theta}L \sin(\theta)(\sin(\theta) \cos(\varphi) + \cos(\theta) \sin(\varphi)) \\ &\quad - \dot{\theta}L \cos(\theta)(\cos(\theta) \cos(\varphi) + \sin(\theta) \sin(\varphi)) \\ &= \dot{x} \sin(\theta + \varphi) - \dot{y} \cos(\theta + \varphi) \\ &\quad - \dot{\theta}L \sin^2(\theta) \cos(\varphi) - \dot{\theta}L \cos^2(\theta) \cos(\varphi) \\ &\quad - \dot{\theta}L \sin(\theta) \cos(\theta) \sin(\varphi) + \dot{\theta}L \cos(\theta) \sin(\theta) \sin(\varphi) \\ &= \dot{x} \sin(\theta + \varphi) - \dot{y} \cos(\theta + \varphi) - \dot{\theta}L(\sin^2(\theta) + \cos^2(\theta)) \cos \varphi \end{aligned}$$

$$= \dot{x} \sin(\theta + \varphi) - \dot{y} \cos(\theta + \varphi) - \dot{\theta}L \cos \varphi \quad (9)$$

La restricción no holonómica en la rueda trasera (Eq. 7) se satisface con $\dot{x} = \cos(\theta)$ y $\dot{y} = \sin(\theta)$ y cualquier múltiplo escalar del mismo. Esta escalar corresponde a la velocidad longitudinal v , de tal manera que:

$$\dot{x} = v \cos(\theta)$$

$$\dot{y} = v \sin(\theta) \quad (10)$$

Aplicando esto a la restricción en la rueda delantera (Eq. 9) se obtiene una solución para $\dot{\theta}$

$$\begin{aligned} \dot{\theta} &= \frac{\dot{x} \sin(\theta + \varphi) - \dot{y} \cos(\theta + \varphi)}{L \cos(\varphi)} \\ &= \frac{v \cos(\theta)(\sin(\theta) \cos(\varphi) + \cos(\theta) \sin(\varphi))}{L \cos(\varphi)} \\ &\quad - \frac{v \sin(\theta)(\cos(\theta) \cos(\varphi) + \sin(\theta) \sin(\varphi))}{L \cos(\varphi)} \\ &= \frac{v(\cos^2(\theta) + \sin^2(\theta)) \sin(\varphi)}{L \cos(\varphi)} \\ &= \frac{v \tan(\varphi)}{L} \end{aligned} \quad (11)$$

Resumiendo el estado del robot queda definido por:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \frac{v \tan(\varphi)}{L} \end{pmatrix} \quad (12)$$

Discretizando el sistema mediante euler:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t + Tv \cos(\theta) \\ y_t + Tv \sin(\theta) \\ \theta_t + \frac{Tv \tan(\varphi)}{L} \end{pmatrix} \quad (13)$$

Modelando el movimiento real:

$$\begin{pmatrix} \hat{v} \\ \hat{\varphi} \end{pmatrix} = \begin{pmatrix} v \\ \varphi \end{pmatrix} + \begin{pmatrix} \varepsilon_{\alpha_1 v^2 + \alpha_2 \varphi^2} \\ \varepsilon_{\alpha_3 v^2 + \alpha_4 \varphi^2} \end{pmatrix} \quad (14)$$

Entonces el modelo real del robot esta dado por:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + T\hat{v} \cos(\theta) \\ y + T\hat{v} \sin(\theta) \\ \theta_t + \frac{T\hat{v} \tan(\hat{\varphi})}{L} \end{pmatrix} \quad (15)$$

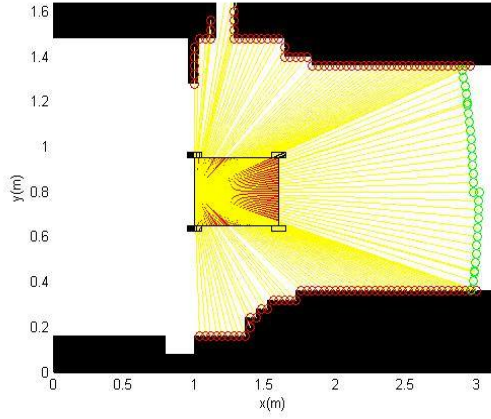


Fig. 6: Medida ideal del sensor

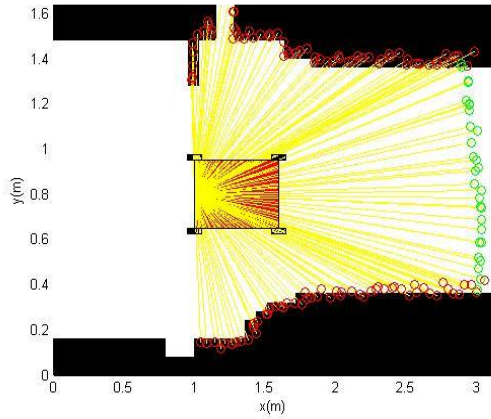


Fig. 7: Medida real del sensor

III. MODELAMIENTO DE LOS SENSORES

Empleamos un mapa de ocupación basado bloques(referencias) las cuales serán sensadas mediante sensores del tipo range finder.

A continuación se presenta el caso ideal (ver Fig. 6):

$$\begin{pmatrix} r_t^i \\ \phi_t^i \\ s_t^{i,j} \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ a \tan 2(m_{j,y} - y, m_{j,x} - x) - \theta \\ (i-1) * M + j \end{pmatrix} \quad (16)$$

Para el caso real (ver Fig. 7) se considera los cuatro componentes de ruido estudiados en el curso. Los parámetros intrínsecos del modelo se obtuvo mediante pruebas, se consideró los pesos de cada componente del ruido como sigue:

$$\begin{aligned} Z_{hit} &= 0.5 \\ Z_{short} &= 0.4499 \\ Z_{max} &= 0.0001 \\ Z_{rand} &= 0.0500 \end{aligned}$$

$$\begin{pmatrix} \hat{r}_t^i \\ \hat{\phi}_t^i \\ \hat{s}_t^{i,j} \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ a \tan 2(m_{j,y} - y, m_{j,x} - x) - \theta \\ (i-1) * M + j \end{pmatrix} + \begin{pmatrix} \varepsilon_{\sigma_r^2} \\ \varepsilon_{\sigma_\phi^2} \\ \varepsilon_{\sigma_s^2} \end{pmatrix} \quad (17)$$

IV. MAPEO

El objetivo del algoritmo de mapeo grid es calcular el posterior sobre el mapa dado todas las mediciones y posiciones anteriores.

$$p(m_t | z_{1:t}, x_{1:t})$$

los controles anteriores, $u_{1:T}$, no se tienen en cuenta en el problema de mapeo dado que se conoce el camino de antemano. El enfoque estándar en grid de ocupación es simplificar el problema de estimar el mapa en una colección de problemas separados, planteándose lo siguiente:

$$p(m_i | z_{1:t}, x_{1:t})$$

para todas las celdas m_i , donde se utilizará probabilidades logarítmicas.

El algoritmo calcula el modelo inverso de la medición se inicializa el mapa de ocupación con una probabilidad inicial l_0 .

$$l_0 = \log(p(m_i = 1)/p(m_i = 0))$$

$$= \log(p(m_i)/(1 - p(m_i)))$$

se utilizara el algoritmo en la figura (Fig. 8)

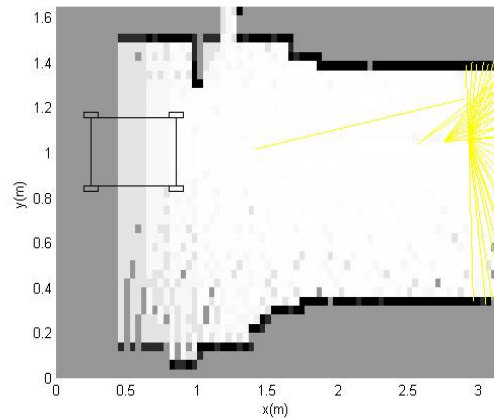


Fig. 8: Estimacion de mapa de ocupacion

También se observa en la figura (Fig. 9) en este algoritmo, se implementa la función inversa sensor model, el cual asigna una probabilidad $locc$ a todas las celdas cercanas a al rango de medición. si la celda esta en el rango de medición pero no cercana a la

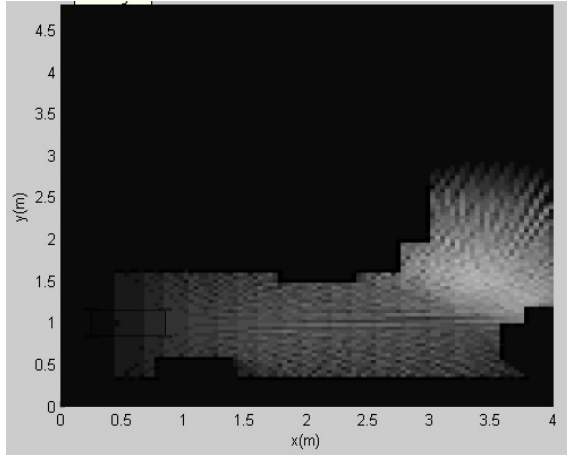


Fig. 9: Estimacion de mapa de ocupacion

medicion, o si el rayo no llega a rebotar, se asigna a la celda una probabilidad l_{free} . Se tiene que

$$l_{free} < l_0 < l_{occ}$$

el las pruebas realizadas se ha asumido las siguientes relaciones:

$$l_{occ} = 0.2 * l_0$$

$$l_{free} = 2.5 * l_0$$

El mapa esta dividido en celdas de 4 cm de espesor. El mapa real se simula asignándole a cada celda un valor binario de ocupación, indicando si está ocupado o no.

Los sensores range finder miden la distancia al objeto más cercano en la dirección que es disparado en rayo láser, o el cono, en el caso de sensores ultrasónicos. En este trabajo se consideró sensor laser cuyo rango de medición es de 180 grados.

V. FILTRO EXTENDIDO DE KALMAN

Usar la información de sus sensores para localizar al robot en su propio entorno es el problema más fundamental que presenta un robot móvil con capacidades autónomas. Puesto que la cinemática del robot y la medida de los sensores son sistemas no lineales (Eq. 15 y Eq. 17), aproximamos dicho modelos de tal modo que usemos las ecuaciones de los casos ideales mas ruido blanco.

El filtro extendido de Kalman es una generalización del filtro de Kalman para sistemas no lineales y una implementación del filtro de Bayes en el caso de sistemas gaussianos lineales, cuya condición es que se conozca el belief inicial asumiéndose que este es del tipo gaussiano.

El núcleo central de este método de localización es la estimación del estado usando toda la data disponible. La creencia del robot (belief) respecto al estado 'x' esta dada por la distribución de la Eq. 18, debiendo ser conocido el belief inicial apriori.

$$belx_t = p(x_t | z_{1:t}, u_{1:t}) \quad (18)$$

Mientras el robot no sea retroalimentado por la medición de sus sensores, su belief ira disminuyendo,

es decir la incertidumbre de su estado aumentará ya que su localización se basa solo en el desarrollo de su modelo (predicción). Si el robot llega a sensor alguna obstáculo (celda ocupada) este actualizará su localización corrigiendo de este modo su estado.

Se presenta a continuación el algoritmo que desarrolla el Filtro extendido de Kalman:

Entradas: $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$

Predicción:

$$\begin{aligned} \bar{\mu}_t &= g(u_t, \mu_{t-1}) \\ \bar{\Sigma}_t &= G_t \Sigma_{t-1} G_t^T + R_t \end{aligned}$$

Corrección:

$$\begin{aligned} K_t &= \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \\ \mu_t &= \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \end{aligned}$$

$$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$$

Retorna: μ_t, Σ_t

Procedemos a calcular las variables usadas en el algoritmo iterativo presentado anteriormente.

Cinemática del robot:

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \quad (19)$$

Medida de los sensores:

$$z_t = h(x_t) + \delta_t \quad (20)$$

Donde:

ε_t : Vector (nx1) de ruido gaussiano de media cero, y matriz de covarianza R_t

δ_t : Vector (kx1) de ruido gaussiano de media cero, y matriz de covarianza Q_t

Predicción:

De la ecuación 13

$$x_t = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix}$$

$$g(u_t, x_{t-1}) = \begin{pmatrix} x_{t-1} + Tv \cos(\theta) \\ y_{t-1} + Tv \sin(\theta) \\ \theta_{t-1} + \frac{Tv \tan(\varphi)}{L} \end{pmatrix}$$

$$\varepsilon_t = N(0, R) \quad (21)$$

Hallando la covarianza R_t :

$$M_t = \begin{pmatrix} \alpha_1 v_t^2 + \alpha_2 \varphi_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4 \varphi_t^2 \end{pmatrix} \quad (22)$$

$$V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t} \quad (23)$$

$$V_t = \begin{pmatrix} T \cos \theta & 0 \\ T \sin \theta & 0 \\ \frac{T}{L} \tan \varphi & \frac{T}{L} v \sec^2 \varphi \end{pmatrix} \quad (24)$$

Entonces:

$$R_t = V_t M_t V_t^T \quad (25)$$

Hallando el jacobiano, producto de la linealización por series de Taylor:

$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

$$G_t = \begin{pmatrix} 1 & 0 & -Tv \sin \theta \\ 0 & 1 & Tv \cos \theta \\ 0 & 0 & 1 \end{pmatrix} \quad (26)$$

Corrección: De la ec. (17)

$$z_t = \begin{pmatrix} \hat{r}_t^i \\ \hat{\phi}_t^i \\ \hat{s}_t^i \end{pmatrix} \quad (27)$$

$$h(x_t) = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ a \tan 2(m_{j,y} - y, m_{j,x} - x) - \theta \\ s_j \end{pmatrix} \quad (28)$$

$$\delta_t = \begin{pmatrix} \varepsilon \sigma_r^2 \\ \varepsilon \sigma_\phi^2 \\ \varepsilon \sigma_s^2 \end{pmatrix} \quad (29)$$

Su matriz de covarianza ser:

$$Q_t = \begin{pmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_s^2 \end{pmatrix} \quad (30)$$

Hallando su respectivo jacobiano:

$$H_t = \frac{\partial h(\bar{\mu}_t, m)}{\partial x_t}$$

$$q = (m_{j,x} - \hat{u}_{t,x})^2 + (m_{j,y} - \hat{u}_{t,y})^2$$

$$H_t = \begin{pmatrix} -\frac{m_{j,x} - \hat{u}_{t,x}}{\sqrt{q}} & -\frac{m_{j,y} - \hat{u}_{t,y}}{\sqrt{q}} & 0 \\ \frac{m_{j,y} - \hat{u}_{t,y}}{q} & -\frac{m_{j,x} - \hat{u}_{t,x}}{q} & -1 \\ 0 & 0 & 0 \end{pmatrix} \quad (31)$$

```

1: Algorithm likelihood_field_range_finder_model( $z_t, x_t, m$ ):
2:    $q = 1$ 
3:   for all  $k$  do
4:     if  $z_t^k \neq z_{\max}$ 
5:        $x_{z_t^k} = x + x_{k,\text{sens}} \cos \theta - y_{k,\text{sens}} \sin \theta + z_t^k \cos(\theta + \theta_{k,\text{sens}})$ 
6:        $y_{z_t^k} = y + y_{k,\text{sens}} \cos \theta + x_{k,\text{sens}} \sin \theta + z_t^k \sin(\theta + \theta_{k,\text{sens}})$ 
7:        $\text{dist}^2 = \min_{x', y'} \{ (x_{z_t^k} - x')^2 + (y_{z_t^k} - y')^2 \mid (x', y') \text{ occupied in } m \}$ 
8:        $q = q \cdot (z_{\text{hit}} \cdot \text{prob}(\text{dist}^2, \sigma_{\text{hit}}^2) + \frac{z_{\text{random}}}{z_{\max}})$ 
9:   return  $q$ 

```

Fig. 10: Algoritmo utilizado para la localización por filtro de partículas

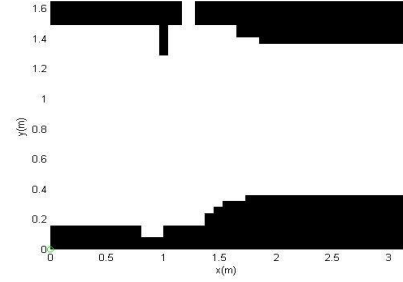


Fig. 11: Mientras el robot no sea retroalimentado por la medición de sus sensores su belief ira disminuyendo, es decir la incertidumbre de su estado aumentará ya que su localización se basará solo en el desarrollo de su modelo (predicción). Si el robot llega a sensar algun obstáculo este actualizará su localización haciendo su debida corrección.

VI. FILTRO DE PARTÍCULAS

El muestreo de la probabilidad de medida es sumamente importante para poder muestrear la probabilidad de una medición en el mapa, se emplea el modelo basado en puntos finales (Likelihood Fields Range Finder Model), el cual mide la distancia del punto final de un rayo al obstáculo más cercano.

El algoritmo para la localización por filtro de partículas es como se ve en la figura (Fig. 10).

Se puede observar en el mapa de la (Fig. 11) donde se muestra el mapa real

Para el mapa real también se tiene su campo de probabilidades y se puede observar en la figura (Fig. 12)

En nuestro caso se trabajará en el mapa de la figura (Fig. 13)

Para nuestro mapa real también se tiene su campo de probabilidades y se puede observar en la figura (Fig. 14)

VII. SIMULACIÓN Y RESULTADOS

Se presenta los resultados obtenidos para el filtro extendido de Kalman. El rango máximo de medición

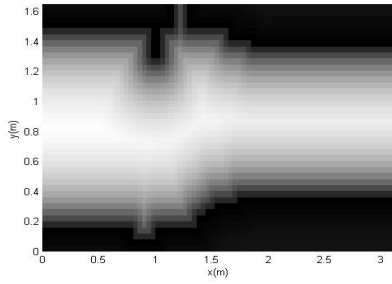


Fig. 12: Mapa de probabilidades del mapa real.

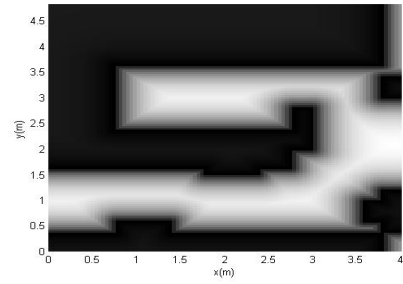


Fig. 14: Mapa de probabilidades del mapa real.

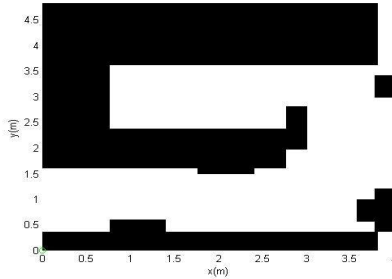


Fig. 13: Mapa de probabilidades del mapa real que utilizaremos para nuestra simulación.

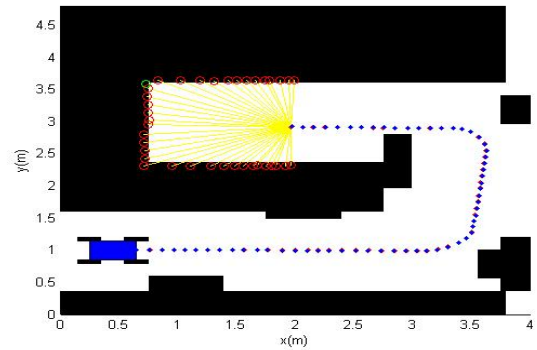


Fig. 15: Para parámetros de ruido relativamente bajos

será 1.5m para todas nuestras pruebas.

Los parámetros del sensor son:

$$\sigma_R = 0.02$$

$$\sigma_B = (1.0 \cdot \pi / 180)$$

$$\sigma_C = 0.01$$

Los parámetros del error son:

$$\alpha_1 = 1e-2$$

$$\alpha_2 = 1e-3$$

$$\alpha_3 = 1e-3$$

$$\alpha_4 = 1e-2$$

Para estos valores se tiene el resultado como se observa en la figura (Fig. 15).

Para parámetros de ruido:

$$\sigma_R = 0.02$$

$$\sigma_B = (2.0 \cdot \pi / 180)$$

$$\sigma_C = 0.01$$

y parámetros del error:

$$\alpha_1 = 1$$

$$\alpha_2 = 1e-1$$

$$\alpha_3 = 1e-1$$

$$\alpha_4 = 1$$

Para estos valores altos se tiene el resultado como se observa en la figura (Fig. 16).

Para el filtro de partículas de igual modo que para el filtro extendido de kalman se realizaron varias pruebas para distintos valores de ruido en el muestreo de la probabilidad y ruido en los sensores, por ejemplo para los parámetros de ruido siguientes:

Parámetros de sensor:

$$\sigma_R = 0.02$$

$$\sigma_B = (1.0 \cdot \pi / 180)$$

$$\sigma_C = 0.01$$

y ruido en el muestreo de probabilidad:

$$Z_{hit} = 0.35$$

$$Z_{short} = 0.1$$

$$Z_{max} = 550$$

$$Z_{rand} = 1$$

se tiene el resultado como se muestra en la (Fig. 17).

Para parámetros de ruido relativamente bajos se tiene la (Fig. 18).

Los resultados mejor obtenidos de localización fueron los que se muestran en el mapa de la (Fig. 19). su distribución de probabilidad (likelihood) es la que se observa en la (Fig. 20).

Para el filtro extendido de Kalman se tiene los resultados de la (Fig. 21).

Para el filtro de partículas se tiene los resultados de la (Fig. 22).

VIII. CONCLUSIONES

- La localización mediante filtro extendido de kalman present muy buenos resultados, localizando casi siempre en la primera iteración, para parámetros de ruido adecuados.
- La localización mediante filtro de partículas depende principalmente del likelihood field para pesar las partículas. Se tuvo dificultades al momento de calibrar al tanteo los parámetros

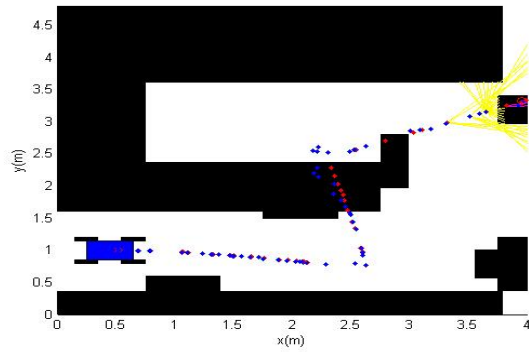


Fig. 16: Para parámetros de ruido relativamente altos

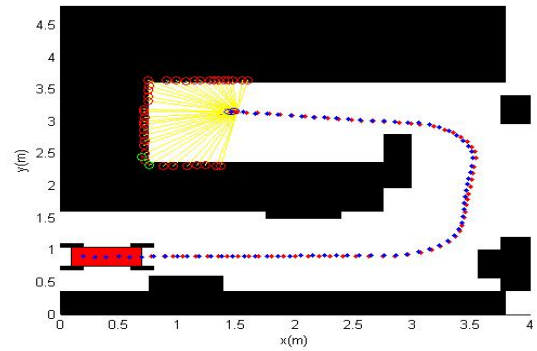


Fig. 18: Para parámetros de ruido relativamente bajos

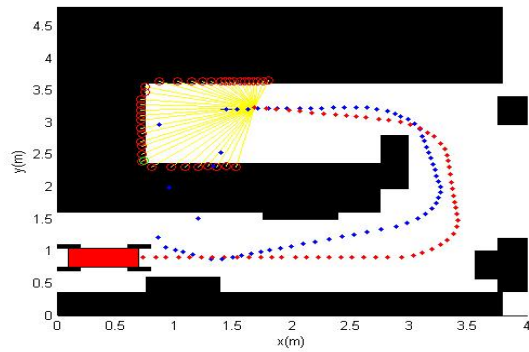


Fig. 17: Para parámetros de ruido relativamente altos

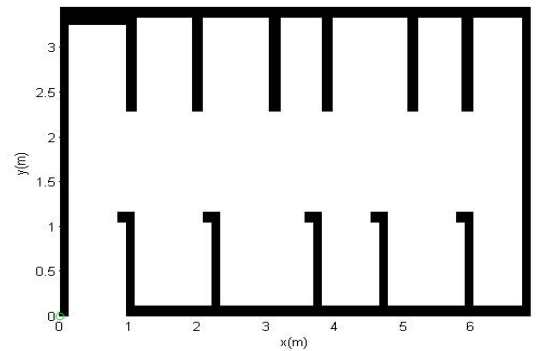


Fig. 19: Para parámetros de ruido relativamente bajos

intrínsecos del sensor para que las partículas convergan.

- El mapeo mediante el modelo inverso del sensor present muy buenos resultados. Aunque la actual eficiencia de la implementación es aceptable, es posible optimizarlo para un mapeo en tiempo real.

REFERENCIAS

- [1] Sebastian THRUN, Wolfram BURGARD, Dieter FOX *Probabilistic Robotics*. EARLY DRAFT & 1999-2000
- [2] Roland SIEGWART, Illah R. NOURBAKHSH *Introduction to Autonomous Mobile Robots*. Massachusetts Institute of Technology & 2004
- [3] Jarrod M. SNIDER *Automatic Steering Methods for Autonomous Automobile Path Tracking*. Robotics Institute & Carnegie Mellon University & Pittsburgh, Pennsylvania & 2009
- [4] Wikipedia, the free encyclopedia. *Ackermann steering geometry*. 24-Oct-2012.

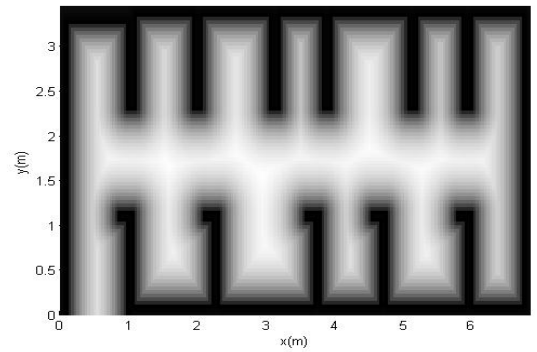


Fig. 20: Para parámetros de ruido relativamente bajos

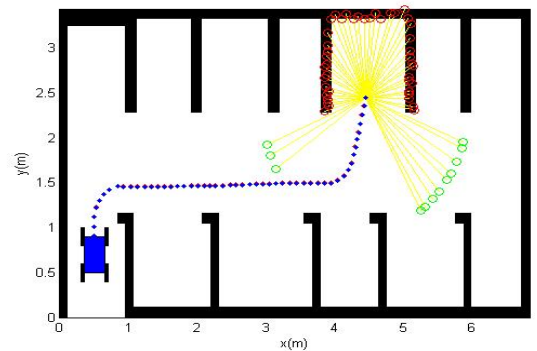


Fig. 21: Para parámetros de ruido relativamente bajos

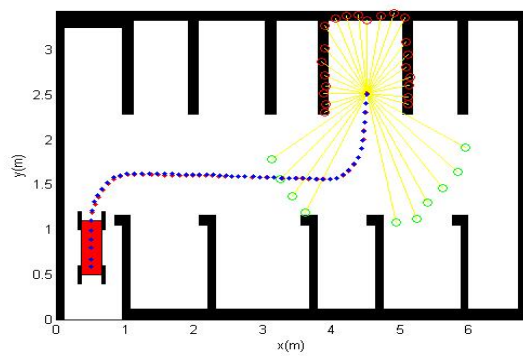


Fig. 22: Para parámetros de ruido relativamente bajos