

# Sistema de SLAM flexible y escalable con estimación completa de movimiento en 3D (Hector SLAM)

Luis Servin  
Taller TRA, Universidad Nacional Autónoma de México

May 11, 2016

## Resumen

Se presenta una técnica de mapeo de ocupación cuadriculado (*occupancy grid map*) en línea (*online learning*) que requiere un bajo costo computacional. Este combina un método robusto de emparejamiento de puntos (*scan matching*), a través de un telémetro láser (LIDAR), con la estimación de la posición en el espacio con base en el sensado de las propiedades inerciales. Como resultado se obtiene un sistema lo suficientemente preciso para funcionar dentro de los escenarios considerados, sin la necesidad de contar de manera explícita con una técnica de cerrado de ciclos (*loop closing*).

## 1 Introducción

La habilidad de aprender el modelo del ambiente que lo rodea y localizarse a sí mismo dentro de él es un requerimiento indispensable para que un robot considerado autónomo sea capaz de operar dentro de escenarios reales.

### Localización y Mapeo Simultáneos (SLAM)

La solución presentada se apegue a los lineamientos establecidos en la API del sistema de navegación (*navigation stack*) de ROS (*Robot Operating System*). Gracias a esto puede ser fácilmente intercambiada por otras soluciones (algoritmos de SLAM) que cumplan con dichos criterios. Este sistema puede ser usado para realizar SLAM en escenarios pequeños que no cuenten con grandes circuitos cerrados y que además se aprovechen las altas frecuencias de muestreo de los sistemas modernos de medición por láser (LIDAR).

### Estado de la técnica

Existe una basta cantidad de soluciones aceptables para espacios internos tipo oficina aplicando un filtro de partículas Rao-Blackwell, i.e. **G-mapping**. Aunque estas soluciones trabajan mejor en ambientes planos, además que dependen altamente de un sistema de odometría suficientemente preciso.

Los sistemas de SLAM actuales se puede categorizar en dos tipos principales:

- **SLAM frontend**: Usado para estimar el movimiento y posición sobre la marcha en tiempo real.
- **SLAM backend**: Usado para realizar optimizaciones sobre una representación gráfica de la posición dadas ciertas restricciones entre las posiciones que han sido generadas antes de usar el sistema "frontend"

La solución planteada en este artículo es de tipo *frontend*.

## 2 Descripción General del Sistema

La técnica a desarrollar busca estimar el estado completo de los seis grados de libertad de la plataforma, el cual consiste en su vector de traslación y rotación. Para lograrlo se divide en dos componentes principales

- **Filtro de navegación:** Combina información de la unidad de medidas inerciales (IMU) junto con otros sensores que se encuentren disponibles para lograr una estimación 3D consistente.
- **SLAM 2D:** Usado para obtener la información de posición y orientación dentro del plano de tierra del sistema.

Se define el sistema de coordenadas global como un sistema de mano derecha. Tiene su origen en el punto inicial, el eje z en dirección positiva hacia arriba y el eje x apuntando en la dirección del ángulo yaw (ángulos de euler,) o frente de la plataforma.

El estado completo de posición y orientación en el espacio esta representado por  $x = (\Omega^T \ p^T \ v^T)^T$  donde  $\Omega = (\phi, \theta, \psi)^T$  son los *ángulos de euler*: roll, pitch y yaw. Además  $p = (p_x, p_y, p_z)^T$  y  $v = (v_x, v_y, v_z)^T$  representan la posición y velocidad de la plataforma con respecto a sistema coordenado de navegación.

Los datos obtenidos del sistema de medición inercial componen el vector de entrada  $u = (\omega^T \ a^T)^T$  donde  $\omega = (\omega_x, \omega_y, \omega_z)^T$  y  $a = (a_x, a_y, a_z)^T$  representan las velocidades y aceleraciones angulares respectivamente. El estado completo de posición y orientación de un cuerpo rígido es representado por:

$$\dot{\Omega} = E_{\Omega} \cdot \omega \quad (1)$$

$$\dot{p} = v \quad (2)$$

$$\dot{v} = R_{\Omega} \cdot a + g \quad (3)$$

Donde  $R_{\Omega}$  representa la matriz cosenos que transforma un vector del marco de referencia en el cuerpo al sistema coordenado de navegación.  $E_{\Omega}$  transforma las velocidades angulares del cuerpo a las derivadas de los ángulos de euler.  $g$  es el vector de gravedad.

### 3 SLAM 2D

Se hace uso de un mapa de ocupación cuadriculado, debido a su capacidad de representar ambientes arbitrarios. Se trata de un método probado para realizar la localización de un robot móvil utilizando sensores láser (LIDAR) en entornos reales. Ya que el telémetro láser puede estar sujeto a movimientos en sus 6 grados de libertad, las lecturas que se obtienen deben ser transformadas a un marco de coordenadas local estabilizado en el plano usando una estimación de las posición real del sensor.

#### 3.1 Acceso al mapa

La naturaleza discreta de los mapas de ocupación de cuadrículas no permite el cálculo directo de valores interpolados o derivadas. Por lo tanto para realizar su cálculo se utiliza el siguiente método:

Dado un mapa coordenado continuo  $P_m$ , la probabilidad de ocupación  $M(P_m)$  así como el gradiente  $\nabla M(P_m) = (\frac{\partial M}{\partial x}(P_m), \frac{\partial M}{\partial y}(P_m))$  pueden ser aproximados usando los cuatro puntos mas cercanos a punto de interés a través de una interpolación lineal en los ejes  $x, y$ :

$$\begin{aligned} M(P_m) \approx & \frac{y - y_0}{y_1 - y_0} \left( \frac{x - x_0}{x_1 - x_0} M(P_{11}) + \frac{x_1 - x}{x_1 - x_0} M(P_{01}) \right) \\ & + \frac{y_1 - y}{y_1 - y_0} \left( \frac{x - x_0}{x_1 - x_0} M(P_{10}) + \frac{x_1 - x}{x_1 - x_0} M(P_{00}) \right) \end{aligned} \quad (4)$$

Donde las derivadas puedes ser aproximadas por:

$$\begin{aligned}\frac{\partial M}{\partial x}(P_m) &\approx \frac{y - y_0}{y_1 - y_0}(M(P_{11}) - M(P_{01})) \\ &\quad + \frac{y_1 - y}{y_1 - y_0}(M(P_{10}) - M(P_{00}))\end{aligned}\tag{5}$$

$$\begin{aligned}\frac{\partial M}{\partial y}(P_m) &\approx \frac{x - x_0}{x_1 - x_0}(M(P_{11}) - M(P_{10})) \\ &\quad + \frac{x_1 - x}{x_1 - x_0}(M(P_{01}) - M(P_{00}))\end{aligned}\tag{6}$$

Se debe hacer notar que la cuadrícula dentro del mapa se encuentra distribuida de tal manera que todas sus unidades se encuentran separadas a una distancia regular de 1 unidad, lo cual simplifica la aproximación del gradiente.

### 3.2 Emparejamiento de puntos (Scan matching)

**Scan matching** es el proceso de comparar y alinear un conjunto de lecturas obtenidas por un telémetro, con otras tomadas anteriormente o contra un mapa ya existente. Para muchos sistemas robóticos la exactitud y precisión de las lecturas obtenidas por un sensor láser es mucho mas grande que los datos de odometría, si es que este sistema se encuentra disponible.

Esta técnica busca optimizar dicho proceso de *scan matching* de tal manera que se aproveche la información obtenida hasta el momento. Ya que las lecturas del laser se encuentren alineadas con el mapa existente esto provoca que implícitamente lo estén con todas las lecturas anteriores.

Partiendo del método de Gauss-Newton. Se busca encontrar la transformación  $\xi = (p_x, p_y, \psi)^T$  tal que minimice:

$$\xi^* = \underset{\xi}{argmin} \sum_{i=1}^n [1 - M(S_i(\xi))]^2\tag{7}$$

Obteniendo aquella transformación que produzca la mejor correlación entre las lecturas obtenidas con el mapa ya existente.  $S_i(\xi)$  representa las coordenadas con respecto a un sistema global de las lecturas actuales  $s_i = (s_{i,x}, s_{i,y})^T$ . Mientras que  $\xi$  es la posición del robot en el sistema coordenado global.

$$S_i(\xi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} \begin{pmatrix} s_{i,x} \\ s_{i,y} \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \end{pmatrix}\tag{8}$$

$M(S_i(\xi))$  calcula el valor de las celdas del mapa, dado  $S_i(\xi)$ . Dada una estimación inicial de  $\xi$  se busca  $\nabla \xi$  que optimice el error.

$$\sum_{i=1}^n [1 - M(S_i(\xi + \Delta\xi))]^2 \rightarrow 0\tag{9}$$

Realizando una expansión por series de Taylor obtenemos:

$$\sum_{i=1}^n \left[ 1 - M(S_i(\xi)) - \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \Delta\xi \right]^2 \rightarrow 0\tag{10}$$

Y estableciendo la derivada parcial con respecto a  $\Delta\xi$  como cero obtenemos:

$$2 \sum_{i=1}^n \left[ \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right]^T \left[ 1 - M(S_i(\xi)) - \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \Delta\xi \right] = 0\tag{11}$$

Resolviendo para  $\Delta\xi$  se obtiene la ecuación de Gauss-Newton para el problema de minimización:

$$\Delta\xi = H^{-1} \sum_{i=1}^n \left[ \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right]^T [1 - M(S_i(\xi))]\tag{12}$$

Donde:

$$H = \left[ \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right]^T \left[ \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right] \quad (13)$$

Una aproximación para obtener el gradiente del mapa  $\nabla M(S_i(\xi))$  se obtiene con base en la ecuación (8):

$$\frac{\partial S_i(\xi)}{\partial \xi} = \begin{pmatrix} 1 & 0 & -\sin(\psi)s_{i,x} - \cos(\psi)s_{i,y} \\ 0 & 1 & \cos(\psi)s_{i,x} - \sin(\psi)s_{i,y} \end{pmatrix} \quad (14)$$

Para muchas aplicaciones una aproximación gaussiana de la incertidumbre de emparejamiento (match uncertainty) es aceptable. Otra solución aceptable es a través de una aproximación de la matriz Hessiana para obtener un estimado de la covarianza. Por lo tanto la matriz de covarianza esta aproximada por:

$$R = Var \{ \xi \} = \sigma^2 \cdot H^{-1} \quad (15)$$

### 3.3 Representación de mapas de múltiple resolución

Ya que el presente trabajo esta basado en un gradiente ascendente, existe el riesgo que dicho algoritmo quede estancado en un mínimo local.

En la solución presentada se usa opcionalmente múltiples mapas de ocupación, donde cada mapa menos detallada tiene la mitad de la resolución de su precedente. Diferentes mapas se mantienen en la memoria y son simultáneamente actualizados usando la posición generada por el proceso de alineación. El proceso de alineación comienza con el mapa menos detallado, el resultado de posición obtenido es usado como posición inicial para la estimación del siguiente nivel y así sucesivamente.

## 4 Estimación del estado en el espacio

### 4.1 Filtro de navegación

Se hace uso de un Filtro Kalman Extendido (EKF) con el modelo general de la plataforma definido por las ecuaciones (1), para realizar la estimación del estado completo en el espacio de la plataforma.

La actualización de posición y velocidad se obtiene de integrar la información de aceleración obtenida de las medidas inerciales. Además la posición y orientación en el plano es actualizada por el algoritmo de alineamiento de puntos "scan matching", mientras que para estimar la posición completa en el espacio (3D) se requiere de un sensor adicional de altura, como puede ser un barómetro o un sensor de distancia.

### 4.2 Integración con el SLAM

Con el objetivo de mejorar el rendimiento, se integra la información obtenida por el sistema de SLAM 2D con la estimación de la posición en el espacio a través del Filtro de Kalman Extendido (EKF). En este sentido si se quiere mejorar el proceso de alineamiento de puntos (scan matching) se puede tomar como punto de inicio la proyección de la posición estimada por el filtro de kalman, en el plano x-y de las coordenadas globales. Mientras que en la dirección contraria se usa la intersección de covarianza para fusionar la posición obtenida a través del algoritmo de SLAM con la estimación del estado de la plataforma en el espacio tridimensional.

Si denotamos la estimación realizada por el filtro de kalman por  $\hat{x}$  con covarianza  $P$  y la estimación de posición a través del algoritmo de SLAM como  $(\xi^*, R)$  como esta definido en (7) y (15). Se obtiene como resultado de fusionar esta información:

$$(P^+)^{-1} = (1 - \omega) \cdot P^{-1} + \omega \cdot C^T R^{-1} C \quad (16)$$

$$\hat{x}^+ = P^+ \left( (1 - \omega) \cdot P^{-1} \hat{x} + \omega \cdot C^T R^{-1} \xi^* \right)^{-1} \quad (17)$$

siendo  $C$  la matriz de observación, la cual proyecta el espacio completo de estados tridimensional en el subespacio planar (x, y,  $\theta$ ) del sistema de SLAM.

Utilizando una analogía entre el Filtro de Kalman y el Filtro de Información dicha ecuación también puede ser representada en su forma de covarianza como:

$$P^+ = P - (\omega - 1)^{-1} \cdot KCP \quad (18)$$

$$\hat{x}^+ = \hat{x} + K(\xi^* - C\hat{x}) \quad (19)$$

con

$$K = PC^T \left( \frac{1 - \omega}{\omega} \cdot R + C^T PC \right)^{-1} \quad (20)$$

Dicha representación es generalmente preferida, ya que computacionalmente es menos costosa.

## 5 Resultados

### 5.1 Escenario urbano de búsqueda y rescate

La presente solución fue utilizada en un vehículo de navegación terrestre (UGV) para realizar la exploración autónoma y detección de víctimas en un escenario "Urbano de búsqueda y rescate" como el que se presenta en la competencia "RoboCup Rescue League". El telémetro láser es estabilizado con respecto al movimiento en los ejes roll y pitch para mantenerlo alineado con el plano de tierra y con ello maximizar la ganancia de información obtenida a través de las lecturas del láser.

### 5.2 Litorales acuáticos

Ya que la presente solución no hace uso de la extracción de características como polilíneas, funciona de manera aceptable en ambientes muy poco estructurados como pueden ser costas con alta vegetación. El hecho que el sensor Hokuyo utilizado no regrese una lectura válida cuando el haz golpea el agua, se puede utilizar como ventaja para prevenir lecturas fallidas que provoquen errores en el sistema de SLAM.

### 5.3 Mapeo con sistemas embebidos

La solución presentada es lo suficientemente exacta para cerrar circuitos (close loops) encontrados en escenarios pequeños sin el uso explícito de un método especializado, manteniendo los requerimientos computacionales bajos y previniendo cambios bruscos al mapa estimado hasta el momento, durante el tiempo de prueba.