

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

Luis Servin

Taller TRA, Universidad Nacional Autónoma de México

December 1, 2015

1 Resumen

- Sistema de rápido aprendizaje en línea (online learning) de un mapa de ocupación de cuadrículas (occupancy grid map) que requiere de un bajo costo computacional.
- Este algoritmo combina un método robusto de emparejamiento de puntos (scan matching) usando las lecturas de un sistema láser (LIDAR) con un sistema de estimación de la posición en 3D basado en las lecturas de sensores inerciales.
- Mostramos que el sistema es suficientemente preciso ya que no requiere de una técnica explícita de ciclos cerrados (loop closing techniques) dados los escenarios que fueron considerados.

2 Introducción

La habilidad de aprender el modelo de un ambiente y localizarse a si mismo dentro de él es uno de los requerimientos indispensables para que un robot autónomo sea capaz de operar dentro de ambientes reales.

3 SLAM (Simultaneous Localization and Mapping) Localización y Mapeo Simultáneos

Nuestra solución se apega a los lineamientos de la API del sistema de navegación (navigation stack) de ROS (Robot Operating System) y por lo tanto puede ser fácilmente intercambiable por otras soluciones de SLAM que cumplan con dichos criterios.

El sistema introducido en este artículo tiene como objetivo obtener tanto una representación del ambiente lo mas precisa posible como la localización del robot dentro de la misma mientras se mantienen unos requerimientos computacionales bajos.

El sistema puede ser usado para realizar SLAM en escenarios pequeños donde grandes circuitos (loops) no tenga que ser cerrados y en los cuales se aprovechen

las altas frecuencias de muestreo de los sistemas modernos de medición por laser (LIDAR).

En esta solución se combina un sistema de SLAM 2D basado en la integración de las lecturas de un sensor laser (LIDAR) en un mapa 2D (plano) y la integración de un sistema de navegación 3D basados en un sensor de medidas inerciales (IMU)

4 Trabajo Relacionado (Estado de la técnica)

Existe una basta cantidad de soluciones aceptables para escenarios interiores tipo oficina usando el filtro de partículas de tipo Rao-Blackwell como "g-mapping". Estas soluciones trabajan mejor en ambientes planos, además dependen de que exista un sistema de odometría suficientemente preciso.

El SLAM 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".

5 Resumen del sistema (System Overview)

Nuestro sistema debe estimar el estado completo de los seis grados de libertad que consiste en la traslación y rotación de la plataforma. Para lograrlo se divide en dos componentes principales

- **Filtro de navegación (Navigation Filter):** Combina información de la unidad de medidas inerciales (IMU) junto con otros sensores que se encuentren disponibles para lograr una estimación 3D consistente.
- **Sistema de 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, que tiene su origen en el punto inicial con el eje z en dirección positiva hacia arriba y el eje x apuntando en la dirección del ángulo yaw (ángulos de euler) de la plataforma al inicio.

El estado completo de posición y orientación en el espacio es 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 representan 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 angulares y aceleraciones 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_{Ω} representa la matriz coseno que mapea un vector del marco de referencia en el cuerpo al sistema coordenado de navegación. E_{Ω} transforma las velocidades angulares del cuerpo a las derivadas de los ángulos de euler. g es el vector de gravedad.

6 SLAM 2D

Se hace uso de un mapa de ocupación por cuadrículas para ser capaz de representar ambientes arbitrarios. El cual es un método probado para realizar la localización de un robot móvil utilizando sensores láser (LIDAR) en entornos reales. Ya que el dispositivo láser puede estar sujeto a tener 6 grados de libertad de movimiento, las lecturas que se obtienen deben ser transformadas a un marco de coordenadas local estabilizado en el plano usando una estimación de la posición real del sensor. A través del uso de la orientación estimada de la plataforma y los valores de las juntas cinemáticas, las lecturas son convertidas en una nube de puntos de lecturas de posición final.

6.1 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 $\nabla M(P_m) = (\frac{\partial M}{\partial x}(P_m), \frac{\partial M}{\partial y}(P_m))$ puede ser aproximado usando los cuatro puntos mas cercanos a nuestro punto de interés a través de una interpolación lineal tanto en los ejes x y 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 pueden ser aproximadas por:

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

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

Se debe hacer notar que las rejillas del mapa están distribuidas de tal manera que todas se encuentran separadas a una distancia regular de 1 unidad, lo cual simplifica la aproximación del gradiente.

6.2 Emparejamiento de puntos (Scan matching)

Scan matching es el proceso de comparar y alinear un conjunto de lecturas obtenidas por láser 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 estos llegan a estar disponibles.

Nuestra solución esta basada en la optimización del proceso de "scan matching" en los puntos obtenidos por el láser con respecto al mapa aprendido hasta el momento. La idea básica es utilizar el método Gauss-Newton. Ya que las lecturas del laser se encuentren alineadas con el mapa existente esto provoca que implícitamente lo estén con todas las lecturas anteriores.

Se busca encontrar la transformación $\xi = (p_x, p_y, \psi)^T$ tal que minimice:

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

Por lo tanto se busca la transformación que produzca la mejor correlación entre las lecturas obtenidas con el mapa ya existente. $S_i(\xi)$ representa las coordenadas de las lecturas obtenidas con respecto al sistema global $s_i = (s_{i,x}, s_{i,y})^T$. Mientras que ξ representa la posición del robot en el sistema coordenado global.

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

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

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

Realizando una expansión por series de Taylor obtenemos:

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

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

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

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

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

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 (??):

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

Para muchas aplicaciones una aproximación gaussiana de la incertidumbre del emparejamiento (match uncertainty) es deseable.

Un método para obtenerla es usar la aproximación de la matriz Hessiana (Hessian Matrix) para llegar a la varianza estimada. De esta manera la matriz de covarianza se obtiene por:

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

6.3 Representación de mapas de múltiple resolución

Ya que el presente trabajo esta basado en un gradiente ascendente, es posible que el 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.

7 Estimación del estado en 3D

7.1 Filtro de navegación

Para realizar la estimación de las 6 coordenadas que definen la posición y orientación de la plataforma se hace uso de un filtro kalman extendido (EKF) con el modelo general de la plataforma definido por las ecuaciones (??).

La actualización de la velocidad y posición se obtiene de integrar la información de aceleración obtenida de las medidas inerciales. La posición y orientación en el plano es actualizada además 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 un barómetro, o un sensor de distancia.

7.2 Integración con el SLAM

Para mejorar el rendimiento, se debe integrar tanto la información obtenida a partir del SLAM 2D con la estimación de la posición en el espacio a través del filtro de kalam 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 (covariance intersection CI) 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 (ξ^*, R) como esta definido en (??) y (??). El resultado de la fusión de esta información esta dado por:

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

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

con C como la matriz de observación, la cual proyecta el espacio completo de estados tridimensional en el subespacio planar (x, y, θ) 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:

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

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

con

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

La cual es una de las formas preferidas ya que computacionalmente es menos costosa.

8 Resultados

8.1 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 sensor láser es estabilizado con respecto del movimiento en los ejes roll y pitch para mantener el láser alineado con el plano de tierra y con ello maximizar la ganancia de información obtenida a través de las lecturas del láser.

8.2 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 las costas con alta vegetación. Ya que el sensor Hokuyo

utilizado no regresa 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.

8.3 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.