Universidad del Valle de Guatemala
Departamento de Ingeniería Mecatrónica
MT3006 - Robótica 2
MSc. Miguel Zea

Laboratorio 8

Mapeo en 2D empleando landmarks puntuales

Objetivos

- Implementar un algoritmo de construcción de mapas con landmarks puntuales mediante un filtro de Kalman extendido y mediciones de un sensor de distancia omnidireccional.
- Recordar e implementar un control de rastreo de trayectorias para un robot móvil diferencial con ruedas.

Antecedentes

Supongamos que, luego de un proceso de estimación de pose, conocemos perfectamente el estado del robot ${}^{I}\boldsymbol{\xi} = \begin{bmatrix}\xi_1 & \xi_2 & \xi_3\end{bmatrix}^\top = \begin{bmatrix}x & y & \theta\end{bmatrix}^\top$ y colocamos a nuestro robot móvil en un entorno desconocido que contiene obstáculos puntuales. Nótese que si bien el robot conoce su pose, no tiene información alguna sobre el mapa del entorno pero aún así debe navegar dentro de él. Por esta razón, equiparemos al robot con un sensor de distancia que sea capaz de medir la magnitud y el ángulo de un obstáculo detectado, relativo al marco de referencia del robot, como se muestra en la Figura 1 (piense en un sensor de distancia omnidireccional como un LiDAR). De esto claramente

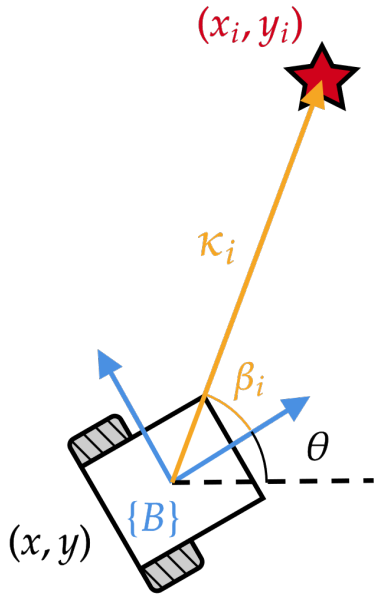

Figura 1: Mediciones provistas por el sensor de distancia omnidireccional.

se observa que el modelo para el sensor de distancia está dado por

$$\mathbf{y}_{i,k} = {}^{B}\begin{bmatrix}\kappa_i \\ \beta_i\end{bmatrix} = {}^{B}\boldsymbol{\mathcal{G}}\left({}^{N}\boldsymbol{\xi}_k, {}^{N}\mathbf{p}_i, \mathbf{v}_k\right) = \underbrace{\begin{bmatrix}\sqrt{(x_i-\xi_{1,k})^2+(y_i-\xi_{2,k})^2} \\ \operatorname{atan2}\left(\frac{y_i-\xi_{2,k}}{x_i-\xi_{1,k}}\right)-\xi_{3,k}\end{bmatrix}}_{\boldsymbol{\mathcal{H}}_i(\mathbf{x}_k)} + \underbrace{\begin{bmatrix}v_\kappa \\ v_\beta\end{bmatrix}}_{\mathbf{v}_k},$$

en donde $\mathbf{p}_i = \begin{bmatrix}x_i & y_i\end{bmatrix}^\top$ corresponde a la posición del i-ésimo landmark puntual detectado por el sensor y $v_\kappa \sim \mathcal{N}(0,\sigma_\kappa^2),\ v_\beta \sim \mathcal{N}(0,\sigma_\beta^2)$ corresponden al ruido de medición. Nótese que para cálculos analíticos puede cambiarse la función atan2 con un arctan para que la derivada se encuentre bien definida.

Si bien este sensor de distancia nos permitirá detectar obstáculos/landmarks a los cuales el robot tiene que llegar o esquivar, podemos emplear el hecho que conocemos perfectamente la pose del robot para extender las capacidades del sensor. Si conocemos la pose del robot, podemos emplear la función

$${}^{N}\mathbf{g}\left({}^{N}\boldsymbol{\xi}_k, \mathbf{y}_{i,k}\right) = \begin{bmatrix}\xi_{1,k}+\kappa_i\cos(\xi_{3,k}+\beta_i) \\ \xi_{2,k}+\kappa_i\sin(\xi_{3,k}+\beta_i)\end{bmatrix},$$

para encontrar un estimado de la posición de los obstáculos/landmarks que vayan siendo detectados por el robot empleando el sensor de distancia. De esta observación, resulta que podemos plantear el problema de construcción de mapas como un problema de estimación a resolver mediante un EKF. Para este caso, definiremos el vector de estado $\mathbf{x}_k = \begin{bmatrix}\mathbf{p}_1^\top & \mathbf{p}_2^\top & \cdots & \mathbf{p}_N^\top\end{bmatrix}$ como la colección de posiciones de todos los N obstáculos/landmarks presentes en el mapa. Ya que sabemos que la posición de estos puntos se mantiene fija, obtenemos el siguiente modelo de predicción trivial

$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1},$$

$$\mathbf{P}_{k|k-1} = \mathbf{P}_{k-1|k-1}.$$

Notemos, sin embargo, que como no conocemos el mapa a-priori sino que se va construyendo conforme avanza el tiempo, el vector $\hat{\mathbf{x}}_k$ inicia vacío y deben irse añadiendo las posiciones de los obstáculos/landmarks conforme van siendo detectados por el robot (también el robot debe ser capaz de discernir si las mediciones obtenidas por el sensor corresponden a un obstáculo/landmark que ya se observó previamente). Efectuaremos este proceso de ajuste del vector de estado, y por consiguiente de la matriz de covarianza del error de estimación, mediante las operaciones

$$\hat{\mathbf{x}}_{k|k}^{\text{new}} = \boldsymbol{\mathcal{Y}}\left(\hat{\mathbf{x}}_{k|k}, {}^{N}\boldsymbol{\xi}_k, \mathbf{y}_{i,k}\right) = \begin{bmatrix}\hat{\mathbf{x}}_{k|k} \\ {}^{N}\hat{\mathbf{p}}_i\end{bmatrix} = \begin{bmatrix}\hat{\mathbf{x}}_{k|k} \\ {}^{N}\mathbf{g}\left({}^{N}\boldsymbol{\xi}_k, \mathbf{y}_{i,k}\right)\end{bmatrix},$$

$$\mathbf{P}_{k|k}^{\text{new}} = \mathbf{Y}_{\mathbf{y}}\begin{bmatrix}\mathbf{P}_{k|k} & \mathbf{0}_{n\times 2} \\ \mathbf{0}_{2\times n} & \mathbf{Q}_{vg}\end{bmatrix}\mathbf{Y}_{\mathbf{y}}^\top,$$

con

$$\mathbf{Q}_{vg} = \begin{bmatrix}\sigma_x^2 & 0 \\ 0 & \sigma_y^2\end{bmatrix}, \qquad \mathbf{Y}_{\mathbf{y}} = \frac{\partial\boldsymbol{\mathcal{Y}}\left(\hat{\mathbf{x}}_{k|k}, {}^{N}\boldsymbol{\xi}_k, \mathbf{y}_{i,k}\right)}{\partial\mathbf{y}} = \begin{bmatrix}\mathbf{I}_{n\times n} & & \mathbf{0}_{n\times 2} \\ \mathbf{G}_{\boldsymbol{\xi}} & \mathbf{0}_{2\times n-3} & \mathbf{G}_{\mathbf{y}}\end{bmatrix} = \begin{bmatrix}\mathbf{I}_{n\times n} & \mathbf{0}_{n\times 2} \\ \mathbf{0}_{2\times n} & \mathbf{G}_{\mathbf{y}}\end{bmatrix},$$

$$\mathbf{G}_{\boldsymbol{\xi}} = \frac{\partial {}^{N}\mathbf{g}\left({}^{N}\boldsymbol{\xi}_k, \mathbf{y}_{i,k}\right)}{\partial {}^{N}\boldsymbol{\xi}} = \begin{bmatrix}0 & 0 & 0 \\ 0 & 0 & 0\end{bmatrix},$$

$$\mathbf{G}_{\mathbf{y}} = \frac{\partial {}^{N}\mathbf{g}\left({}^{N}\boldsymbol{\xi}_k, \mathbf{y}_{i,k}\right)}{\partial\mathbf{y}} = \begin{bmatrix}\cos(\xi_{3,k}+\beta_i) & -\kappa_i\sin(\xi_{3,k}+\beta_i) \\ \sin(\xi_{3,k}+\beta_i) & \kappa_i\cos(\xi_{3,k}+\beta_i)\end{bmatrix},$$

en donde σ_x^2, σ_y^2 corresponden a las varianzas del ruido blanco que corrompe a los estimados ${}^N\hat{\mathbf{p}}_i,\ i = 1, 2, ..., M$ de la posición de los obstáculos, n es el tamaño del estimado del vector de estado previo de su aumento de tamaño y la matriz $\mathbf{G}_{\boldsymbol{\xi}}$ es cero a causa del conocimiento perfecto de la pose del robot (lo cual cambiará cuando estudiemos el problema de localización y mapeo simultáneo). Para finalizar con el EKF, sólo resta definir la matriz $\mathbf{C}[k]$, la cual tomará la forma

$$\mathbf{C}[k] = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_M \end{bmatrix},$$

$$\mathbf{C}_i = \frac{\partial \boldsymbol{\mathcal{H}}_i\left(\hat{\mathbf{x}}_{k|k-1}\right)}{\partial \mathbf{p}_i} = \begin{bmatrix} \frac{x_i - \xi_{1,k}}{\sqrt{(x_i-\xi_{1,k})^2+(y_i-\xi_{2,k})^2}} & \frac{y_i - \xi_{2,k}}{\sqrt{(x_i-\xi_{1,k})^2+(y_i-\xi_{2,k})^2}} \\ -\frac{y_i - \xi_{2,k}}{(x_i-\xi_{1,k})^2+(y_i-\xi_{2,k})^2} & \frac{x_i - \xi_{1,k}}{(x_i-\xi_{1,k})^2+(y_i-\xi_{2,k})^2} \end{bmatrix}.$$

Con esto, el término $\mathbf{y}_k - \mathbf{C}[k]\hat{\mathbf{x}}_{k|k-1}$ en la etapa de corrección toma la forma

$$\begin{bmatrix} \mathbf{y}_{1,k} \\ \mathbf{y}_{2,k} \\ \vdots \\ \mathbf{y}_{N,k} \end{bmatrix} - \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots \vdots & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_N \end{bmatrix} \hat{\mathbf{x}}_{k|k-1}$$

y se define completamente el algoritmo de construcción de mapas. Este término puede calcularse también elemento por elemento mediante la expresión

$$\mathbf{y}_{i,k} = \underbrace{\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{C}_i & \cdots & \mathbf{0} \end{bmatrix}}_{\in \mathbb{R}^{2\times 2N}} \hat{\mathbf{x}}_{k|k-1}.$$

Esta estructura indica que la observación de uno de los obstáculos/landmarks sólo provee información para estimarse a sí mismo, pero no a otros. Nótese que las cantidades $\mathbf{y}_{i,k}, \boldsymbol{\mathcal{H}}_i\left(\hat{\mathbf{x}}_{k|k-1}\right) \equiv \boldsymbol{\mathcal{H}}_i\left(\hat{\mathbf{p}}_{i,k|k-1}\right)$ y $\mathbf{C}_i$ existen sólo si el i-ésimo obstáculo se encuentra dentro del rango de detección del sensor de distancia. Por lo tanto, para obstáculos fuera del rango del sensor deben hacerse $\mathbf{C}_i = \mathbf{0}$ y $\mathbf{y}_{i,k} - \boldsymbol{\mathcal{H}}_i\left(\hat{\mathbf{x}}_{k|k-1}\right) = \mathbf{0}$ para evitar que la etapa de corrección los tome en consideración.

Procedimiento

En la práctica de esta semana usted empleará MATLAB para implementar un filtro de Kalman extendido que permita construir un mapa de un entorno desconocido que contiene obstáculos/landmarks puntuales, asumiendo que tiene un conocimiento perfecto de la pose del roboto. Para lograr esto, realice lo siguiente:

1. Descargue de Canvas el script base `lab8.m` (que se encuentra dentro del archivo `mt3006lab8.zip`) y ábralo en MATLAB. Este script es idéntico al del Laboratorio 7 con la diferencia que se ha equipado al robot Pioneer3-DX con un LiDAR con un rango de medición máximo de 2m, que produce los vectores `range, bearing` que contienen las distancias κ y los ángulos β de todos los obstáculos/landmarks detectados dentro del rango de medición del sensor. Adicionalmente, se le da acceso a las mediciones reales de posición y orientación `x,y,theta` del robot. Corra el script con los parámetros por defecto y observe la forma que presenta tanto la trayectoria del robot como la detección de obstáculos mediante el sensor.

2. Emplee el comando `spline`, junto con una serie de puntos de su elección, para diseñar una trayectoria cerrada (es decir, que termine en donde inició) que pase observando todos los obstáculos/landmarks por lo menos una vez sin colisionar con ellos.

3. Implemente un controlador (de su elección, según lo que mejor le haya funcionado en laboratorios y proyectos previos) que le permita al robot rastrear la trayectoria del inciso anterior lo suficientemente rápido para evitar colisiones con los obstáculos/landmarks.

4. Utilice las mediciones provistas `range, bearing` por el sensor de distancia, junto con los comandos `mean` y `std`, para estimar la varianza/desviación estándar de los errores de medición v_κ, v_β y de proceso w_x, w_y, asumiendo que todos corresponden a ruido blanco no correlacionado.

5. Con base en las varianzas estimadas y los modelos de predicción y corrección presentados en los antecedentes, implemente el filtro de Kalman que construya el mapa (emplee el array `map` para esto) del entorno desconocido conforme el robot recorre repetidamente la trayectoria diseñada. Genere el mapa que construyó el robot mediante un `scatter` plot del vector `map`. Tome en consideración que el mapa contiene únicamente 10 obstáculos/landmarks, por lo que su código debe ser capaz de asociar las mediciones ruidosas del sensor de distancia a los obstáculos/landmarks ya detectados, para evitar la repetición de los mismos.

6. Luego de haber implementado correctamente el filtro de Kalman, muestre al catedrático de laboratorio las figuras que genera el código por defecto y una adicional en donde se despliegue el mapa construido (el mapa también puede sobreponerse en la figura del entorno que genera el código). Recuerde que entregas tardías representan una penalización del 25 % por semana.