Localización y Mapeado

Iñaki Rañó

Electrónica y Computación, USC

Tabla de Contenidos

Bibliografía

- S. Thrun, W. Burgard & D. Fox. Probabilistic robotics, MIT Press, 2005. Chapters 2, 3, 4 & 10.
- H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki & S. Thrun. Principles of Robot Motion. Theory, Algorithms and Implementations. MIT Press, 2005. Chapter 8 & 9.

Introducción

El filtro de Kalman permite estimar el estado de un robot móvil para, p.e. localización:

- Modelo de movimiento del robot (ecuación de transición de estado) con ruido de estado (errores en el movimiento)
- Modelo de lectura de sensores (ecuación de medida) con ruido de medida
- Para localización el modelo de medida incluye información del entorno (mapa)
- Suposición de correspondencias percepción-mapa conocidas
- Inicializar filtro de Kalman (*EKF*)
  - Media estado: Lo mejor que se pueda
  - Desviación estándar (incertidumbre): Tan grande como sea necesario
- Filtro de Kalman (EKF) converge a la posición del robot con incertidumbre pequeña.

Introducción a la Asociación de Datos

Hasta ahora hemos supuesto que el robot conoce la asociación de datos correcta entre las lecturas sensoriales y las *landmarks* del mapa.

El mapa está formado por una serie de posiciones (*landmarks*) que el robot puede identificar de manera única.

Si el robot observa (mide) la posición de una *landmark* sabe cual es la posición de dicha *landmark* en el mapa, pero ¿qué son estas *landmarks*?

- Puntos del Lidar (¿Cómo diferenciarlos?)
- Segmentos lineales de tamaño conocido
  - Errores en longitud
  - ¿Qué punto del segmento se escoge?
- Esquinas, i.e. intersección entre segmentos a 90°

A veces este problema se ignora en localización y se trabaja con *landmarks* "abstractas"

Landmarks y sus Signaturas

Landmarks son objetos distintivos del entorno (e.g. puertas, ventanas,. . .) que se usan para la navegación de un robot y pueden ser detectados por algún sensor (e.g. ángulo y distancia).

Para cada *landmark* es interesante disponer también de una *signatura*, i.e. un valor numérico o vectorial asociado a la *landmark* que caracteriza de forma única a ésta.

Caracterización de *landmarks*

- Posición $\mathbf{p}$
  - Posición absoluta en el mapa $\mathbf{p} = [x, y]$
  - Posición relativa respecto al robot $\mathbf{p} = [x, y]$ o $\mathbf{p} = [r, \phi]$
- Signatura $\mathbf{s}$, escalar o vector que identifica de manera única la *landmark*

Si no se dispone de signatura se puede usar la posición para identificar las *landmarks*.

Función de Asociación

Para la localización hemos supuesto que las *landmarks* tienen un identificador único que permite construir la función de asociación.

- El mapa tiene n_l *landmarks* en $\mathbf{p}_{li}$ $i = 1, 2, \cdots, n_l$ con una signatura s única $s_i \in \{1, 2, \cdots, n_l\}$
- El robot observa $p(k)$ *landmarks* en $\hat{\mathbf{p}}_j$ $j = 1, 2, \cdots, p(k)$ y es capaz de identificar su signatura $s_j \in \{1, 2, \cdots, n_l\}$

Función de asociación $a : \{1, 2, \cdots, p(k)\} \to \{1, 2, \cdots, n_l\}$ asigna a cada observación una *landmark* del mapa.

$$a(j) = \underset{i \in \{1, \cdots, n_l\}}{\arg\min} |s_j - s_i|$$

Es decir, la *landmark* observada $(\hat{\mathbf{p}}_j, s_j)$ se asocia con la *landmark* del mapa $(\mathbf{p}_{li}, s_i)$ si $s_j = s_i$ (identificador único)

Función de Asociación General

Si las signaturas son vectores con ciertas características $\mathbf{s}_i$, la función de asociación puede ser la misma

- El mapa tiene n_l *landmarks* en $\mathbf{p}_{li}$ $i = 1, 2, \cdots, n_l$ con una signatura $\mathbf{s}_i \in \{1, 2, \cdots, n_l\}$
- El robot observa $p(k)$ *landmarks* en $\hat{\mathbf{p}}_j$ $j = 1, 2, \cdots, p(k)$ e identifica su signatura $\hat{\mathbf{s}}_j \in \{1, 2, \cdots, n_l\}$

Función de asociación:

$$a(j) = \arg\min_{i \in \{1, \cdots, n_l\}} |\hat{\mathbf{s}}_j - \mathbf{s}_i|$$

- La signatura observada ($\hat{\mathbf{s}}$) no tiene por qué ser idéntica a la del mapa ($\mathbf{s}$)
- Ejemplo de signaturas: color RGB, longitud y ángulo de un segmento, ángulo de una esquina,...

Función de Asociación sin Signaturas

En ocasiones no es trivial encontrar signaturas para *landmarks* y solo se dispone de la posición de las *landmarks* en el mapa ($\mathbf{p}_{li}$) o medidas por el robot ($\hat{\mathbf{p}}_j$).

- Las posiciones de las *landmarks* en el mapa ($\mathbf{p}_{li}$) están referidas al sistema de referencia del mundo.
- Las posiciones de las *landmarks* detectadas por el robot ($\hat{\mathbf{p}}_j$) están referidas al sistema de referencia del robot.
- Dada la posición (y orientación) del robot $\mathbf{x}$ (R) podemos convertir las posiciones de las *landmarks* detectadas al sistema de referencia del mundo

$$^{W}\hat{\mathbf{p}}_j = \mathbf{x} + R\hat{\mathbf{p}}_j$$

- Función de asociación:

$$a(j) = \underset{i \in \{1, \cdots, n_l\}}{\arg\min} \left| {}^{W}\hat{\mathbf{p}}_j - \mathbf{p}_{li} \right|$$

Asociar j con la *landmark* más próxima en el mapa.

Función de Asociación sin Signaturas (ii)

El problema es que no conocemos la posición (ni la orientación) del robot $\mathbf{x}$ (R), ya que estamos intentando localizar al robot (i.e. estimar su estado)

- Tenemos una estimación de la posición (pose) del robot $\hat{\mathbf{x}}_{k|k}$ en cada instante de tiempo
- La función de asociación se necesita para la actualización (*update*), i.e. $\hat{\mathbf{x}}_{k|k}$ no está actualizada
- Tras la predicción tenemos cierta idea del estado (posición o pose) del robot $\hat{\mathbf{x}}_{k+1|k}$ al ejecutar la acción $\mathbf{u}_k$.

Por tanto para la función de asociación se puede utilizar la estimación *a priori* del estado $\hat{\mathbf{x}}_{k+1|k}$ (error de innovación)

$$a(j) = \underset{i \in \{1, \cdots, n_l\}}{\arg\min} \left| {}^W\hat{\mathbf{p}}_j - \mathbf{p}_{li} \right|$$

donde

$${}^W\hat{\mathbf{p}}_j = \hat{\mathbf{x}}_{k+1|k} + \hat{R}\hat{\mathbf{p}}_j$$

Función de Asociación sin Signaturas (iii)

La definición anterior de la función de asociación se basa en minimizar el error de innovación del filtro de Kalman:

$$\tilde{\mathbf{y}}_{k+1} = \mathbf{p}_{li} - \left[\hat{\mathbf{x}}_{k+1|k} + \hat{R}\hat{\mathbf{p}}_j\right]$$

pero no considera considera la incertidumbre en la posición del robot ni en la medida (covarianza de la innovación).

El error de innovación se puede pesar usando la matriz de covarianza de la innovación S_{k+1}, que considera tanto la incertidumbre en la posición del robot como en la medida.

$$\chi^2 = \tilde{\mathbf{y}}_{k+1}^T S_{k+1}^{-1} \tilde{\mathbf{y}}_{k+1}$$

Distancia de Mahalanobis: Distancia de un punto a una distribución normal: $d(\mathbf{p}, \mathcal{N}(\mu, \Sigma)) = \sqrt{(\mathbf{p} - \mu)^T \Sigma^{-1} (\mathbf{p} - \mu)}$

Distancia de Mahalanobis y χ^2

Esta forma de asociación se puede ver como:

- Distancia de Mahalanobis al cuadrado
- Distribución de probabilidad χ^2 (suma de variables aleatorias Gaussianas $\mathcal{N}(0,1)$)

Si M es una matriz simétrica ($M = M^T$) y definida positiva ($\mathbf{x}^T M \mathbf{x} > 0$) su descomposición en valores singulares $M = UDV^T$ cumple que $U = V$ (matriz de rotación).

Supongamos $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, por lo que $\Sigma = RDR^T = RD^{\frac{1}{2}}D^{\frac{1}{2}}R^T$, y definimos $\mathbf{y} = D^{-\frac{1}{2}}R^T[\mathbf{x} - \boldsymbol{\mu}]$, entonces $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, I)$, i.e.

$$\begin{aligned}
(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}) &= (\mathbf{x}-\boldsymbol{\mu})^T[RD^{\frac{1}{2}}D^{\frac{1}{2}}R^T]^{-1}(\mathbf{x}-\boldsymbol{\mu}) \\
&= (\mathbf{x}-\boldsymbol{\mu})^T R^{-T} D^{-\frac{1}{2}} D^{-\frac{1}{2}} R^{-1}(\mathbf{x}-\boldsymbol{\mu}) \\
&= [D^{-\frac{1}{2}}R^T(\mathbf{x}-\boldsymbol{\mu})]^T I [D^{-\frac{1}{2}}R^T(\mathbf{x}-\boldsymbol{\mu})] = \mathbf{y}^T\mathbf{y}
\end{aligned}$$

Detección de *Outliers*

Para construir la función de asociación puede usarse la distancia de Mahalanobis con la covarianza de la innovación:

$$d^2 = [\mathbf{y} - C\hat{\mathbf{x}}]^T S^{-1} [\mathbf{y} - C\hat{\mathbf{x}}]$$

Con lo que siempre se asocia una observación a la *landmark* del entorno más próxima.

Si la medida/observación **y** es un *outlier* el filtro de Kalman probablemente divergirá, pero como d^2 sigue una distribución χ^2 puede estimarse la probabilidad de que **y** sea un *inlier* (en función de la dimension de **y**, percentiles)

dim.	95 %	99 %	99,5 %
2	5,99	9,21	10,6
3	7,81	11,34	12,84

i.e. $d^2 > 9{,}21$ la observación (2D) es probablemente un *outlier*

Función de Asociación

Dado un mapa con n_l *landmarks* sin signatura en $\mathbf{p}_{li}$ $i = 1, 2, \cdots, n_l$, un conjunto de $p(k)$ observaciones $\hat{\mathbf{p}}_j$ $j = 1, 2, \cdots, p(k)$ y la estimación a priori de la pose del robot $\hat{\mathbf{x}}_{k+1|k}$, la función de asociación es:

$$a(j) = \underset{i\in\{1,\cdots,n_l\}}{\arg\min} \left[(\mathbf{p}_{li} - (\hat{\mathbf{x}}_{k+1|k} + \hat{R}\hat{\mathbf{p}}_j))^T S_{k+1}^{-1} (\mathbf{p}_{li} - (\hat{\mathbf{x}}_{k+1|k} + \hat{R}\hat{\mathbf{p}}_j))\right]$$

- Distancia de Mahalanobis considera error de medida e incertidumbre de la estimación ($S_{k+1} = CP_{k+1|k}C^T + R$)
- Se usa la estimación *a priori* del estado (pose) $\hat{\mathbf{x}}_{k+1|k}$ para calcular la posición de la *landmark* observada en el mapa
- Siempre se asocia una observación al mapa (¿Qué pasa si es un *outlier*?)

Introducción al Problema de *SLAM*

- Significa *Simultaneous Localisation and Mapping* (Localización y mapeado simultaneo)
- Problema de construir o actualizar un mapa mientras se mantiene la localización del robot en él
- Mapas del entorno pueden ser incorrectos, desactualiados, no existir o costosos
- El robot tiene que localizarse en el mapa: necesita conocer la posición de las *landmarks* del mapa para estimar su pose
- El robot tiene que construir un mapa: necesita conocer su posición (pose) para poder convertir las posiciones de las *landmarks* que detecta al sistema de referencia del mapa
- Problema de tipo 'el huevo y la gallina' (como *ICP*)
- Puede resolverse con el filtro de Kalman o el filtro de Kalman extendido (*EKF*)

SLAM Simple: Suposiciones

- Robot tipo integrador con estado $\mathbf{x} = [x, y]$, acciones $\mathbf{v} = [v_x, v_y]$ y ruido de estado $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, Q)$
- En el entorno hay n_l *landmarks* que el robot puede identificar de forma única (signatura) pero no conoce sus posiciones $\mathbf{x}_{l_i}$ para $i = 1, 2, \cdots, n_l$
- El robot puede observar la posición Cartesiana de las *landmarks* relativas a su posición, i.e. $\mathbf{x}_{l_i} - \mathbf{x}$, con cierto ruido de medida $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, R)$
- El robot puede observar todas las *landmarks* en cada instante de tiempo

Estas suposiciones son bastante duras pero permiten ilustrar el problema y puede resolverse de manera simple (sistema lineal).

SLAM Simple: Estado

En el caso del *SLAM* queremos estimar la posicion (pose) del robot, pero también la de las *landmarks* del entorno.

- El estado es el vector que contiene la posición del robot y la de todas las *landmarks* del entorno:

  $$\mathbf{X}^T = [\mathbf{x}, \mathbf{x}_{l1}, \mathbf{x}_{l2}, \cdots, \mathbf{x}_{ln_l}]$$

  dimensión $2(n_l + 1)$
- Conocemos la posición inicial del robot $\mathbf{x} = \mathbf{0}$, si no es el origen del mundo ($\mathbf{x}_0$) solo hay que trasladar el estado $\mathbf{x} \rightarrow \mathbf{x} + \mathbf{x}_0$, y las *landmarks* $\mathbf{x}_{li} \rightarrow \mathbf{x}_{li} + \mathbf{x}_0$
- Incertidumbre en la posición inicial del robot nula $P_{0|0} = 0$ (sabemos donde empieza el robot)
- Incertidumbre en la posición inicial de las *landmarks* infinita $P_0 = \infty \cdot I_{2\times 2}$ (o medir primero y $P_0 = R$)

SLAM Simple: Transición de Estado

- Matriz de transición de estado $I_{2(n_l+1)\times 2(n_l+1)}$ (matriz indentidad)
- Las acciones ($\mathbf{v} = [v_x, v_y]$) afectan solo a la posición del robot (las *landmarks* no se mueven), i.e. la matriz B ($2(n_l+1) \times 2$) tiene la identidad 2×2 seguida de $2n_l$ filas de ceros
- El ruido de estado ($\mathbf{w}$) afecta solo al movimiento del robot y su covarianza es la matriz cero excepto un bloque 2×2 que contiene Q

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \begin{bmatrix} T & 0 \\ 0 & T \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

SLAM Simple: Ecuación de Medida

- Podemos medir la posición de las *landmarks* pero no la del robot (solo la estimamos)
- La medida para una *landmark* individual es la posición relativa respecto al robot, para la *landmark* $i = 1, 2, \cdots, n_l$ las filas $2i-1$ y $2i$ de la matriz C serán

$$C_i = \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

- El ruido de medida ($\mathbf{v} = [v_x, v_y]$) se aplica a cada *landmark* de forma individual

$$\hat{\mathbf{y}}_k = \begin{bmatrix} -1 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -1 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & -1 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \mathbf{X}_k + \begin{bmatrix} v_{x_1} \\ v_{y_1} \\ \vdots \\ v_{x_{n_l}} \\ v_{y_{n_l}} \end{bmatrix}$$

SLAM Simple: Filtro de Kalman

- A diferencia del proceso de localización, las ecuaciones para *SLAM* son lineales (transición de estado y medida), i.e. se puede usar el filtro de Kalman
- Estimación inicial de la posición de las *landmarks* aleatoria, covarianza infinita (grande)
- Estimación inicial de la posicion (pose) del robot $\hat{\mathbf{x}}_{0|0} = \mathbf{0}$, covarianza cero
- La covarianza del estado $\hat{P}_{k|k}$ (inicialmente diagonal) termina incluyendo covarianzas cruzadas entre *landmarks* distintas y entre las *landmarks* y el robot
- La incertidumbre en la posición del robot aumenta (en la predicción) pero llega a un límite (relacionado con la covarianza de la medida)

Range & Bearing SLAM: Suposiciones

- Robot tipo uniciclo con estado $\mathbf{x} = [x, y, \theta]$, acciones $\mathbf{v} = [v, \omega]$ y ruido de estado $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, Q)$
- Hay n_l *landmarks* que el robot puede identificar de forma única pero no conoce sus posiciones $\mathbf{x}_{li}$ para $i = 1, 2, \cdots, n_l$
- El robot puede observar la distancia y ángulo a todas las *landmarks* con ruido de estado $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, R)$

Suposiciones:

- Número de *landmarks* conocidas *a priori*: Simplifica la implementación
- Función de asociación perfecta: Asegura convergencia y simplifica la implementación
- Rango infinito: simplifica la implementación y elimina el problema de cerrado de bucle

Range & Bearing SLAM: Estado

Igual que en el caso anterior el estado incluye el estado del robot y las posiciones de todas las *landmarks*

- Dimensión del vector de estado $2n_l + 3$

$$\mathbf{X}^T = [x, y, \theta, \mathbf{x}_{l1}, \cdots, \mathbf{x}_{ln_l}]$$

- Pose inicial del robot p.e. $\mathbf{x}_0 = [0, 0, 0]$ con incertidumbre $P_r = 0$
- Posición inicial de las *landmarks* p.e. aleatoria con incertidumbre infinita $P_{li} = \infty I_{2\times 2}$

Si las *landmarks* se van encontrando (rango sensorial finito) posición inicial como posición medida e incertidumbre infinita (*upcoming*)

Range & Bearing SLAM: Transición de Estado

- Transición de estado no lineal
- Las acciones ($\mathbf{v} = [v, \omega]$) afectan solo a la posición del robot
- El ruido de estado ($\mathbf{w}$) afecta solo al movimiento del robot no a las *landmarks*

$$\mathbf{X}_{k+1} = \mathbf{X}_k + T \begin{bmatrix} \cos\theta_k & 0 \\ \sin\theta_k & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_\theta \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Range & Bearing SLAM: Ecuación de Medida

- Podemos medir la posición de las *landmarks* pero no la del robot
- Para una *landmark* $i = 1, 2, \cdots, n_l$ en el mapa, la medida es:

$$\hat{\mathbf{y}}_i = \begin{bmatrix} \sqrt{[x_{li} - x]^2 + [y_{li} - y]^2} \\ \arctan\left[\frac{y_{li}-y}{x_{li}-x}\right] - \theta \end{bmatrix}$$

- El ruido de medida se aplica a cada *landmark* individual
- La dimensión del vector de salida es siempre $2n_l$ porque el rango sensorial es infinito

Range & Bearing SLAM: Jacobianos

Como la ecuación de transición de estados es no lineal necesitamos calcular su Jacobiano para la predicción de la covarianza del estado:

$$J_{\mathbf{X}}\mathbf{F} = \begin{bmatrix} 1 & 0 & -v\sin\theta & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & v\cos\theta & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

- El estado ($\mathbf{X}$) en SLAM incluye las posiciones de las *landmarks*
- Solo cambia la pose del robot y no depende de las posiciones de las *landmarks*
- *Sparse matrix*; matriz con pocos elementos distintos de cero

Como la ecuación de medida es no lineal necesitamos calcular su Jacobiano para la actualización de la covarianza del estado (cálculo de S, covarianza de la innovación, y K, ganancia de Kalman):

$$J_{\mathbf{X}}\mathbf{G} = \begin{bmatrix} \frac{x-x_{l1}}{r_1} & \frac{y-y_{l1}}{r_1} & 0 & \frac{x_{l1}-x}{r_1} & \frac{y_{l1}-y}{r_1} & 0 & 0 & \cdots & 0 & 0 \\ \frac{y_{l1}-y}{r_1^2} & -\frac{x_{l1}-x}{r_1^2} & -1 & -\frac{y_{l1}-y}{r_1^2} & \frac{x_{l1}-x}{r_1^2} & 0 & 0 & \cdots & 0 & 0 \\ \frac{x-x_{l2}}{r_2} & \frac{y-y_{l2}}{r_2} & 0 & 0 & 0 & \frac{x_{l2}-x}{r_2} & \frac{y_{l2}-y}{r_2} & \cdots & 0 & 0 \\ \frac{y_{l2}-y}{r_2^2} & -\frac{x_{l2}-x}{r_2^2} & -1 & 0 & 0 & -\frac{y_{l2}-y}{r_2^2} & \frac{x_{l2}-x}{r_2^2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{x-x_{ln_l}}{r_{n_l}} & \frac{y-y_{ln_l}}{r_{n_l}} & 0 & 0 & 0 & 0 & 0 & \cdots & \frac{x_{ln_l}-x}{r_{n_l}} & \frac{y_{ln_l}-y}{r_{n_l}} \\ \frac{y_{ln_l}-y}{r_{n_l}^2} & -\frac{x_{ln_l}-x}{r_{n_l}^2} & -1 & 0 & 0 & 0 & 0 & \cdots & -\frac{y_{ln_l}-y}{r_{n_l}^2} & \frac{x_{ln_l}-x}{r_{n_l}^2} \end{bmatrix}$$

donde $r_i = \sqrt{[x_{li} - x]^2 + [y_{li} - y]^2}$

Range & Bearing SLAM

Con esta formulación podemos estimar las posiciones de las *landmarks* en el mapa y la posición del robot usando el filtro de Kalman extendido.

- Se puede adaptar a otros modelos de movimiento y medidas, pero uniciclo y *range & bearing* es lo más típico
- Explotar el hecho de que las matrices son *sparse* para optimización de la implementación
- Dimensiones de matrices y vectores:
  - Estado $\mathbf{X}$: Dimensión $3 + 2n_l$
  - Covarianza del estado: $(3 + 2n_l) \times (3 + 2n_l)$
  - Observación ($\mathbf{y}$) y predicción ($\hat{\mathbf{y}}$): Dimensión $2n_l$
  - Jacobiano de la transición de estados: $(3 + 2n_l) \times (3 + 2n_l)$
  - Jacobiano de la medida: $2n_l \times (3 + 2n_l)$

SLAM Real: Percepción Limitada

En realidad un robot siempre tiene un rango de percepción limitado (rango, oclusiones...)

- Suponemos número de *landmarks* en el entorno conocido a priori (n_l)
- La dimensión del vector de estado $\mathbf{X}$ es conocida y fija
- En cada instante el robot solo ve $p(k)$ *landmarks*
- Los vectores de observación ($\mathbf{y}_k$) tiene dimensión $2p(k)$
- El vector de predicción de la observación ($\hat{\mathbf{y}}_k$) tiene que tener la misma dimensión (y sus componentes ser correspondientes)
- Mientras el robot vea *landmarks* la covarianza de su pose estará acotada
- Cuando el robot deja de ver *landmarks* la covarianza de su pose aumenta

SLAM Real: Número de *Landmarks*

En realidad un robot no tiene por qué conocer el número de *landmarks* (n_l) en un entorno *a priori*

- Rango de percepción limitado, no puede ver todas de golpe
- Función de asociación conocida, i.e. el robot sabe a qué componentes del estado $\mathbf{X}$ corresponde cada observación
- La dimensión del vector de estado $\mathbf{X}$ cambia según va encontrando *landmarks*, i.e. se extiende al encontrar una nueva *landmark* ($n_l(k)$ *landmarks* vistas hasta k)
$$\mathbf{X} \rightarrow [\mathbf{X}, \mathbf{x}_l]$$
- La matriz de covarianza del estado se extiende al encontrar una nueva *landmark*
$$P \rightarrow \begin{bmatrix} P & 0 \\ 0 & P_l \end{bmatrix}$$
donde $P_l = \infty I_{2\times 2}$ (incertidumbre de la nueva *landmark*) u otra alternativa mejor.

SLAM Real: Función de Asociación

Tanto en localización como en *SLAM* los robots reales deben construir su función de asociación de datos.

- Con signaturas apropiadas se usa la signatura para las correspondencias
- *Landmarks* sin signaturas, se utiliza la posición y la covarianza de la información para calcular la distancia de Mahalanobis (asociación por distancia mínima)
- Para detectar *outliers* se usa el test de χ^2, i.e. por ejemplo para una distancia de Mahalanobis mínima d_{min}, si $d_{min} > 9{,}21$ *outlier* (2D)
- Test de χ^2 se usa también para detectar *landmarks* nuevas, i.e. por ejemplo para una distancia de Mahalanobis mínima d_{min}, si $d_{min} > 9{,}21$ nueva *landmark* (2D)

Algoritmo general de *EKF-SLAM*

$\mathbf{X}_{0|0} \leftarrow \mathbf{0}$ ▷ Initial pose
$P_{0|0} \leftarrow 0_{3\times3}$ ▷ Initial covariance
$[\mathbf{y}_i, R_i] \leftarrow \mathit{measure}()$ ▷ Landmarks $i = 1, \cdots, p(0)$
$[\mathbf{X}_{0|0}, P_{0|0}] \leftarrow \mathit{add\_landmarks}(\mathbf{X}_{0|0}, P_{0|0}, \mathbf{y}_i, R_i)$
for $k = 0, 1, 2, \cdots$ **do**
    $[\mathbf{v}_k, Q] \leftarrow \mathit{move}()$
    $[\mathbf{X}_{k+1|k}, P_{k+1|k}] \leftarrow \mathit{predict}(\mathbf{X}_{k|k}, P_{k|k}, \mathbf{v}_k, Q)$
    $[\mathbf{y}_i, R_i] \leftarrow \mathit{measure}()$ ▷ Landmarks $i = 1, \cdots, p(k)$
    $a \leftarrow \mathit{association}(\mathbf{X}_{k+1|k}, P_{k+1|k}, \mathbf{y}_i, R_i)$
    $[\mathbf{X}_{k+1|k+1}, P_{k+1|k+1}] \leftarrow \mathit{update}(\mathbf{X}_{k+1|k}, P_{k+1|k}, \mathbf{y}_i, R_i, a)$
    $[\mathbf{X}_{0|0}, P_{0|0}] \leftarrow \mathit{add\_landmarks}(\mathbf{X}_{0|0}, P_{0|0}, \mathbf{y}_i, R_i)$
end for

- Inicializar estado con la pose y una primera lectura de los sensores
- Función de asociación de datos $a : \{1, \cdots, p(k)\} \rightarrow \{1, \cdots, n_l(k)\}$ con $n_l(k)$ número de *landmarks* observadas hasta ahora

Algoritmo de Asociación de Datos

La asociación de datos (medidas y estado de las *landmarks* en el estado) es crítica para la convergencia del *EKF-SLAM*.

De las ecuaciones de actualización del filtro de Kalman:

$$\mathbf{X}_{k+1|k+1} = \mathbf{X}_{k+1|k} + K\tilde{\mathbf{y}}_k$$
$$P_{k+1|k+1} = P_{k+1|k+1} - KS_kK^T$$

La función de asociación afecta a $\tilde{\mathbf{y}}_k$, S_k y $K = P_{k+1|k}[J\mathbf{G}_k]^T S_k^{-1}$.

- Función de asociación correcta:
  - La media del estado estimado se acerca a la media real
  - La covarianza estimada se reduce
  - El algoritmo converge
- Función de asociación incorrecta:
  - La media del estado estimado se aleja de la media real
  - La covarianza estimada se reduce
  - El algoritmo diverge

Algoritmo de Asociación de Datos (ii)

Algoritmo del vecino más próximo (algoritmo voraz)

Input: Measures $(\mathbf{y}_i, R)$ $(i = 1, \cdots, p(k))$, Predictions $(\hat{\mathbf{y}}_j, P_j$ $(j = 1, \cdots, n_l(k))$,
Output: Function $a : \{1, \cdots, p(k)\} \rightarrow \{0, 1, \cdots, n_l(k)\}$
for $i = 1, 2, \cdots, p(k)$ **do**
  $d_m^2 \leftarrow [\mathbf{y}_i - \hat{\mathbf{y}}_1]^T \left[[J\mathbf{G}_k]P_j[J\mathbf{G}_k]^T + R\right]^{-1} [\mathbf{y}_i - \hat{\mathbf{y}}_1]$
  $j_m \leftarrow 1$
  for $j = 2, \cdots, n_l(k)$ **do**
    $d^2 \rightarrow [\mathbf{y}_i - \hat{\mathbf{y}}_1]^T \left[[J\mathbf{G}_k]P_j[J\mathbf{G}_k]^T + R\right]^{-1} [\mathbf{y}_i - \hat{\mathbf{y}}_1]$
    if $d^2 < d_m^2$ **then**
      $d_m^2 \leftarrow d^2$
      $j_m \leftarrow j$
    end if
  end for
  if $d_m^2 < 9{,}21$ **then** ▷ 99 % 2D χ^2 quantile
    $a(i) \leftarrow j_m$
  else
    $a(i) \leftarrow 0$ ▷ No association
  end if
end for
return a

Algoritmo de Asociación de Datos (iii)

- Hemos extendido la función de asociación para permitir que observaciones/medidas no se asocien con ninguna *landmark* en el estado
- Resultado de la asociación podría cambiar si se recorren las observaciones en otro orden (primer bucle)
- Habría que buscar **todas** las posibles asociaciones y entontrar la correcta (¿cómo?)
- Hay distintos algoritmos de asociación de datos en *SLAM* mejores que el vecino más próximo
  - *Sequential Compatibility Nearest Neigbour*
  - *Joint Compatibility Branch and Bound*: explora el árbol de interpretación y toma la función con mayor número de asociaciones
  - *Multiple Hypothesis Joint Compatibility Branch and Bound*

Añadir Nueva Landmark

- Cuando una observación/medida no se puede asociar con una *landmark* existente en el estado hay que añadirla al estado
- Dada la observación $\mathbf{y}_j$ con covarianza de medida R
- Dadas las landmarks almacenadas en el estado $\hat{\mathbf{y}}_i = \mathbf{G}_i(\mathbf{X})$ con covarianzas $P_i = [J_{\mathbf{X}}\mathbf{G}_i]P_{k|k}[J_{\mathbf{X}}\mathbf{G}_i]^T$
- Si $d_{ji}^2 = [\mathbf{y}_j - \hat{\mathbf{y}}_i]^T\,[P_i + R]^{-1}\,[\mathbf{y}_j - \hat{\mathbf{y}}_i] > 9{,}21$ para todo i, entonces la observación corresponde con una nueva *landmark*
- Extender el estado $\mathbf{X}$ con la posición de la nueva *landmark* (si es posible $\mathbf{x}_l = \mathbf{G}^{-1}(\mathbf{y}_j)$, i.e. $\mathbf{X} = [\mathbf{X}, \mathbf{x}_l]$)
- Extender la matriz de covarianza:
  - Covarianza del robot e inversa de la medida decorrelada con el resto del estado $P_l = P_r + [J\mathbf{G}^{-1}]R[J\mathbf{G}^{-1}]T$
  - Definir $\mathbf{Y} = [\mathbf{X}, \mathbf{y}]$, $P_{\mathbf{Y}} = diag(P_{k|k}, R)$, $\mathbf{X}^{new} = \mathcal{F}(\mathbf{Y})$ y hacer $P_{k|k} = [J\mathcal{F}]P_{\mathbf{Y}}[J\mathcal{F}]^T$ (covarianza correlada con todo el estado, necesita también $\mathbf{G}^{-1}(\mathbf{y})$)

Cerrado de Bucles (*Loop Closure*)

- A medida que el robot explora el entorno (se mueve) su incertidumbre de posición aumenta (covarianza P_r de la pose, sub-matriz de $P_{k|k}$)
- Nuevas *landmarks* encontradas se ven afectadas por la incertidumbre del robot, i.e. covarianza nueva *landmark* P_l relacionada con la suma de P_r y R
- Cuanto mayor la distancia recorrida mayor incertidumbre en las *landmarks*
- Al volver a observar una *landmark* la incertidumbre puede ser alta. Si. . .
  - La observación se asocia mal, el robot cree que la *landmark* es nueva y la añade al estado
  - La observación se asocia bien, (*loop closure*) se reduce la incertidumbre de todas las landmarks (últimas visitadas) y de la posición del robot

Limitaciones del Filtro de Kalman

Tanto para localización como para *SLAM* el filtro de Kalman y el *EKF* funcionan más o menos bien, pero

- En algunos casos necesitan una buena estimación del estado inicial (media), i.e. problema de inicialización
- Si la asociación de datos es incorrecta pueden diverger (y no recuperarse)
- Solo hay una *hipótesis* del estado (probabilidad Gaussiana), lo cual puede ser un problema, e.g. para localización, en entornos con zonas similares (e.g. entornos de oficinas)
- Las probabilidades no son en general Gaussianas (no linealidad)

¿Hay alguna otra forma de hacer estimación? $\rightarrow$ Filtros de partículas

Filtro de Partículas

- Es un método de estimación recursiva Bayesiana no paramétrico
- Tiene dos etapas: predicción y actualización
- Funciona en sistema no lineales (para la transición de estados y/o para la medida)
- Representa las distribuciones de probabilidad a través de muestras aleatorias ($\mathbf{x}_i$ llamadas partículas) con pesos asociados (w_i factor de importancia)

$$p(\mathbf{x}) = \sum_{i=1}^{n_s} w_i \delta(\mathbf{x} - \mathbf{x}_i)$$

  donde n_s número de muestras, $\delta(x)$ delta de Dirac y los pesos suman 1.
- Cada muestra es una 'hipotesis' del estado

Filtro de Partículas: Algoritmo

Input: Set of particles $\mathcal{X}_{k-1}$, Action $\mathbf{u}_{k-1}$, Observation $\mathbf{y}_k$
Output: Set of particles $\mathcal{X}_k$
$\tilde{\mathcal{X}} \leftarrow \emptyset$
$\mathcal{X} \leftarrow \emptyset$
for $i = 1, 2, \cdots, |\mathcal{X}_{k-1}|$ **do**
  $\mathbf{x}_k^i \leftarrow sample(p(\mathbf{x}_k|\mathbf{u}_{k-1}, \mathbf{x}_{k-1}^i))$ ▷ State trans.
  $w_k^i \leftarrow p(\mathbf{y}_k|\mathbf{x}_k^i)$ ▷ Observation
  $\tilde{\mathcal{X}} \leftarrow \tilde{\mathcal{X}} \cup \{(\mathbf{x}_k^i, w_k^i)\}$
end for
for $i = 1, 2, \cdots, |\tilde{\mathcal{X}}|$ **do** ▷ Resampling
  $j \leftarrow draw(p \sim w_k^j)$
  $\mathcal{X} \leftarrow \mathcal{X} \cup \{(\mathbf{x}_k^i, w_k^i)\}$
end for
return $\mathcal{X}$

Filtro de Partículas: *Importance Sampling*

- Tras aplicar la predicción (transición de estado y observación) si no se re-muestrean las partículas algunas estarán en zonas de baja probabilidad ($p(\hat{\mathbf{x}}_k|\mathbf{y}_{0:k-1},\mathbf{u}_{1:k-1})$)
- En el paso de *resampling* se recalculan los pesos w^i_{k+1} de las partículas para obtener una estimación de la probabilidad *a posteriori* ($p(\hat{\mathbf{x}}_k|\mathbf{y}_{0:k},\mathbf{u}_{1:k-1})$)
- *Importance sampling* permite estimar propiedades de una distribución de probabilidad a partir de muestras generadas por otra distribución

Resumen

- Asociación de datos, *landmarks* y signaturas
- Distancia de Mahalanobis y χ^2 para detección de *outliers* (y nuevas *landmarks*)
- El problema de *SLAM* en coordenadas Cartesianas y *range & bearing*
- Asociación de datos en *SLAM*
- Cerrado de bucles
- Filtro de partículas