

### 3. El filtro de Kalman extendido

#### 3.1. Introducción

En el apartado anterior se ha presentado la definición del problema de la localización y el mapeo simultáneo al igual que una ligera descripción del filtro de Kalman extendido. En este apartado se va a profundizar en la formulación matemática del mismo y su particularización para el problema del SLAM.

El robot con el que se va a trabajar será de direccionamiento lateral, es decir, el direccionamiento viene dado por la diferencia de velocidad entre las ruedas laterales, las cuales también se encargan de la tracción. Adicionalmente tendrá una tercera rueda de soporte adelantada con respecto a las dos laterales.

El estado estimado del robot se representa como  $x_r = [x_r, y_r, \theta_r]^T$  y el estado de control como  $u = [v_1, v_2]^T$  donde  $v_1$  y  $v_2$  son las velocidades de la rueda derecha y de la rueda izquierda respectivamente.

Un modelo dinámico general del vehículo puede definirse como

$$x_{rk+1} = f(x_{rk}, u_k) + q_k \quad (35)$$

donde

$f$  es un modelo no lineal del vehículo que toma el estado actual  $x_{rk}$  y la entrada de control  $u_k$  como variables.

$q_k$  es el ruido del proceso con media cero y covarianza  $Q_k$ .

$k$  simboliza un índice de tiempo discreto, el modelo se actualizará en los instantes constantes de tiempo  $t = kT$  para un período  $T$ .

Con estas condiciones se puede realizar la siguiente definición para  $f$

$$f = \begin{bmatrix} x_r + \Delta S \cos\left(\theta_r + \frac{\Delta\theta}{2}\right) \\ y_r + \Delta S \sin\left(\theta_r + \frac{\Delta\theta}{2}\right) \\ \theta_r + \Delta\Theta \end{bmatrix} \quad (36)$$

donde  $\Delta S = \frac{v_1 + v_2}{2}$  y  $\Delta\Theta = \frac{v_1 - v_2}{B}$ .

Con esta definición puramente cinemática del robot se pasa a aplicar una aproximación al filtro de Kalman extendido.

### 3.2. Aproximación al filtro de Kalman extendido

El filtro de Kalman es un conjunto de ecuaciones que proporcionan una solución eficiente al problema de mínimos cuadrados. El filtro es muy potente en diferentes aspectos ya que proporciona una estimación para instantes pasados, presentes y futuros, incluso cuando la naturaleza exacta del sistema es desconocida a priori. El FKE es una técnica de linealización de modelos dinámicos no lineales para poder aplicar sobre ellos el filtro de Kalman.

Con estas condiciones el FKE se presenta como un buen candidato para crear un mapa del entorno que simultáneamente se utilizará para localizar el robot.

Antes de comenzar a desarrollar las ecuaciones hay que aclarar las notaciones que se van a utilizar:

- $x_{k|k}$  Es la estimación de  $x$  en el instante  $k$  dando todas las medidas disponibles. Normalmente llamado estimación.
- $x_{k+1|k}$  Es la estimación de  $x$  en el instante  $k+1$  dando las  $k$  primeras medidas. Normalmente llamado predicción.

Realizada esta aclaración tendremos que el estado del robot será

$$x_{k|k} = X_k + \eta_k \tag{37}$$

El vector de estado completo será  $x = [x_r^T x_1^T x_2^T \dots x_N^T]^T$  donde  $x_r = [x_r y_r \theta_r]^T$  es la posición estimada del robot y  $x_i = [x_i y_i]^T$  es la estimación de la posición del hito  $i$ .  $X$  es el verdadero vector de estados y  $\eta$  es el error en la estimación. La covarianza del error estimado  $P_{k|k} = E[\eta_k \eta_k^T]$  del estado del sistema tendrá la forma

$$P_{k|k} = \begin{bmatrix} P^{rr} & P^{r1} & \dots & P^{rN} \\ P^{1r} & P^{11} & \dots & P^{1N} \\ \vdots & \vdots & \ddots & \vdots \\ P^{Nr} & P^{N1} & \dots & P^{NN} \end{bmatrix} \tag{38}$$

Las submatrices  $P^{rr}, P^{ri}, P^{ii}$  y  $P^{ij}$  son respectivamente la covarianza del robot consigo mismo, del robot con los hitos que conforman el entorno, la covarianza de los hitos consigo mismos, y la covarianza cruzada entre hitos.

El robot y el mapa están representados por un único vector de estados  $x$  con la correspondiente estimación del error de covarianza  $P$  en cada instante de tiempo. Dada la ecuación (35) y las ecuaciones del sistema para los hitos

$$x_{i_{k+1}} = x_{i_k} \tag{39}$$

El FKE se emplea para estimar el estado  $x$  y la covarianza  $P$  dada la medición  $z$ . La estimación se produce a través de un paso de predicción, causado por el movimiento del

vehículo, y un paso de actualización, que tiene lugar cuando los hitos se vuelven a observar. Cuando un nuevo hito se observa hay que añadirlo al vector de estados y a la matriz de covarianza. Así la evolución estocástica del mapa se divide en tres pasos: movimiento del vehículo por tanto una actualización de la odometría, integración de los hitos y reobservación de los mismos.

### 3.2.1 Movimiento del vehículo (paso de predicción)

Cuando el robot realiza el siguiente movimiento el estado estimado del vehículo en el instante de tiempo  $k+1$  viene dado teniendo en cuenta la esperanza en el modelo del estado de transición por

$$x_{r_{k+1|k}} = E[f(X_k, u_k)] \approx f(x_{r_{k|k}}, u_k) \quad (40)$$

Dado que el ruido del proceso tiene media cero, el término se vuelve cero cuando se evalúa el valor estimado en la ecuación (35) para determinar  $x_{r_{k+1|k}}$ . La covarianza  $P^{rr}_{k|k}$  se propaga a través del modelo lineal del estado de la transición del FKE proporcionando  $P^{rr}_{k+1|k}$  que viene dado por

$$P^{rr}_{k+1|k} = J_x P^{rr}_{k|k} J_x^T + Q_k \quad (41)$$

donde  $Q_k = E[q_k q_k^T]$  y  $J_x$  es el jacobiano de  $f$  con respecto a  $X$  evaluado para  $x_{k+1|k}$ . Este paso proporciona una posición estimada del robot exclusivamente basada en el modelo del robot y la odometría. Es también necesario actualizar las correlaciones cruzadas robot-hito

$$P^{ri}_{k+1|k} = J_x P^{ri}_{k|k} \quad (42)$$

Sin embargo no es necesario actualizar las correlaciones cruzadas entre los hitos ni la covarianza para cada uno de ellos, ya que la posición del hito en el mapa no se ve afectada por un cambio en el estado del robot.

### 3.2.2 Integración del nuevo hito

Cuando un nuevo hito  $L_{new} = [\rho \quad \theta]$  es observado y validado el estado del mismo  $x_{N+1}$  se incorpora al vector de estado del sistema.

$$x_{N+1} = m(x_{k+1|k+1}, L_{new}) \quad (43)$$

$$= \begin{bmatrix} x_{r_{k+1|k+1}} + \rho \cos(\theta_{r_{k+1|k+1}} + \theta) \\ y_{r_{k+1|k+1}} + \rho \sin(\theta_{r_{k+1|k+1}} + \theta) \end{bmatrix} \quad (44)$$

$$x_{k+1|k+1} \leftarrow \begin{bmatrix} x_{k+1|k+1} \\ x_{N+1} \end{bmatrix} \quad (45)$$

Se completa la matriz de covarianza  $P$  para que abarque al nuevo hito, para ello se calcularán las submatrices siguientes que son respectivamente la covarianza del hito  $N+1$ , la covarianza cruzada hito-robot y la covarianza cruzada hito-hito

$$P^{N+1N+1} = J_{x_r} P^{rr}_{k+1|k+1} J_{x_r}^T + J_z R J_z^T \quad (46)$$

$$P^{rN+1} = P^{N+1r} = P^{rr}_{k+1|k+1} J_{x_r}^T \quad (47)$$

$$P^{N+1i} = P^{iN+1} = J_{x_r} P^{ri}_{k+1|k+1} \quad (48)$$

Las matrices  $J_{x_r}$  y  $J_z$  son los jacobianos de  $m$  con respecto al estado del robot  $x_r$  y al punto de referencia  $L_{new}$ .

### 3.2.3 Reobservación de los hitos (paso de actualización)

Cuando un hito presente en el vector de estados es observado se utiliza el paso de actualización del FKE para actualizar el estado del mapa incluyendo la posición del robot. Definiendo

$$\begin{aligned} \bar{x}_i &= x_{i_{k+1|k}} - x_{r_{k+1|k}} \\ \bar{y}_i &= y_{i_{k+1|k}} - y_{r_{k+1|k}} \end{aligned} \quad (49)$$

La ecuación del modelo de observación para el punto de referencia  $i$  tiene la forma

$$z_{ik+1} = \begin{bmatrix} \rho_{ik+1} \\ \theta_{ik+1} \end{bmatrix} = \begin{bmatrix} \sqrt{\bar{x}_i^2 + \bar{y}_i^2} \\ \arctan \frac{\bar{y}_i}{\bar{x}_i} - \theta_{r_{k+1|k}} \end{bmatrix} + n_{zi} = h_i(x_{k+1|k}) + n_{zi} \quad (50)$$

Se considera que el ruido del proceso  $n_{zi}$  es un ruido blanco gaussiano con covarianza  $R_i$ . Si los  $N$  hitos son observados el modelo de la observación vendrá dado por las siguientes ecuaciones

$$z_{k+1} = \begin{bmatrix} z_{1k+1} \\ \vdots \\ z_{Nk+1} \end{bmatrix} \quad (51)$$

$$h = \begin{bmatrix} h_1 \\ \vdots \\ h_N \end{bmatrix} \quad (52)$$

$$R_{k+1} = \begin{bmatrix} R_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & R_N \end{bmatrix} \quad (53)$$

Denotando al jacobiano de  $h$  como  $H_x$ , las ecuaciones del paso de actualización son

$$x_{k+1|k+1} = x_{k+1|k} + K_{k+1}(z_{k+1} - h(x_{k+1|k})) \quad (54)$$

$$P_{k+1|k+1} = (I - K_{k+1}H_x)P_{k+1|k} \quad (55)$$

Donde  $K_{k+1}$  es la ganancia de Kalman que se calcula como

$$K_{k+1} = P_{k+1|k+1}H_x^T(H_xP_{k+1|k+1}H_x^T + R_{k+1})^{-1} \quad (56)$$

### 3.3. Programación del filtro de Kalman extendido

Para completar el desarrollo que se está llevando a cabo sobre el FKE, el siguiente paso lógico será explicar cómo a partir de la formulación matemática dada en el apartado anterior se consigue programar el filtro. A continuación se muestra el esquema general que se ha de seguir.

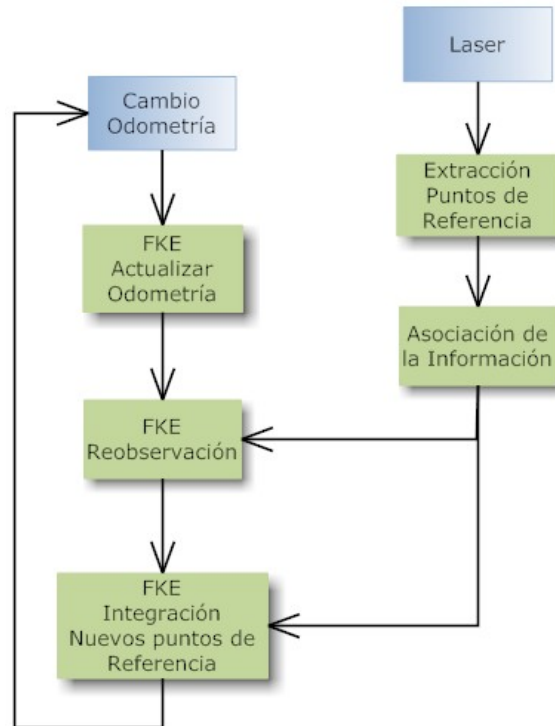


Figura 10

Cada vez que se produzca un cambio en la odometría tendremos que calcular la nueva posición en la que se encuentra el robot, actualizando las coordenadas  $x$ ,  $y$  y  $\theta$  con las últimas señales de control que se han proporcionado a las ruedas laterales para alcanzar el estado actual. A continuación, y considerando las medidas que proporciona el láser, se estudiarán todos los hitos que se observan. Con los que ya eran conocidos de iteraciones anteriores se corrige la posición actual del robot. Dado que conocemos la estimación de sus coordenadas de la posición anterior y sabemos el avance que se ha producido, calculando la diferencia entre la posición real de los hitos y la que se espera de los mismos suponiendo que el robot se encuentre realmente en la posición que le indica la odometría, se puede reajustar el estado actual del robot. Los hitos que se han observado por primera vez se incluyen en el conjunto de hitos reconocidos para futuras iteraciones.

Hay que observar que para la programación del filtro se han alternado los pasos 2 (Integración de los nuevos puntos de referencia) y 3 (Reobservación). Con idea de reducir los cálculos en el paso de reobservación y para no tratar los nuevos hitos hasta tener un mejor conocimiento de los mismos, ya que no se tratan en la primera iteración cuando son reconocidos por primera vez sino que se tratan en las posteriores cuando ya se han observado más veces.

### 3.3.1 Variables principales durante la simulación

Primero se van a presentar todas las variables que se van a utilizar para luego explicar como se van actualizando y calculando cada una de las mismas a través de los diferentes pasos.

#### 3.3.1.1 Vector de estados $X$

Es un vector columna que contiene la posición estimada actual del robot, coordenadas  $x$ ,  $y$  y  $\theta$ . Junto con las posiciones de todos los hitos utilizados en el proceso del FKE. Por tanto, este vector tendrá dimensiones  $(3+2n) \times 1$ , siendo  $n$  el número de hitos.

$$X = [x_r \quad y_r \quad \theta_r \quad x_1 \quad y_1 \quad \dots \quad x_n \quad y_n]^T$$

#### 3.3.1.2 Matriz de covarianza $P$

La matriz de covarianza contiene las relaciones de la posición del robot consigo mismo, con las posiciones de los hitos, las relaciones entre las posiciones de los hitos y las relaciones cruzadas entre hitos.

$$\begin{bmatrix} P^{rr} & \dots & P^{ri} & \dots & \dots \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ P^{ir} & \dots & \ddots & P^{ij} & \vdots \\ \vdots & \dots & P^{ji} & \ddots & \vdots \\ \vdots & \dots & \dots & \dots & P^{n+n+1} \end{bmatrix}$$

La primera submatriz  $P^{rr}$  es la covarianza del estado del robot por tanto será una matriz  $3 \times 3$ . Las tres primeras filas de la matriz, representadas por  $P^{ri}$ , son las correlaciones entre el robot y los diferentes hitos, al igual que las tres primeras columnas

que son las correlaciones de los hitos con el estado del robot. La primera será una matriz  $3 \times (n-3)$  mientras que la segunda será  $(n-3) \times 3$ . Todas las matrices por debajo de la diagonal serán las traspuestas de las matrices correspondientes por encima de la diagonal. El resto de submatrices que conforma la matriz de covarianza serán matrices  $2 \times 2$ , ya que los hitos sólo se caracterizan por sus coordenadas  $x_i$  e  $y_i$ .

#### 3.3.1.3 Ganancia de Kalman $K$

La ganancia de Kalman es un ponderador que le dará mayor o menor peso a la odometría en detrimento o a favor de los hitos observados respectivamente. Es decir, si se parte de la base de que el sistema de medida para observar los hitos es inferior en

exactitud que el sistema de odometría, está claro que se valorará más la estimación de la posición obtenida por la odometría que la calculada a partir de los hitos observados. Por el contrario si el sistema de observación de los hitos es muy superior al utilizado para obtener el cambio en la odometría, se considerará más la posición estimada por los hitos que la proporcionada por la odometría.

Usando la predicción de la posición actual es posible estimar dónde deberían encontrarse los hitos. Normalmente hay ciertas discrepancias, a la diferencia entre la posición predicha del robot y la posición real se le conoce como innovación.

$$\begin{bmatrix} x_r & x_b \\ y_r & y_b \\ \theta_r & \theta_b \\ x_{1r} & x_{1b} \\ y_{1r} & y_{1b} \\ \vdots & \vdots \\ x_{nr} & x_{nb} \\ y_{nr} & y_{nb} \end{bmatrix}$$

Las tres primeras filas muestran cuánto se debe conseguir de la innovación a partir de la primera, segunda y tercera fila del vector de estado respectivamente. La primera fila muestra cuánto habría que aumentar el primer término del vector  $X$  para cubrir la diferencia entre la posición estimada y la actual. La primera columna de la primera fila muestra cuánto debe ser ganado de la innovación en términos de rango, mientras que la segunda muestra en términos de orientación. La matriz continúa de la misma forma para el resto de coordenadas del robot y para las de los hitos, por tanto será una matriz de  $(3+2n) \times 2$ .

### 3.3.1.4 Jacobiano del modelo de medida $H$

El modelo de medida define como obtener la estimación del rango y de la orientación para un punto de referencia

$$\begin{bmatrix} \text{rango} \\ \text{orientación} \end{bmatrix} = \begin{bmatrix} \sqrt{(\lambda_x - x)^2 + (\lambda_y - y)^2} + v_r \\ \tan^{-1}\left(\frac{\lambda_y - y}{\lambda_x - x}\right) - \theta + v_\theta \end{bmatrix} \quad (57)$$

donde  $\lambda_x$  y  $\lambda_y$  representan la posición almacenada del hito,  $x$ ,  $y$  y  $\theta$  representan la posición del robot. Esto dará una estimación para la posición del punto de referencia visto desde la nueva posición del robot.

El jacobiano de esta matriz respecto a las coordenadas de la posición del robot será

$$H_x = \begin{bmatrix} \frac{x - \lambda_x}{r} & \frac{y - \lambda_y}{r} & 0 \\ \frac{\lambda_y - y}{r^2} & \frac{\lambda_x - x}{r^2} & -1 \end{bmatrix} \quad (58)$$

$H_x$  muestra cuánto cambia el rango y la orientación cuando cambian  $x$ ,  $y$  y  $\theta$ . El primer elemento de la primera fila es el cambio del rango con respecto a  $x$ . El segundo

elemento en la primera fila es con respecto a  $y$ . El tercer valor es con respecto a  $\theta$  y es cero porque el rango no cambia con la rotación del robot. La segunda fila proporciona la misma información pero con relación a la orientación.

Es necesaria cierta información adicional para llevar a cabo el FKE para SLAM.

$$\begin{matrix} x_r & y_r & \theta_r & x_1 & y_1 & x_2 & y_2 & x_3 & y_3 \\ \left[ \begin{array}{ccccccccc} A & B & C & 0 & 0 & -A & -B & 0 & 0 \\ D & E & F & 0 & 0 & -D & -E & 0 & 0 \end{array} \right] \end{matrix}$$

Se supone que se ven tres hitos desde la nueva posición. Cuando se trata el segundo hito se utilizará una matriz como la descrita arriba. La fila superior es sólo para indicar a que coordenada se refiere cada columna, primero estarán las del robot y luego las de los hitos. Las tres primeras columnas se corresponden con las tres columnas de la matriz  $H_x$ , por cada hito se añaden dos columnas más. Cuando se utiliza la matriz para el segundo hito se rellenarán sus columnas como se muestra arriba, el resto de las columnas relacionadas con otros hitos se igualan a cero.

### 3.3.1.5 Jacobiano del modelo de predicción $J_x$

El jacobiano del modelo de predicción está estrechamente relacionado con el modelo de predicción. Éste define cómo predecir la posición del robot a partir de la posición de anterior. Esto se consigue a partir de la siguiente fórmula

$$f = \begin{bmatrix} x + \Delta t \cos \theta + q \Delta t \cos \theta \\ y + \Delta t \sin \theta + q \Delta t \sin \theta \\ \theta + \Delta \theta + q \Delta \theta \end{bmatrix} \quad (59)$$

Donde  $x$  e  $y$  son las coordenadas del robot,  $\theta$  es la rotación,  $\Delta t$  es el cambio en la orientación y  $q$  es el término de error. Asumiendo que se puede acceder directamente a los incrementos a través de las señales de control, que el ruido del proceso se engloba en una única variable que se tratará más adelante y que se va a trabajar con un modelo linealizado, el jacobiano del modelo de predicción quedará como

$$\begin{bmatrix} 1 & 0 & -\Delta t \sin \theta \\ 0 & 1 & \Delta t \cos \theta \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & -\Delta y \\ 0 & 1 & \Delta x \\ 0 & 0 & 1 \end{bmatrix} \quad (60)$$

### 3.3.1.6 Jacobianos específicos del SLAM $J_{xr}$ y $J_z$

Cuando se implementa el FKE para SLAM se utilizan jacobianos que no suelen aparecer en las implementaciones básicas del filtro. Estos jacobianos aparecen en la integración de nuevos hitos que es el único paso que difiere de la estimación normal del FKE.

El primero es el jacobiano  $J_{xr}$  y es básicamente igual que el jacobiano del modelo de predicción excepto por el hecho de que comienza sin el término de rotación. Es el jacobiano de la predicción de los hitos, que no incluye la predicción de  $\theta$ , respecto a las coordenadas del robot.

$$J_{xr} = \begin{bmatrix} 1 & 0 & -\Delta t \sin \theta \\ 0 & 1 & \Delta t \cos \theta \end{bmatrix} \quad (60)$$

El segundo es el jacobiano  $J_z$  y es también el modelo de predicción para los hitos pero en este caso con respecto al rango y la orientación

$$J_z = \begin{bmatrix} \cos(\theta + \Delta\theta) & -\Delta t \sin(\theta + \Delta\theta) \\ \sin(\theta + \Delta\theta) & \Delta t \cos(\theta + \Delta\theta) \end{bmatrix} \quad (60)$$

### 3.3.1.7 Ruido del proceso $Q$

Se ha asumido que el proceso tiene un ruido gaussiano proporcional a los controles dados al robot. La matriz  $Q$  se calcula como  $Q = WCW^T$  donde  $W = [\Delta t \cos \theta \quad \Delta t \sin \theta \quad \Delta\theta]^T$ .

La matriz  $C$  es una representación de cómo de exacta es la odometría. El valor suele estimarse realizando experimentos y comprobando cual es el valor que mejor se ajusta a los resultados obtenidos.

### 3.3.1.8 Ruido de la medida $R$ y $V$

Se supone que la gama de dispositivos de medición también tienen ruido Gaussiano proporcional al rango y a la orientación. Se calcula como  $VRV^T$ , con  $V$  como una matriz identidad de 2x2.  $R$  es también una matriz 2x2 de números sólo en la diagonal. En la esquina superior izquierda se tiene el rango  $r$  multiplicado por la constante  $c$  y en la esquina inferior derecha la orientación  $b$  multiplicada por la

constante  $d$ . Las constantes deben representar la exactitud de la medición del dispositivo. Si, por ejemplo, el error del rango es de 1 cm,  $c$  debe ser una gaussiana con varianza igual a 0,01. Si el error de en la orientación es siempre de 1 grado, un buen valor para  $bd$  sería la unidad. Suponiendo que se utilizan grados para las mediciones de la orientación. Por lo general, no va a tener sentido hacer que el error en el ángulo sea proporcional con el tamaño del ángulo.

### 3.3.2 Pasos del proceso

#### 3.3.2.1 Extracción de los puntos de referencia

Se ha comentado durante todo el texto la necesaria utilización de los puntos de referencia para la corrección del estado del robot pero aún no se ha comentado que se entiende exactamente por un punto de referencia o hito.

Son marcas fácilmente reobservables y distinguibles del entorno. Son utilizadas por el robot para saber donde se encuentra. Las características principales que deben cumplir son:

- deben ser reobservables.
- deben distinguirse las unas de las otras fácilmente sin dar lugar a error.
- deben cubrir todo el terreno que conforma el entorno, ya que no debe haber grandes zonas sin que el robot encuentre ningún punto de referencia ya que induce a que éste se pierda.
- deben ser estacionarias. No es buena idea crear un punto de referencia móvil.

Cómo extraer los puntos de referencia del entorno:

Hay muchas formas de hacerlo y depende en mayor parte del tipo de punto de referencia que se quiera localizar.

##### Spike landmarks

Se basa en las variaciones bruscas en la lectura de dos sensores consecutivos. Si la diferencia es inferior a una distancia, a 0.5 cm por ejemplo, significará que uno de los sensores está detectando un objeto que el consecutivo no ve. También se puede determinar dando tres valores consecutivos cercanos entre ellos A, B y C. Restamos A menos B y C menos B, añadiendo los dos resultados podemos obtener un valor que indique si hay alguna anomalía.

RANSAC (Random Sampling Consensus)

Este sistema sirve para extraer líneas de un láser. RANSAC encuentra las líneas de referencia arbitrariamente tomando muestras de las lecturas del láser y aproximando éstas para encontrar la mejor línea que adapte las lecturas. Una vez se encuentra la línea correcta se comprueba cuántas de las lecturas del láser se aproximan correctamente a la línea calculada. Si el número es superior a un cierto umbral (valor de consenso) se puede afirmar que se ha localizado una línea recta, lo que implica que se "ha visto" un segmento de pared.

El algoritmo está pensado para un láser de 180 ° de observación y con un rango de medida por grado:

mientras

- haya todavía lecturas desasociadas
- el nº de lecturas es superior al consenso
- se hayan hecho menos de N pruebas

hacer

- Seleccionar una lectura del láser
- Tomar S lecturas en un rango de D grados entorno al valor inicial
- Con la lectura inicial y las S lecturas posteriores calcular la recta que mejor ajusta los puntos
- Determinar cuántas lecturas pertenecen a la línea calculada con una distancia a la misma inferior a X centímetros
- Si el nº de lecturas del láser que conforman la línea es superior al nº de consenso C hacer:
  - Calcular la línea que mejor se ajusta por mínimos cuadrados a todas las lecturas láser que pertenecen a la recta anterior.
  - Añadir esta nueva línea a las líneas que ya hayan sido extraídas.
  - Eliminar las lecturas asociadas a la recta de la lista de las no asociadas.

Fin Hacer

N: número máximo de prueba para encontrar líneas

S: número de lecturas necesarias para ajustar la primera línea

D: número de grados de la medida inicial de donde se tomarán los primeros S valores

X: máxima distancia permitida para que una lectura esté asociada a una línea

C: número de puntos que deben pertenecer a la línea para que ésta sea considerada.

Múltiples estrategias Scan-Matching

Consiste en corregir la odometría localmente realizando dos barridos láser sucesivos. Esto implica que hay dos correcciones consecutivas y que nos permiten descartar puntos de referencia conflictivos.

### 3.3.2.2 Asociación de la información

El principal problema de la asociación de la información es identificar puntos de referencia extraídos de diferentes lecturas del láser, es decir, reconocer que el nuevo hito que se está “viendo” a través de los sensores es el mismo que se vio en un instante anterior en otras coordenadas diferentes del robot.

Los problemas que pueden surgir al intentar realizar una asociación son: que no se observan los mismos puntos de referencia en cada paso de la simulación, que puede identificarse un punto de referencia como tal y fallar el algoritmo de asociación porque no vuelve a verse, o asociar de forma incorrecta un nuevo hito con otro visto anteriormente de forma que el robot crea estar en un punto totalmente diferente en el mapa de su posición actual.

Para los dos primeros casos incluso disponiendo de un perfecto algoritmo de extracción de hitos se puede incurrir en estos fallos, así que un buen método para la asociación de la información será aquel que minimice estos errores.

Se supone que existe una base de datos con todos los hitos que se han ido observando en todas las lecturas recibidas del láser, ésta estará inicialmente vacía. La primera regla que se impone tiene como fin reducir al mínimo el segundo error descrito y es que no se utilizará ningún hito en el proceso del SLAM hasta que no haya sido visto un mínimo de  $N$  veces. Para iniciar un nuevo punto de referencia visto por el láser se utiliza la técnica llamada del vecino más próximo desarrollada por Feder en 1999, y que consiste en lo siguiente:

Cuando llega una nueva lectura del láser se usa la extracción de hitos para sacar todos los hitos visibles

Se asocia cada nuevo hito extraído con el hito más próximo de la base de datos que se haya visto más de  $N$  veces.

Someter cada par de asociación (hito nuevo – hito en la base de datos) a una puerta de validación.

Si el par pasa la puerta de validación debe ser el mismo punto de referencia que se ha vuelto a ver, así que se incrementará en la base de datos el número de veces que se ha visto a este punto de referencia.

Si el par falla, habrá que añadir el punto de referencia nuevo en la base de datos inicializando el número de veces que se ha visto a uno.

Normalmente la forma de calcular el vecino más cercano es mediante la distancia Euclídea o utilizando la distancia de Mahalanobis. La puerta de validación usa el hecho de que la implementación del FKE proporciona un área de incertidumbre sobre la

observación del hito. De tal forma que se puede determinar si un hito observado es uno de los hitos almacenados en la base de datos comprobando si descansa sobre el área de incertidumbre de alguno de ellos. Esta área se puede representar gráficamente y se conoce como elipse de error. Esta puerta de validación basada en la aproximación está desarrollada en el espacio de innovación [41] incorporando la incertidumbre de las medidas y la incertidumbre del estado del robot. La matriz de innovación  $S_i$  para el punto de referencia  $i$  vendrá dada por:

$$S_i = H_{x_i} \begin{bmatrix} P^{rr} & P^{ri} \\ P^{ir} & P^{ii} \end{bmatrix} H_{x_i}^T + R_i \quad (53)$$

$$\text{Donde } H_{x_i}^T = \frac{dh_i([x_r, x_i])}{d[x_r, x_i]}$$

Si se define la innovación como  $v_i = z_i - h_i(x_i)$  la puerta de validación puede expresarse como

$$v_i^T S_i^{-1} v_i \leq \lambda \quad (54)$$

Para un sistema con dos grados de libertad, un valor de  $\lambda = 9.0$  proporciona la región de mínimo volumen que contiene las medidas con probabilidad del 98.9% .

### 3.3.2.3 Actualización de la odometría (Primer paso FKE)

Se ha producido un cambio en el estado del robot, habrá que actualizar el vector de estados porque se han modificado sus tres primeras componentes y habrá también que actualizar las submatrices de la matriz de covarianza  $P$  que involucren el estado del robot.

Para hacer una estimación de la nueva posición se parte de los controles dados al robot

$$\begin{bmatrix} x + \Delta t \cos \theta + q \Delta t \cos \theta \\ y + \Delta t \sin \theta + q \Delta t \sin \theta \\ \theta + \Delta \theta + q \Delta t \end{bmatrix}$$

En el caso de que se considere un modelo simplificado de odometría se pueden añadir directamente los controles para actualizar el estado.

$$\begin{bmatrix} x + \Delta x \\ y + \Delta y \\ \theta + \Delta \theta \end{bmatrix}$$

Una vez que se han modificado las tres primeras filas del vector de estados, lo siguiente sería modificar la matriz  $J_x$  que es el jacobiano del modelo de predicción que como se explicó en el anterior apartado sería calcular la siguiente matriz

$$\begin{bmatrix} 1 & 0 & -\Delta y \\ 0 & 1 & \Delta x \\ 0 & 0 & 1 \end{bmatrix}$$

Finalmente hay que calcular el ruido que afecta a la odometría, que vendrá dado por la matriz

$$Q = \begin{bmatrix} c\Delta x^2 & c\Delta x\Delta y & c\Delta x\Delta t \\ c\Delta y\Delta x & c\Delta y^2 & c\Delta y\Delta t \\ c\Delta t\Delta x & c\Delta t\Delta y & c\Delta t^2 \end{bmatrix}$$

Se conocen todas las variables necesarias para actualizar la matriz de covarianza

$$P^{rr}_{k+1|k} = J_x P^{rr}_{k|k} J_x^T + Q_k$$

$$P^{ri}_{k+1|k} = J_x P^{ri}_{k|k}$$

### 3.3.2.4 Corregir el estado a partir de la re-observación de los puntos de referencia (Segundo paso FKE)

La predicción hecha para la posición del robot no es completamente exacta debido a los errores de la odometría del robot. Se pueden compensar estos errores usando los puntos de referencia. Ya se han tratado los hitos, incluyendo como observarlos y como asociarlos a hitos que ya se conocen. Usando los puntos de referencia asociados se puede calcular la posición del robot en comparación a la posición que se cree que es la correcta. Esto es lo que se pretende realizar en este segundo paso, que se lleva a cabo para cada hito re-observado.

Los nuevos hitos no son tratados hasta el siguiente paso, retrasando así su incorporación para disminuir el coste computacional de este paso ya que las matrices de covarianza y del estado son menores.

Primero se trata de estimar donde se encuentra el hito usando la posición actual del robot y la posición almacenada del hito en la base de datos. Utilizando la ecuación (57) se consigue la distancia y la orientación al hito, se almacenarán estos datos en el vector  $h$ , esto se puede comparar con la distancia y la orientación que se consigue de la asociación de la información que se denotará  $z$ .

Se necesitan conocer algunos datos más antes de conocer la corrección que hay que hacer a la posición predicha del robot para obtener la real. De apartados anteriores se tenía para un caso genérico en el que sólo hay tres hitos

$$\begin{matrix} x_r & y_r & \theta_r & x_1 & y_1 & x_2 & y_2 & x_3 & y_3 \\ \left[ \begin{array}{ccccccccc} A & B & C & 0 & 0 & -A & -B & 0 & 0 \\ D & E & F & 0 & 0 & -D & -E & 0 & 0 \end{array} \right] \end{matrix}$$

$$\text{donde } A = \frac{x - \lambda_x}{r}, B = \frac{y - \lambda_y}{r}, C = 0, D = \frac{\lambda_y - y}{r^2}, E = \frac{\lambda_x - x}{r^2} \text{ y } F = -1$$

La matriz  $R$  que representa el ruido en la medida debe ser actualizada para reflejar los errores de la distancia y la orientación actuales. Un buen valor de comienzo para  $rc$  puede ser el valor del rango multiplicado por 0.01, lo que significa un error del 1% en la distancia. Un buen valor para el error de la orientación  $bd$  puede ser 1, lo que significa que hay un grado de error en la medida.

Ahora se puede calcular la ganancia de Kalman mediante la siguiente fórmula

$$K = PH^T(HPH^T + R)^{-1} \quad (55)$$

La ganancia de Kalman contiene un conjunto de número indicando cuánto de cada posición de los hitos y de la posición del robot debe ser considerada de acuerdo a la re-observación de los hitos para corregir la posición. El término  $(HPH^T + R)$  se llama covarianza de la innovación  $S$ , ya se mencionó en la parte de asociación de la información cuando se calcula la puerta de validación para un punto de referencia.

Finalmente se puede calcular un nuevo vector de estados usando la ganancia de Kalman

$$X = X + K(z - h) \quad (56)$$

Esta operación va a actualizar la posición del robot con todas las posiciones de los hitos. Observar que  $(z - h)$  produce un resultado de dos números que es el desplazamiento en distancia y orientación denotado como  $v$ .

### 3.3.2.5 Añadir los nuevos puntos de referencia al estado actual (Tercer paso FKE)

En este paso se va a actualizar en vector de estados  $X$  y la matriz de covarianza  $P$  con los nuevos hitos. El propósito es tener más puntos de referencia que puedan ser encontrados-asociados, así el robot tiene más hitos que pueden ser localizados y ayudarle en la creación y navegación sobre el mapa.

$$\text{Primero se añade un nuevo hito al vector de estados } X = [X \quad x_N \quad y_N]^T$$

También es necesario añadir una nueva fila y una nueva columna por cada nuevo hito encontrado a la matriz de covarianza. Primero se añade la covarianza para el nuevo hito en la casilla C calculada como

$$P^{N+1N+1} = J_{xr} P^{rr} J_{xr}^T + J_z R J_z^T$$

$$\begin{bmatrix} P^{rr} & \dots & P^{ri} & \dots & P^{rN+1} \\ \vdots & \ddots & & & \vdots \\ P^{ir} & & \ddots & & P^{iN+1} \\ \vdots & & & \ddots & \vdots \\ P^{N+1r} & \dots & P^{N+1i} & \dots & P^{N+1N+1} \end{bmatrix}$$

Después se añade la covarianza del robot con el nuevo hito en la esquina superior derecha de la matriz de covarianza calculándola como

$$P^{rN+1} = P^{rr} J_{xr}^T$$

La covarianza del hito con el robot es el valor traspuesto de la covarianza del robot con el hito, correspondiente a la esquina inferior izquierda de la matriz de covarianza

$$P^{N+1r} = (P^{rN+1})^T$$

Finalmente se calculan las covarianzas cruzadas hito-hito que se añaden en la última columna y en la última fila

$$P^{N+1i} = (P^{iN+1})^T = J_{xr} P^{ri}$$

Esto completa el último paso del proceso del FKE para SLAM. El robot está ahora preparado para moverse otra vez, observar los hitos, asociarlos, actualizar el estado del sistema usando la odometría, actualizar el estado del sistema usando la re-observación y por último añadir los nuevos puntos de referencia antes de seguir con el proceso.