Universidad Carlos III de Madrid
Departamento de Ingeniería de Sistemas y Automática

Robot Móviles

Filtro de Kalman

El filtro de Kalman es un algoritmo de estimación óptimo (de acuerdo a la función de coste de mínimos cuadrados) siempre que se trabaje con un sistema lineal y que el ruido que aparece, tanto en el sistema dinámico (robot en movimiento en nuestro caso) como en los sensores utilizados tenga una distribución normal. El problema abordado es el siguiente:

- Se cuenta con un modelo de la planta y del sistema sensorial que representa cómo evoluciona el sistema y cómo se miden las variables de estado en tiempo discreto:

$$x_k = F\left(x_{k-1}, u_{k-1}, v_{k-1}\right)$$
$$y_k = G\left(x_k, n_k\right)$$

  donde los subíndices hacen referencia a los instantes de tiempo discreto, la variable aleatoria x_k representa el estado del sistema, u_k es la entrada que genera el cambio en el sistema y v_k modela el ruido que se produce al hacer el movimiento. Además, y_k representa los valores medidos por los sensores y n_k modela el ruido que hay en las mediciones obtenidas por el sensor.

Bajo estas hipótesis, el objetivo es estimar, en cada instante $k \geq 1$, el estado x_k.

En el script de Matlab que acompaña a esta práctica (Tutorial_pos.m), se modela un robot que se mueve en 1 sola dimensión, con velocidad constante y sin entrada alguna que provoque cambios en el sistema.

$$x_k = x_{k-1} + v_k * dt$$
$$v_k = v_{k-1} = 0{,}2$$

Tareas:

1. Los ruidos del sistema (Q) y de las mediciones (R) tienen valores parecidos (del mismo orden de magnitud). Comprobar qué pasa cuando es mucho más alto que el otro y entender por qué.

2. El ruido introducido en la simulación de los sensores (sim_sen) y en el sistema sensorial tienen valores parecidos (del mismo orden de magnitud). Comprobar qué pasa cuando Q es mucho menor que sim_sen y entender por qué.

Partiendo del anterior, supongamos ahora un modelo más real, todavía en una sola dimensión, en el que existe una entrada constante al sistema, la aceleración. Ahora, al sistema anterior

$$x_k = x_{k-1} + v_k * dt + 0{,}5 * a * dt^2$$
$$v_k = v_{k-1} + a * dt$$
$$a = 0{,}1$$

Partiendo del script con nombre "Ejercicio_accel.m", que contienem el indicado en comentarios, el esqueleto de los pasos que se deben seguir. Programar el filtro de Kalman correspondiente al modelo con aceleración constante en el que los sensores del robot miden, de manera directa, su posición y su velocidad.

1. Para modelar los ruidos que afectan a la dinámica y a los sensores, elegir las desviaciones típicas a utilizar.

2. Dibujar los resultados y analizar si son correctos.

NOTA: se recomienda seguir la misma nomenclatura que en el scrpit anterior para facilitar la comprensión, y que el código dedicado a dibujar funcione.

Filtro extendido de Kalman

El filtro extendido de Kalman es una variación que se propone con el objetivo de abordar el problema de la estimación en sistemas cuyas funciones de estado y medición (*F, G*) puedan ser no lineales. Para ello, se definen aproximaciones de primer orden de estas expresiones que permiten linealizar el sistema:

$$A_{k-1} = \frac{\partial F}{\partial x}\left(x_{k-1|k-1}, u_{k-1}, 0\right)$$
$$C_k = \frac{\partial G}{\partial x}\left(x_{k|k-1}, 0\right)$$

Cuando el modelo es severamente no lineal, estas aproximaciones pueden generar esperanzas y covarianzas muy distintas a las esperanzas y covarianzas reales producto de la transformación de la variable aleatoria a través del modelo no lineal. Estas discrepancias, pueden conducir a un desempeño muy pobre del filtro o incluso a su divergencia.

Otro inconveniente del EKF es que requiere de las matrices jacobianas, cuyo cálculo de no es trivial en la mayor parte de las aplicaciones, por lo que es muy factible cometer errores de cálculo que luego son difíciles de detectar. Además, el requerimiento de estas matrices complica la implementación genérica del filtro (no restringida a una aplicación particular).

- **Ejemplo de aplicación**

Vamos a usar el filtro extendido de Kalman para estimar la posición de un robot cuya función de movimiento es no lineal. El robot móvil se mueve en el plano según el siguiente esquema:

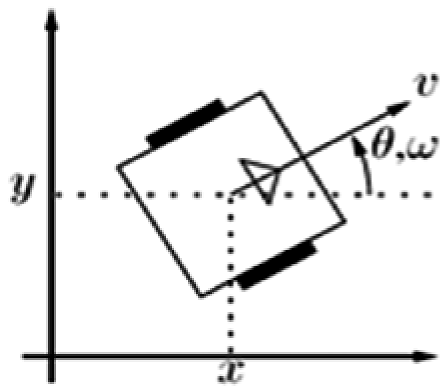

Figura 1. Movimiento del robot móvil en el plano.

Como se puede ver en la *Figura 1*, el avance lineal del robot viene determinado por la velocidad lineal (v) y por su orientación (Θ), definida como el ángulo del robot con respecto al eje X. Además, el robot cambia su orientación de manera lineal de acuerdo a su velocidad angular (w). Las velocidades angular y lineal aceleran hasta un valor constante según la ecuación:

$$v = V_{max} * \left(1 - e^{-\frac{t}{10}}\right)$$

$$w = W_{max} * \left(1 - e^{-\frac{t}{10}}\right)$$

El robot incorpora sensores para medir de manera directa la posición (x, y), la orientación (Θ) y la velocidad angular (w). V_{max} =1, W_{max} =5 .

Partiendo del script con nombre "Ejercicio_EKF.m", que contiene, indicado en comentarios, el esqueleto de los pasos que se deben seguir. Programar el filtro de extendido de Kalman correspondiente al modelo no lineal que se acaba de explicar.

Tareas:

1. Definir las ecuaciones de movimiento con respecto al tiempo.

2. Definir cuáles son las variables de estado del sistema, las entradas y las variables medidas.

3. Escribir de manera algebraica las ecuaciones del modelo dinámico y sensorial del sistema según:

$$x_k = f(x,u) = Ax_{k-1} + Bu_{k-1} + v_{k-1}$$

$$z_k = h(x) = \; Cx_k + n_k$$

4. Calcular las matrices Jacobianas de las funciones anteriores según:

$$F(x,u) = \frac{\partial f}{\partial x}(x_{k-1}, u_{k-1,}, 0)$$
$$H(x,u) = \frac{\partial h}{\partial x}(x_{k-1}, 0)$$

<u>IMPORTANTE</u>: puede haber matrices de los puntos 3 y/o 4 cuyos valores dependan del estado del robot, y por lo tanto no son contantes. En este caso, definir la matriz dentro de una función que tome como parámetro el estado. Las funciones en un script de Matlab deben definirse al final.

5. Rellenar el script de Matlab que se adjunta en la práctica completando todos los pasos del filtro extendido de Kalman. Para modelar los ruidos que afectan a la dinámica y a los sensores, elegir las desviaciones típicas a utilizar.

6. Dibujar los resultados y analizar si son correctos.

Anexo: recordatorio de la matriz Jacobiana.

$$y_i = F_i(x_1, \ldots, x_n)$$

$$\begin{bmatrix} \dfrac{\partial y_1}{\partial x_1} & \cdots & \dfrac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial y_m}{\partial x_1} & \cdots & \dfrac{\partial y_m}{\partial x_n} \end{bmatrix}$$