

Fundamentos de robótica móvil

TP5: Estimación de orientación con filtro de Kalman

Gonzalo Perez Paina, David Gaydou y Diego Gonzalez Dondo

Propuesta

Comprender mediante simulación la necesidad de la estimación de estado aplicada a un multirrotor de un grado de libertad tipo balancín a partir de mediciones de sensores inerciales MEMS (acelerómetro y giróscopo) que simulen mediciones realistas. Para esto es necesario, en primera instancia, generar mediciones con ruido a partir del estado verdadero (no disponible en un robot real) o ground-truth del multirrotor, para luego utilizar dichas mediciones en un filtro de Kalman que permita estimar el ángulo de orientación.

Descripción de la actividad

Unidades Inerciales

Las unidades inerciales están compuestas generalmente por tres giróscopos ortogonales y tres acelerómetros ortogonales, para medir velocidades angulares y aceleraciones lineales respectivamente. Los sensores de las unidades inerciales están afectados por diferentes fuentes de error. Para realizar una simulación de los mismos, y que esta sea lo más parecido o fiel a la realidad, es necesario modelarlos incluyendo todas las posibles fuentes de error.

Simulador balancín

La figura 1 muestra una representación esquemática de un modelo simplificado de multirrotor tipo balancín que gira a velocidad angular ω y describe un ángulo θ respecto a la horizontal. El giróscopo está ubicado de forma tal de poder medir la velocidad angular (ω) de rotación alrededor del eje de giro y el acelerómetro la componente (a) de la gravedad (g) a lo largo de la barra. O sea que en este caso ambos sensores son de un único eje.

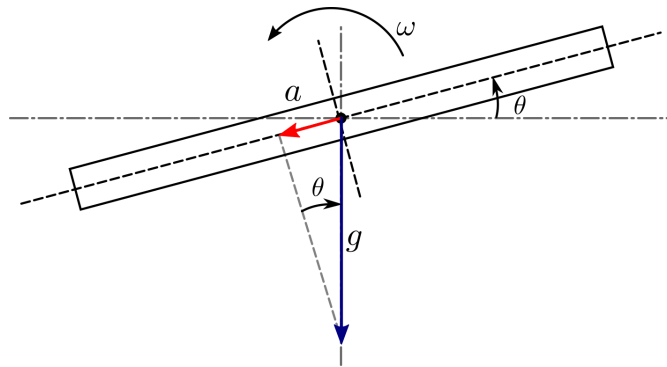


Figura 1: Balancín con medición del acelerómetro (a) y del giróscopo (ω).

En (Farrenkopf, 1978) se presenta el siguiente modelo para la estimación de actitud de una aeronave

$$\begin{aligned}\theta_M &= \theta_T + \eta_\theta \\ \omega_M &= \omega_T + \omega_I \\ \omega_I &= b_\omega + \eta_\omega \\ \dot{b}_\omega &= \eta_\alpha,\end{aligned}\tag{1}$$

donde b_ω es el bias del gir6scopo y los ruidos son: $\eta_\theta \sim \mathcal{N}(0, \sigma_\theta)$, $\eta_\omega \sim \mathcal{N}(0, \sigma_\omega)$ y $\eta_\alpha \sim \mathcal{N}(0, \sigma_\alpha)$. El sub6ndice T hace referencia al valor verdadero y M al valor medido (Perez Paina et al., 2011).

Consigna

En el repositorio de NaveGo¹ se encuentra un Toolbox de c6digo abierto escrito para MATLAB y GNU/Octave que permite realizar simulaciones y caracterizar unidades inerciales.

1. Clonar el repositorio y poner en marcha el simulador NaveGo.

Una vez instalado el repositorio realizar una simulaci6n de las dos unidades inerciales pre-configuradas con datos sint6ticos:

2. Correr el ejemplo que en la documentaci6n est6 bajo la secci6n “*INS/GNSS integration example using synthetic (simulated) data*” ([examples/synthetic-data/navego_example_synth.m](#)) y analizar las gr6ficas que se generan y comparar los resultados obtenidos con los sensores pre-configurados.

Los gir6scopos miden velocidades 6ngulares en cada uno de los ejes $\omega_M = \{\omega_{Mx}, \omega_{My}, \omega_{Mz}\}$ y los aceler6metros miden aceleraciones lineales en cada uno de los ejes $\mathbf{a}_M = \{a_{Mx}, a_{My}, a_{Mz}\}$. Las mediciones de estas magnitudes est6n afectadas por diferentes fuentes de error.

3. Analizar el script `gyro_gen.m` y `acc_gen.m` (bajo el directorio `simulation`) donde se implementa el modelo del gir6scopo y el aceler6metro con sus fuentes de errores.

En base a los scripts del punto anterior, estudiar c6mo se lleva a cabo la generaci6n de ruido de los sensores simulados y sus par6metros para aplicarlos en las simulaciones del balanc6n. Buscar la hoja de datos del sensor MicroStrain 3DM-GX1 y utilizar los par6metros de ruido en el simulador.

4. Modificar los scripts de simulaci6n del balanc6n para obtener datos de sensores simulados a partir del estado verdadero del robot (ground-truth) que incluya los ruidos correspondientes.

Teniendo en cuenta el modelo dado en (1) y las ecuaciones de proceso (2) y medici6n (3) necesarias para la formulaci6n del filtro de Kalman

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k + \mathbf{L}\mathbf{w}_k \quad (2)$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k; \quad (3)$$

definiendo adem6s los vectores como $\mathbf{x} = [\theta \quad b_\omega]^T$, $\mathbf{u} = \omega_M$ y $\mathbf{z} = \theta_M$.

5. Determinar las matrices \mathbf{F} , \mathbf{G} , \mathbf{L} y \mathbf{H} .

Dada la formulaci6n del filtro de Kalman

Predicci6n:

$$\hat{\mathbf{x}}_k^- = \mathbf{F}\hat{\mathbf{x}}_{k-1} + \mathbf{G}\mathbf{u}_{k-1}$$

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}_{k-1}$$

Inicializaci6n: \mathbf{x}_0 , \mathbf{P}_0

Correcci6n:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k)^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-$$

6. Determinar las matrices que modelan los ruidos de proceso y medici6n \mathbf{Q}_k y \mathbf{R}_k e implementar el filtro en un script de GNU/Octave para la estimaci6n de la orientaci6n θ .

Referencias

R.L. Farrenkopf. Analytic steady-state accuracy solutions for two common spacecraft attitude estimators. *AIAA Mechanics and Control of Flight Conference*, 1(4):282–284, 1978.

G. Perez Paina, D. Gaydou, J. Redolfi, C. Paz, and L. Canali. Experimental comparison of kalman and complementary filter for attitude estimation. In *Proceedings of the AST in 40th Argentine Conference on Informatics (JAIIO)*, pages 205–215, 2011.

¹<https://github.com/rodraz/NaveGo>