

FUSIÓN DE SENSOR ÓPTICO PARA NAVEGACIÓN INTEGRADA

Augusto J. Zumarraga

Centro Tecnológico Aeroespacial (CTA), FI-UNLP

uku@ing.unlp.edu.ar

Bernardo Concia

Departamento de Ingeniería Aeroespacial - FI-UNLP



Introducción

Para la implementación de sistemas de vuelo autónomo se requiere contar con información de posición y actitud (orientación en el espacio) de la aeronave, en tiempo real. En general esto se obtiene de un sistema de navegación basado en el uso de sensores inerciales. Dado que la navegación inercial pura no ofrecen una estimación estable (los errores crecen con el tiempo), salvo para ciertas aplicaciones y contando con sensores inerciales de grado adecuado, esto se transforma en un problema insalvable.

Mediante el procesamiento de imágenes es posible identificar "puntos clave" en el campo visual, y seguir su movimiento relativo en imágenes sucesivas; lo cual puede correlacionarse con los cambios de actitud y posición de la cámara. Esto proveen información útil para la navegación, particularmente para vuelo estacionario o a baja velocidad.

A continuación se presenta el esquema utilizado para el sensor visual, y los desarrollos matemáticos necesarios para su fusión con la navegación inercial utilizando un filtro de Kalman Extendido (EKF). Los resultados se obtienen mediante simulación.

Ecuaciones para la Navegación

Conociendo la condición inicial, es posible obtener estimaciones de posición y actitud integrando la velocidad angular y la fuerza específica (fuerza por unidad de masa, lo cual no incluye los efectos gravitatorios) medida por sensores inerciales solidarios al vehículo (*navegación strapdown*).

Con cualquier representación de la actitud es posible obtener la proyección \mathbf{h}^a de un vector $\mathbf{h} \in \mathbb{R}^3$ (fuerza, velocidad, etc.) en una terna- a , a partir de su proyección \mathbf{h}^b en una terna- b de diferentes maneras, como por ejemplo, computando para la actitud entre ambas ternas la matriz de rotación (o "cosenos directores") $\mathbf{C}_b^a \in \mathbb{R}^{3 \times 3}$: $b \rightarrow a$, tal que:

$$\mathbf{h}^a = \mathbf{C}_b^a \mathbf{h}^b$$

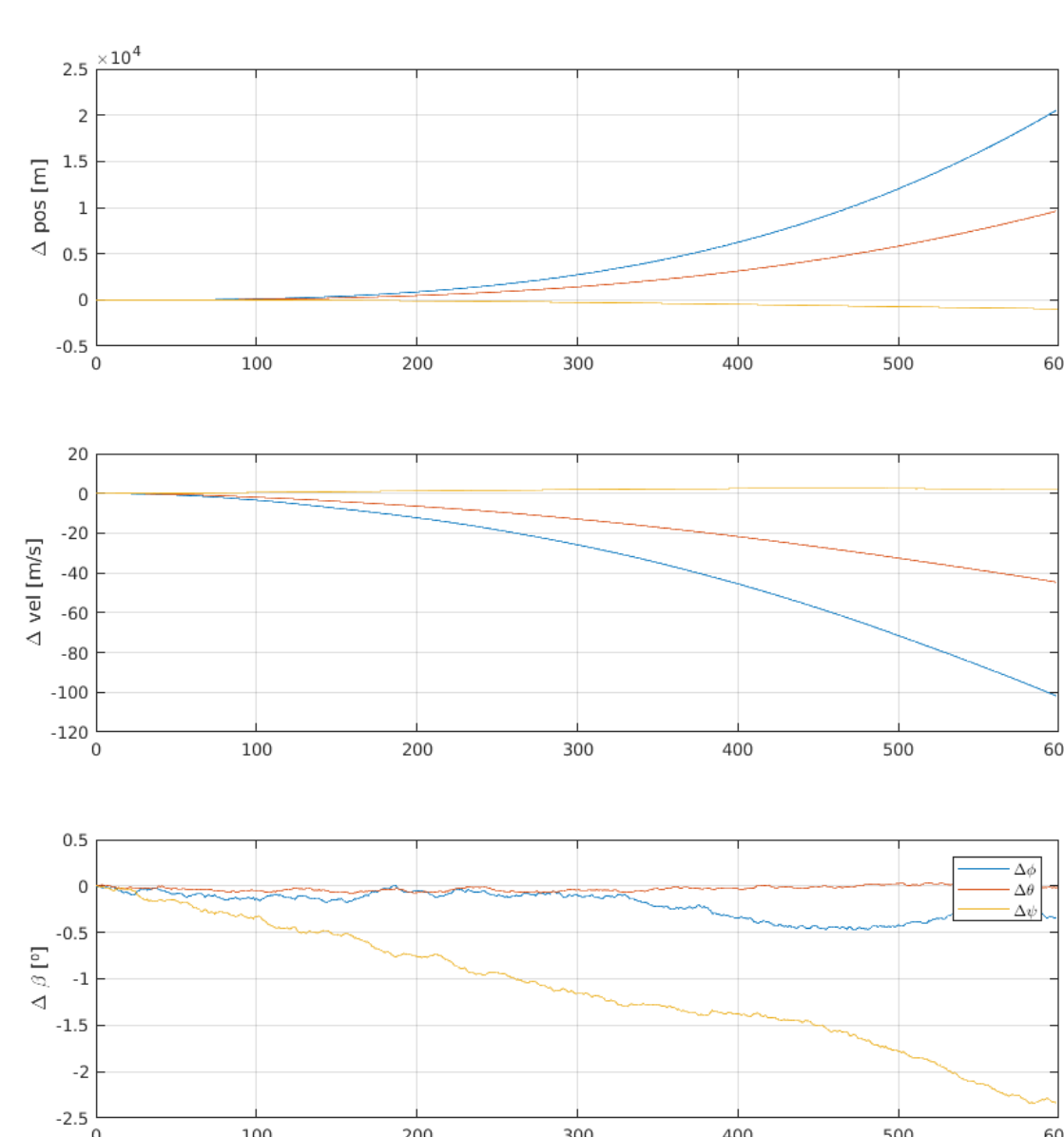
Con esto pueden proyectarse mediciones inerciales en la terna- b sobre la terna- e . Utilizando coordenadas ECEF las ecuaciones de navegación son [1]:

$$\begin{aligned} \dot{\mathbf{p}}^e &= \mathbf{v}^e \\ \dot{\mathbf{v}}^e &= \mathbf{C}_b^e(\mathbf{q}_b^e) \mathbf{f}^b + \boldsymbol{\gamma}^e(\mathbf{p}^e) - 2\boldsymbol{\Omega}_{ie}^e \times \mathbf{v}^e \\ \dot{\mathbf{q}}_b^e &= \frac{1}{2} \mathbf{q}_b^e \left\{ \begin{matrix} \boldsymbol{\omega}_{eb}^b \\ 0 \end{matrix} \right\} - \frac{1}{2} \left\{ \begin{matrix} \boldsymbol{\Omega}_{ie}^e \\ 0 \end{matrix} \right\} \mathbf{q}_b^e \end{aligned} \quad (1)$$

donde \mathbf{p} es el vector posición, \mathbf{v} es la velocidad lineal y $\boldsymbol{\omega}_{eb}$ es la velocidad angular entre las ternas e y b . La actitud está representada mediante el cuaternión unitario \mathbf{q} .

Además, $\boldsymbol{\Omega}_{ie}^e$ es la velocidad de rotación de la tierra respecto de la terna ECI, $\boldsymbol{\gamma}^e$ la aceleración gravitatoria ($\in \mathbb{R}^3$), y $\mathbf{C}_b^e \in \mathbb{R}^{3 \times 3}$ es la matriz de rotación para proyectar vectores de la terna móvil a la terrestre (determinada de forma biunívoca por la actitud).

La estimación obtenida de integrar las ecuaciones de navegación no es asintóticamente estable, y por lo tanto, diverge debido a los ruidos de medición de los sensores inerciales. Con sensores de bajo costo, como los utilizados en teléfonos celulares y drones, la estimación de posición es prácticamente inutilizable a los pocos segundos, como puede apreciarse en la esta figura.



Fusión

Para estabilizar la estimación de la navegación inercial es necesario obtener información adicional recurriendo a sensores denominados "exoeceptivos", dado que estos requieren interactuar con el entorno, a diferencia de los inerciales que son "introceptivos".

Habitualmente se utilizan receptores GNSS que dan información completa de posición, barómetros que permiten determinar variaciones de altura y magnetómetros que ofrecen información parcial de orientación en el espacio terrestre.

Para la fusión de datos de los sensores exoeceptivos con la navegación inercial usamos un *filtro de Kalman Extendido* (EKF). En este se asume un modelo de la forma:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (2)$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}, t) \quad (3)$$

donde la primera ecuación es el modelo dinámico utilizado para la estimación (ecuación de estados, donde \mathbf{x} es el vector de estados y \mathbf{u} las perturbaciones), y la segunda es un modelo para la relación entre el estado \mathbf{x} (que es lo que se estima) y las mediciones de los sensores

exoeceptivos \mathbf{y} .

Para la navegación inercial, el modelo (2) se corresponde con (1), siendo $\mathbf{x} = \{\mathbf{p}^e, \mathbf{v}^e, \mathbf{q}_b^e\}$; mientras que $\mathbf{u} = \{\mathbf{f}^b, \boldsymbol{\omega}_{eb}^b\}$ son medidas de forma directa por los sensores inerciales. El vector de salida \mathbf{y} son las mediciones de los sensores exoeceptivos.

El EKF consiste en computar una aproximación lineal en cada instante de muestreo para (2) y (3), y aplicar el algoritmo de Kalman [2] para el caso lineal. El modelo linealizado en tiempo discreto tiene la forma:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k [\mathbf{u}_k + \mathbf{w}_k] \\ \mathbf{y}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \end{aligned}$$

en donde \mathbf{u} es la parte conocida (o medida) de la perturbación, y \mathbf{w} la incertidumbre; mientras que \mathbf{v} es el ruido de medición. Linealizar y discretizar en el tiempo implica realizar los siguientes cálculos:

$$\begin{aligned} \mathbf{A}_k &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_k}, \quad \mathbf{B}_k = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{x}_k}, \quad \mathbf{H}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}_k} \\ \mathbf{F}_k &= \Phi_k(t_s) = \mathcal{L}^{-1} \left\{ [s\mathbf{I} - \mathbf{A}_k]^{-1} \right\} \Big|_{t=ts} \\ \mathbf{G}_k &= \int_0^{t_s} \Phi_k(t - \tau) \mathbf{B}_k d\tau \end{aligned}$$

En el presente caso se obtiene una predicción "a priori" \mathbf{x}_k^- (luego será corregida) a partir de la integración del modelo no-lineal en un instante de muestreo, para lo cual usamos el algoritmo adaptado de Savage [3, 4] por su eficiencia computacional.

Dado que la perturbación en este modelo es la fuerza específica y la velocidad angular, el ruido de proceso es el de la medición de estas perturbaciones. Para estimar la incertidumbre de esta predicción, en cada instante de integración debemos propagar la incertidumbre de la estimación precedente en el intervalo de integración, que depende de la dinámica, que representada por \mathbf{F} en el modelo lineal, y el impacto \mathbf{Q} del ruido de proceso sobre el resto de los estados:

$$\begin{aligned} \mathbf{Q}_k &= \mathbf{G}_{k-1} \mathbf{W} \mathbf{G}_{k-1}^T \\ \mathbf{P}_k^- &= \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_k \end{aligned}$$

donde \mathbf{W} es la covarianza de los ruidos de mediciones inerciales (acelerómetros y giróscopos).

La ecuación de salida queda definida por los sensores a fusionar, y puede cambiar de un instante de cómputo al siguiente en función de la disponibilidad de estos sensores. Por lo tanto será necesario computar la ganancia de Kalman en cada instante de muestreo con la ecuación:

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

computando \mathbf{H} para los sensores exoeceptivos disponibles en ese instante. Con esta ganancia corregimos la estimación a priori \mathbf{x}_k^- obtenida de la integración de las ecuaciones de navegación:

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k (\mathbf{y} - \hat{\mathbf{y}})^{-1}$$

donde $\hat{\mathbf{y}}$ es una predicción de lo que se debería medir según el modelo (3) a partir de la estimación a priori \mathbf{x}_k^- . Finalmente actualizamos la covarianza de la estimación corregida con la ecuación:

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

Medición y Movimiento de la Cámara

En [5] se presentó un sensor óptico basado en visión monocular. Este se basa en reconocer *puntos clave* en una imagen mediante el algoritmo ORB [6], y su identificación y seguimiento en imágenes sucesivas mediante el algoritmo de Lucas-Kanade [7] para computar el flujo óptico. En este trabajo, en lugar de computar el flujo óptico, utilizamos estas técnicas para determinar el movimiento de puntos fijos de la escena respecto de una imagen inicial, a fin de contar con mediciones de movimiento relativo de la cámara desde dicho instante. La medición es entonces la posición en la imagen de un punto, en principio, fijo en la superficie terrestre.

Asumamos conocidas las posiciones de un cierto conjunto de puntos identificados como puntos clave en la imagen. Asumamos que estos puntos están fijos en la terna- e .

Al moverse la cámara, para cada punto j , podemos re-calculer la proyección en el plano de la (nueva) escena para la estimación de la medición (modelo para la ecuación (3)):

$$\begin{Bmatrix} x_j^i \\ y_j^i \end{Bmatrix} = \frac{1}{z_j^b} \mathbf{F} \mathbf{C}_b^e \mathbf{C}_e^b \mathbf{d}_j^e, \quad \mathbf{d}_j^e = \mathbf{p}_j^e - \mathbf{c}^e$$

siendo \mathbf{C}_b^e la matriz de rotación que proyecta vectores en la terna- b (terna fija al vehículo) en la terna- e correspondiente a la orientación relativa de la cámara respecto de dicha terna, \mathbf{C}_e^b es la correspondiente a proyecciones de la terna terrestre sobre la terna del móvil, \mathbf{c}^e es la posición de la cámara en la terna- e , y:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

La posición de la cámara en la terna- e es $\mathbf{c}^e = \mathbf{p}^e + \mathbf{C}_b^e \mathbf{r}_{cam}^b$ donde \mathbf{r}_{cam}^b es su posición en terna- b . Con esto:

$$\begin{Bmatrix} x_j^i \\ y_j^i \end{Bmatrix} = \frac{1}{z_j^b} \mathbf{F} \mathbf{C}_b^e \mathbf{C}_e^b (\mathbf{p}_j^e - \mathbf{p}^e) - \frac{1}{z_j^b} \mathbf{F} \mathbf{C}_b^e \mathbf{r}_{cam}^b, \quad z_j^b = \mathbf{C}_b^e \mathbf{d}_j^e \hat{\mathbf{k}}$$

Derivando esta expresión respecto de los estados tendremos la matriz jacobiana para la fusión:

$$\mathbf{H}_i = [\mathbf{H}_{iq} \quad \mathbf{H}_{ip} \quad \mathbf{0}^{3 \times 3}]$$

donde

$$\mathbf{H}_{iq} = -\frac{1}{z_j^b} \mathbf{F} \mathbf{C}_b^e \frac{\partial \mathbf{C}_b^e}{\partial \mathbf{q}} \mathbf{d}_j^e, \quad \mathbf{H}_{ip} = -\frac{1}{z_j^b} \mathbf{F} \mathbf{C}_b^e \mathbf{C}_e^b - \frac{\partial z_j^b}{\partial \mathbf{p}^e} \mathbf{F} \mathbf{C}_b^e \mathbf{C}_e^b \mathbf{d}_j^e$$

Teniendo en cuenta la relación entre matriz de rotación y cuaternión, y la norma unitaria del cuaternión de actitud, podemos plantear:

$$\mathbf{C}_e^b = 2 \begin{bmatrix} \frac{1}{2} - y^2 - z^2 & xy + \eta z & xz - \eta y \\ xy - \eta z & \frac{1}{2} - x^2 - z^2 & yz + \eta x \\ xz + \eta y & yz - \eta x & \frac{1}{2} - x^2 - y^2 \end{bmatrix}$$

de lo cual:

$$\frac{\partial \mathbf{C}_e^b}{\partial \eta} = \mathbf{C}_\eta = 2 \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix}, \quad \frac{\partial \mathbf{C}_e^b}{\partial x} = \mathbf{C}_x = 2 \begin{bmatrix} 0 & y & z \\ y & -2x & \eta \\ z & -\eta & -2x \end{bmatrix}$$

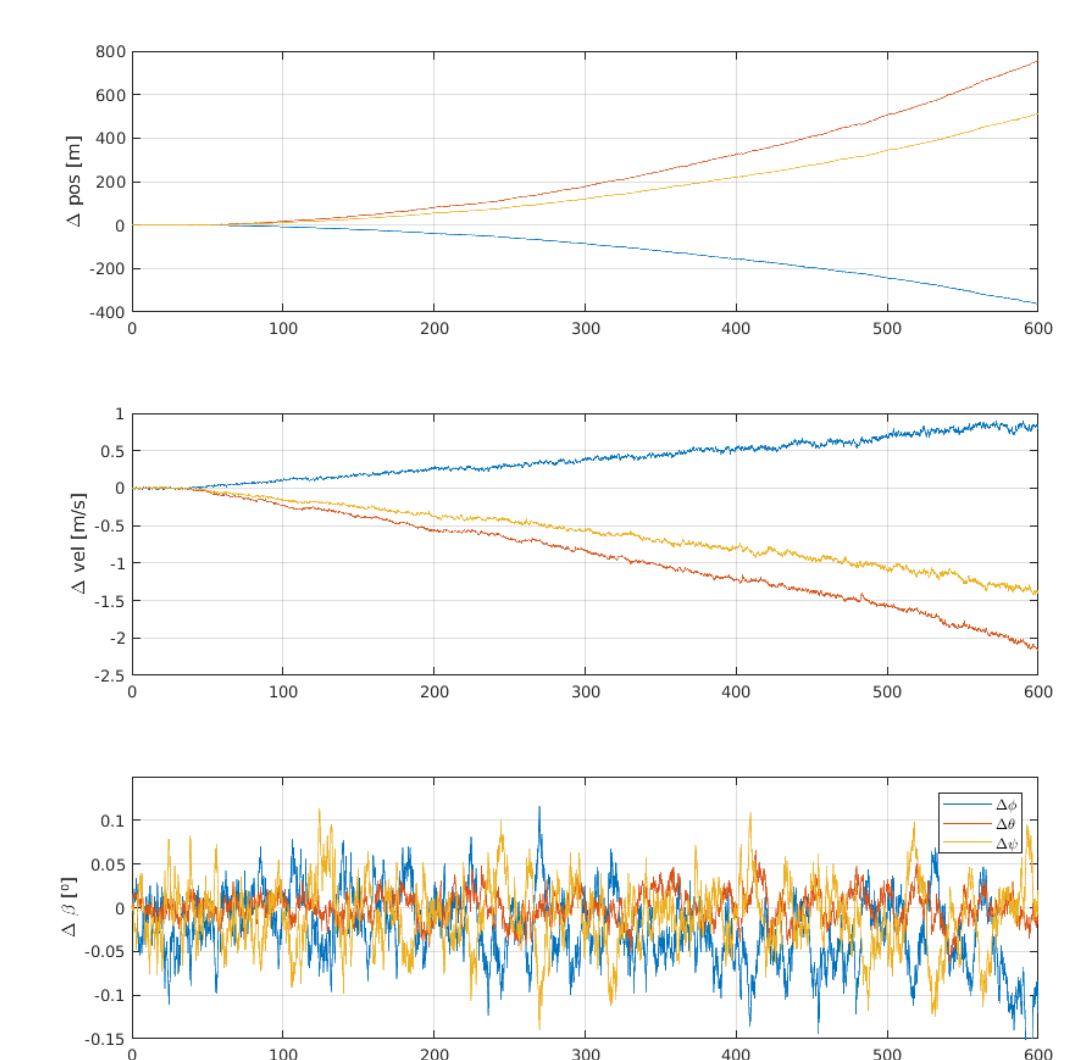
$$\frac{\partial \mathbf{C}_e^b}{\partial y} = \mathbf{C}_y = 2 \begin{bmatrix} -2y & x & -\eta \\ x & 0 & z \\ \eta & z & -2y \end{bmatrix}, \quad \frac{\partial \mathbf{C}_e^b}{\partial z} = \mathbf{C}_z = 2 \begin{bmatrix} -2z & \eta & x \\ -\eta & -2z & y \\ x & y & 0 \end{bmatrix}$$

Resultados y Conclusiones

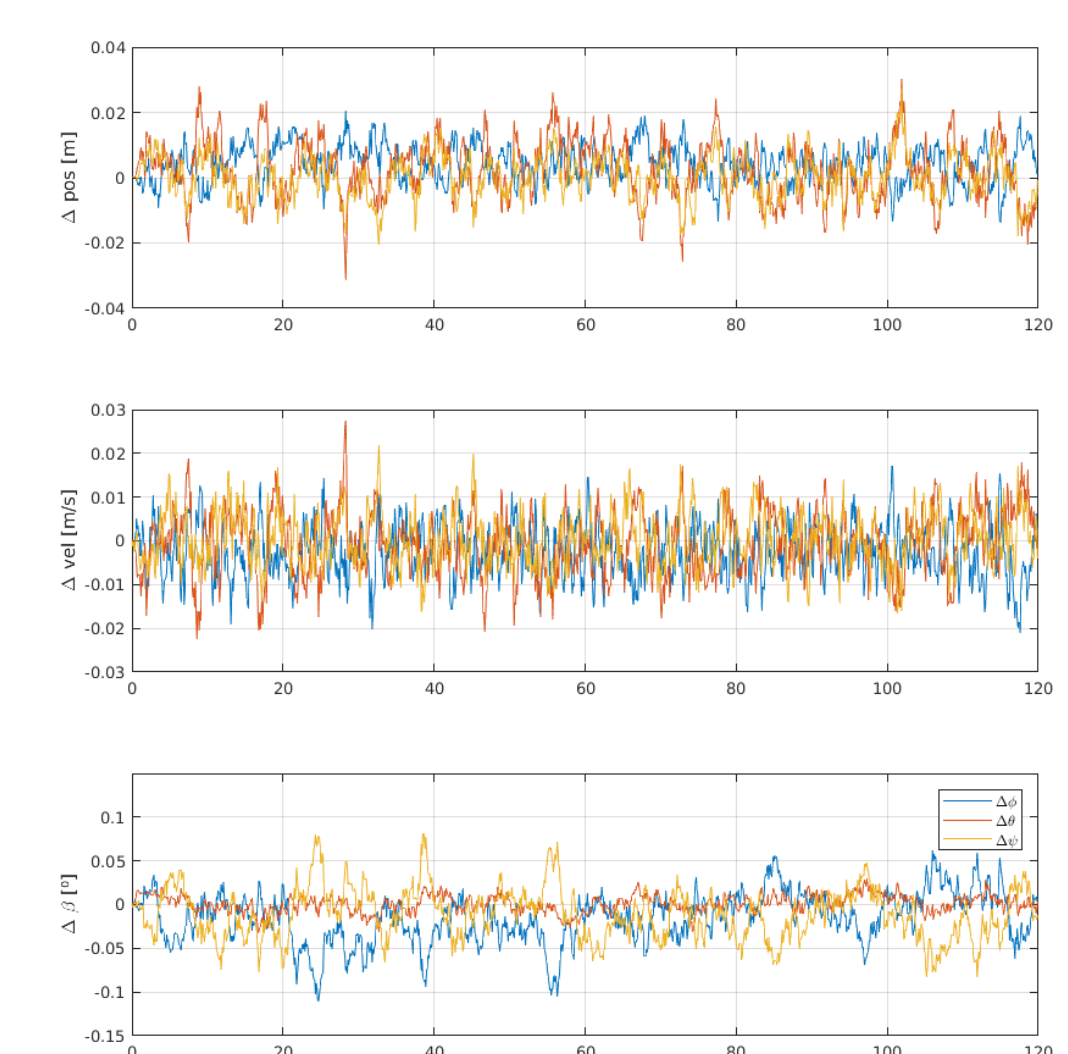
Implementamos estos cálculos en scripts de Octave/Matlab, y simulamos las mediciones de los sensores introceptivos y exoeceptivos para un vehículo estacionario, introduciendo ruidos de medición simulados comparables con sensores MEMS comerciales y cámara monocular VGA.

En primer término fusionamos la navegación inercial con el sensor visual, obteniendo los resultados mostrados en esta figura:

Vemos que se estabiliza la estimación de actitud y mejora sustancialmente la de posición, aunque en esta última estimación la inestabilidad persiste.



Esto queda resuelto tan solo agregando un sensor de altura (barómetro o telémetro), sin requerir datos del receptor GNSS, como se observa en estos gráfico.



Concluimos que la navegación integrada con un sensor visual permite estabilizar la estimación de actitud, y reduce la divergencia en la estimación de posición.

Incluyendo adicionalmente un sensor de altura se estabiliza la estimación completa.

BIBLIOGRAFIA

- [1] M. España, *Sistemas de Navegación Integrada con Aplicaciones*. CONAE, 2 ed., 2016.
- [2] G. Welch and G. Bishop, "An introduction to the kalman filter," tech. rep., Department of Computer Science, University of North Carolina at Chapel Hill, 2004.
- [3] P. G. Savage, "Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 19-28, jan, feb 1998.
- [4] P. G. Savage, "Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 208-221, mar 1998.
- [5] M. P. Concia Bernardo, Zumarraga Augusto, "Caracterización de sensor de flujo óptico para aplicación en navegación integrada," in *5^{as} Jornadas de Investigación, Transferencia y Extensión - FI - UNLP*, 2019.
- [6] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf," *2011 International Conference on Computer Vision*, pp. 2564-2571, 2011.
- [7] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, (San Francisco, CA, USA), pp. 674-679, Morgan Kaufmann Publishers Inc., 1981.