

Double Coupling Between Inertial Sensors and Visual Odometry in Multicopters

Claudio J. Paz

Centro de Investigación en Informática para la Ingeniería
Universidad Tecnológica Nacional, Facultad Regional Córdoba
Córdoba, Argentina
Email: cpaz@frc.utn.edu.ar

Sergio Nesmachnow

Centro de Cálculo, Instituto de Computación
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
Email: sergion@fing.edu.uy

Abstract—This article presents a system for determining the position and orientation of unmanned aerial vehicles. The system has two hardware layers coupled in both directions. One layer stabilizes the vehicle while the other determines lateral displacements. The two-directional coupling between layers allows the system to minimize estimation errors. The proposed implementation is evaluated on real hardware and evaluation results show that a real time implementation is feasible.

Keywords—Unmanned Aerial Vehicles, Visual Odometry, Inertial Sensors, Extended Kalman Filter.

I. INTRODUCCIÓN

Los vehículos aéreos no tripulados (VANT o UAV, acrónimo de *Unmanned Aerial Vehicles*), y en particular los multicopteros, han tenido un crecimiento exponencial en la última década debido a su relativo bajo costo de construcción y a los numerosos recursos de *software* y *hardware* disponible [1]. Su capacidad de volar manteniendo su posición en un mismo punto, maniobra conocida como *hovering*, junto con vuelos de baja velocidad de desplazamiento, le permiten tomar imágenes que pueden ser utilizadas en diversas aplicaciones, por ejemplo fotografía deportiva, monitoreo ambiental [2], inspección de zonas peligrosas o de desastres [3], relevamiento de cultivos y otro tipo de terrenos [4], etc.

Los multicopteros cuentan con distintos tipos de sensores, incluyendo acelerómetros y giróscopos, conocidos como sensores inerciales, que permiten su estabilización en el aire [5]; telémetros para determinar la distancia del vehículo hasta el suelo [6]; y suelen utilizar cámaras de bajo costo para determinar los desplazamientos laterales, generalmente utilizando detectores de características y flujo óptico [7].

Debido a que los multicopteros tienen sus hélices fijas, para desplazarse deben perder balance en la dirección que se desea el movimiento. En ese momento, el telémetro que se utiliza para determinar la altitud pierde la perpendicularidad de su eje sensible respecto del plano formado por el suelo, por lo que su lectura es errónea si se considera directamente como altura. En esta situación se produce una pérdida de sustentación por dos motivos: el primero tiene que ver con la pérdida de empuje en sentido vertical ya que para una misma fuerza de empuje aparece una componente horizontal para producir el desplazamiento lateral, disminuyendo el módulo

de la componente vertical; el segundo es que, para evitar esta pérdida se utiliza información proveniente del sensor de altura, que devuelve una lectura mayor a la distancia real al suelo por lo que el controlador disminuye el empuje para compensar este incremento en la distancia. De compensarse la altura en el segundo caso, indirectamente se evitaría la pérdida de sustentación por el primero.

La Figura 1 presenta el esquema de fusión propuesto para determinar posición y orientación. Se identifican los dos bloques más importantes del esquema propuesto: el autopiloto y el módulo de odometría visual. En el autopiloto, por medio de un filtro de Kalman extendido (*Extended Kalman filter*, EKF), se determina la orientación del vehículo respecto del plano formado por el suelo usando acelerómetros y giróscopos. De esta manera, es posible estabilizar el vehículo para un vuelo estacionario. También se puede determinar la altura a partir de un sensor de rango o telémetro que mida la distancia del vehículo al suelo. Por otro lado, el módulo de odometría visual determina el desplazamiento respecto de los ejes coordenados x e y , y además el ángulo de rotación alrededor del eje z , llamado ángulo de guiñada, o simplemente ψ .

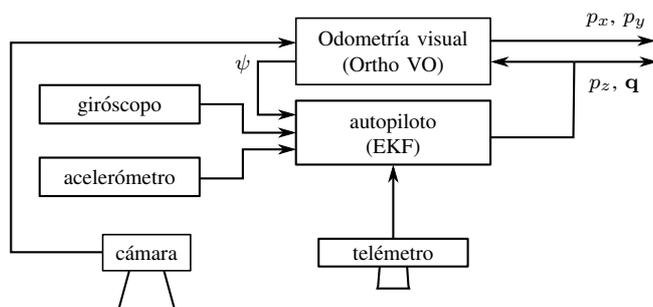


Figura 1. Esquema para la estimación de posición y orientación.

El esquema de la Figura 1 muestra que los módulos están acoplados: la información de altura y orientación, representados con p_z y por el cuaternión unitario \mathbf{q} respectivamente, se utiliza en el módulo de visión para procesar las imágenes y realizar la odometría visual. Existe también un acoplamiento en el otro sentido, ya que el ángulo de guiñada ψ se utiliza en la estimación de la orientación. El doble acoplamiento entre ambos módulos permite minimizar los errores de estimación.

El acoplamiento usando el ángulo ψ minimiza los errores en la orientación debidos a la integración de sesgos en las mediciones de los giróscopos. Por otro lado, el acoplamiento usando p_z y q minimiza los errores en la odometría visual debidos a la errónea generalización de la escala en la imagen.

La principal contribución de este artículo consiste en una propuesta para una estimación con doble acoplamiento entre los módulos de odometría visual y autopiloto para el control de vuelo de multicopteros. Los principales resultados demuestran que utilizando el sistema propuesto se obtienen mejoras en la estimación de posición y orientación respecto a los métodos tradicionales. Adicionalmente, el análisis de eficiencia computacional indica que las mejoras no implican demoras adicionales en el tiempo de cómputo, y el proceso de estimación puede realizarse en tiempo real.

El artículo se estructura del modo que se describe a continuación. La Sección II presenta los aspectos teóricos de los dos principales módulos del sistema propuesto: EKF y el módulo visual. La Sección III describe en detalle la implementación del sistema propuesto. La evaluación del sistema propuesto utilizando simulaciones y plataformas reales se describe en la Sección IV. Finalmente, la Sección V presenta las conclusiones del trabajo y las principales líneas de trabajo futuro.

II. MÉTODOS CONSIDERADOS EN EL ESTUDIO

Esta sección presenta los conceptos teóricos en los que se basan los módulos EKF y de odometría visual. Los valores estimados utilizando ambos módulos completan un único vector de estados. Los valores calculados por el módulo EKF son probabilísticos de distribución gaussiana, mientras que los valores obtenidos por la odometría visual son completamente determinísticos, obtenidos de forma cerrada geoméricamente.

II-A. Filtro de Kalman Extendido

Un sistema, planta o proceso puede ser modelado por una sucesión de variables de estado $\{\mathbf{x}_k : k \in \mathbb{N}_0\} \in \mathbb{R}^n$ donde n es la dimensión del vector de estados del sistema y k representa la sucesión de instantes temporales. La evolución de las variables de estado entre instantes de tiempo se puede modelar como $\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{w}_{k-1})$. La función de transición del sistema $f_k(\cdot)$ puede depender del instante k y \mathbf{w}_{k-1} es un ruido blanco con media cero y covarianza conocida que puede escribirse como $E\{\mathbf{w}_{k-1}\} = 0$ y $E\{\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T\} = Q_{k-1}$, en donde $\mathbf{w}_{k-1} \in \mathbb{R}^n$ y $Q_{k-1} \in \mathbb{R}^{n \times n}$.

En general, el estado del sistema no puede ser observado de forma directa sino a través de mediciones parciales y posiblemente perturbadas por ruido de diversas fuentes. Se puede describir esta situación modelando la relación entre el estado del sistema y la sucesión de observaciones $\{\mathbf{z}_k : k \in \mathbb{N}_0\} \in \mathbb{R}^m$ como $\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{v}_k)$, donde la función de observación $h_k(\cdot)$ también puede depender del instante k , por ejemplo en sistemas con sensores de múltiples tasas de muestreo. El ruido \mathbf{v}_k también es ruido blanco con media cero y covarianza conocida $E\{\mathbf{v}_{k-1}\} = 0$ y $E\{\mathbf{v}_{k-1}\mathbf{v}_{k-1}^T\} = R_{k-1}$, en donde $\mathbf{v}_{k-1} \in \mathbb{R}^m$ y $R_{k-1} \in \mathbb{R}^{m \times m}$, siendo m la dimensión del vector de observación.

Si el sistema bajo análisis se puede describir mediante $\mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}$ y $\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k$, siendo $f_k(\cdot)$ y $h_k(\cdot)$ funciones no lineales diferenciables, es posible aproximar las funciones mediante una serie de Taylor (genéricamente tanto para f_k como para h_k) $f(\mathbf{x}) = f(\hat{\mathbf{x}}) + Jf(\hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}}) + \mathcal{O}(|\mathbf{x} - \hat{\mathbf{x}}|^2)$, donde $Jf(\hat{\mathbf{x}})$ es el jacobiano de la función f , que se calcula diferenciando la función respecto del estado estimado.

Si en el instante k se cuenta con la estimación $\hat{\mathbf{x}}_{k-1|k-1}$ (media de la pdf $p(\mathbf{x}_{k-1}|\mathbf{x}_{k-1})$ en el tiempo $k-1$) y su covarianza $P_{k-1|k-1}$, se puede aproximar \mathbf{x}_k mediante una serie de Taylor de f_{k-1} de primer orden alrededor de $\hat{\mathbf{x}}_{k-1|k-1}$ como $\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1} + \hat{F}_{k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}) + \mathbf{w}_{k-1}$, en donde $\hat{\mathbf{x}}_{k|k-1} = f_{k-1}(\hat{\mathbf{x}}_{k-1|k-1})$ y $\hat{F}_{k-1} = Jf_{k-1}(\hat{\mathbf{x}}_{k-1|k-1})$.

Siguiendo un desarrollo similar al desarrollo para el filtro de Kalman [8] se pueden obtener las ecuaciones para la etapa de predicción:

$$\hat{\mathbf{x}}_{k|k-1} = f_{k-1}(\hat{\mathbf{x}}_{k-1|k-1}), \quad (1)$$

$$\hat{F}_{k-1} = Jf_{k-1}(\hat{\mathbf{x}}_{k-1|k-1}) = \left. \frac{\partial f_{k-1}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1}}, \quad (2)$$

$$P_{k|k-1} = Q_{k-1} + \hat{F}_{k-1}P_{k-1|k-1}\hat{F}_{k-1}^T. \quad (3)$$

Análogamente, para la etapa de actualización se tiene:

$$\hat{\mathbf{z}}_{k|k-1} = h_k(\hat{\mathbf{x}}_{k|k-1}), \quad (4)$$

$$\hat{H}_k = Jh_k(\hat{\mathbf{x}}_{k|k-1}) = \left. \frac{\partial h_k}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}, \quad (5)$$

$$K_k = P_{k|k-1}\hat{H}_k^T(R_k + \hat{H}_kP_{k|k-1}\hat{H}_k^T)^{-1}, \quad (6)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}), \quad (7)$$

$$P_{k|k} = (I - K_k\hat{H}_k)P_{k|k-1}, \quad (8)$$

siempre que las funciones f y h sean diferenciables, y mientras que las condiciones iniciales $\hat{\mathbf{x}}_{0|0}$ y $P_{0|0}$ sean conocidas.

II-B. Odometría Visual

II-B1. Pose a partir de información visual: Se conoce como odometría visual a la estimación de la posición y orientación, también conocida como pose, a partir de la integración de información provista por una cámara. En robótica móvil, el término odometría se asocia generalmente a la integración sucesiva de los giros de las ruedas de los vehículos. En cambio, la odometría visual estima la posición y orientación del vehículo a partir de posiciones relativas entre las distintas capturas de una o múltiples cámaras. Las ventajas de la odometría visual frente a la tradicional es que se evitan los errores no sistemáticos de la última, debido principalmente al tránsito sobre terreno desparejo o sobre obstáculos inesperados, deslizamiento de las ruedas, entre otros. La expresión *odometría visual* fue usada por primera vez por Nister et al. [9], quienes mostraron los resultados del uso de cámaras estéreo para determinar la posición y orientación de un vehículo autónomo terrestre. Sin embargo, la estimación de la posición y la orientación a partir de la información proveniente de las cámaras tiene sus orígenes a finales de la década de 1970, motivada por la exploración marciana [10].

El método asume conocidas por un lado la posición y la orientación relativa entre dos capturas de una cámara, $t_{k,k-1}$ y $R_{k,k-1}$ respectivamente, y por otro la posición y la orientación de la captura en el instante de tiempo $k-1$ respecto a un sistema de referencia n , $t_{n,k-1}$ y $R_{n,k-1}$. Cumplidas estas condiciones, se conoce como odometría visual al proceso de integración mediante el cual se puede conocer en cada instante k el valor de la posición y la orientación $t_{n,k}$ y $R_{n,k}$.

La Figura 2 presenta un esquema de tres capturas consecutivas de una misma cámara en tres momentos diferentes, en tres tonos diferentes de azul (el más oscuro representa la captura más reciente). El origen del sistema de referencia de navegación se representa por el triedro de color negro. Las flechas desde el origen hasta cada captura representan la posición y orientación de la cámara en cada instante respecto del sistema de referencia. Los distintos tonos de rojo (el más oscuro es el más reciente) muestran la posición y orientación entre dos vistas consecutivas, respecto de la más reciente.

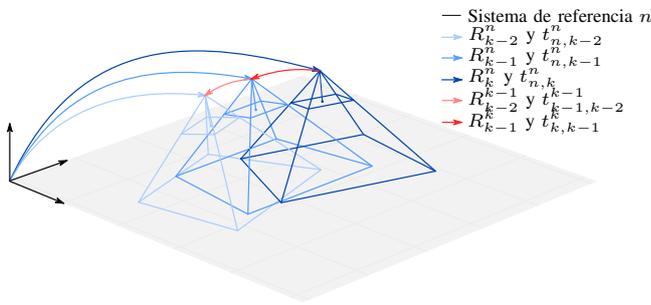


Figura 2. Composición gráfica de la posición y orientación de una cámara.

El diagrama de la Figura 2 muestra que para obtener R_k^n y $t_{n,k}^n$ se puede componer una traslación y una rotación usando las flechas. Por ejemplo, en un tiempo k se cuenta con la estimación de posición y orientación obtenida en $k-1$, esto es R_{k-1}^n y $t_{n,k-1}^n$. En ese momento, el algoritmo visual devuelve R_{k-1}^k y $t_{k-1,k-1}^k$. La composición de la rotación se puede realizar considerando el gráfico, teniendo en cuenta el sentido de las flechas. En el caso de que al seguir una flecha se encuentre una con sentido contrario, en la composición se debe agregar esta última pero invertida. Entonces, siguiendo R_{k-1}^n al encontrarse con R_{k-1}^k se debe invertir la última para componer R_k^n mediante

$$R_k^n = R_{k-1}^n R_{k-1}^k{}^T. \quad (9)$$

También se puede operar directamente teniendo en cuenta los subíndices y superíndices, los cuales indican el sistema de coordenadas en cuestión y el de referencia, respectivamente.

Para el caso de la traslación el mecanismo es similar, pero no solo teniendo en cuenta el sentido de los vectores que componen el vector buscado, sino también en el sistema de coordenadas que se expresan. Por ejemplo, el módulo de visión devuelve el vector que une los orígenes de los sistemas de coordenadas de la cámara en k y $k-1$, pero referido al sistema de la cámara en k . Sin embargo, para sumarlo a la posición estimada en el sistema de referencia de navegación

hasta el instante $k-1$, primero hay que referirlo a ese sistema. Además, dado que las flechas están en sentido contrario hay que invertirlos, lo que en traslaciones solo significa un cambio de signo. Entonces, el vector $t_{k,k-1}^k$ es expresado en n es $t_{k,k-1}^n = R_k^n t_{k,k-1}^k$ por lo que la composición resulta

$$t_{n,k}^n = t_{n,k-1}^n - t_{k,k-1}^n. \quad (10)$$

En el Algoritmo 1 se resumen los pasos para obtener la odometría visual a partir de imágenes consecutivas, suponiendo que se conoce la altura h de la cámara en el tiempo $k-1$ y la matriz de parámetros intrínsecos P .

Algoritmo 1 Estimación de la posición y la orientación mediante odometría visual

```

function VISUALODOMETRY( $i_k, i_{k-1}$ )
  Inicializar  $R_0^n$  y  $t_{n,0}^n$ 
  for ever do
     $R_{k-1}^k, \mathbf{n}, T/d \leftarrow$  POSEESTIMATION( $i_k, i_{k-1}$ )
     $t_{k-1}^k = P^{-1}T/d * h$ 
     $R_k^n = R_{k-1}^n R_{k-1}^k{}^T$ 
     $t_{n,k}^n = t_{n,k-1}^n - R_k^n t_{k,k-1}^k$ 
  end for
end function

```

II-B2. Detección y seguimiento de características salientes de la imagen: El movimiento relativo entre la cámara y el mundo tridimensional en el que se encuentra se puede determinar indirectamente a partir del movimiento de puntos salientes en la imagen entre una captura y la siguiente.

Estos puntos son tratados en la literatura como puntos de interés o características, definidos como cualquier punto en la imagen en el cual la información que aporte cambie en cualquier dirección. Un grupo de píxeles formando un borde con forma de L, T o Y pueden ajustarse a esta definición, pero también si forman un punto en una región con cierto nivel de contraste. La literatura identifica diversos tipos de detectores de características, separados en tres categorías: los de ajuste por modelo paramétrico, los basados en bordes o contornos y los basados en la intensidad [11].

En este trabajo se utilizan detectores de características basados en la intensidad del punto de interés. Para determinar la localización puntual de una característica se debe evaluar la variación de intensidad de la zona de interés buscando un valor que sea máximo en todas direcciones. El cambio de intensidad puede evaluarse a partir de los de autovalores y autovectores de una matriz formada por los gradientes de la zona a evaluar [12] o simplemente utilizando el mínimo de los autovalores para determinar la calidad de la característica [13].

Luego de determinar la posición de una característica en la imagen, se debe encontrar la nueva posición de la misma en una imagen posterior, para hallar el desplazamiento relativo entre la cámara que toma la imagen y la característica. Este seguimiento de la característica de una imagen a otra se conoce como *flujo óptico*. Barron et al. [14] presentan un análisis de los diferentes tipos de flujo óptico con sus ventajas y desventajas, incluyendo al método Lucas-Kanade [15] considerado el estándar de oro hasta la actualidad.

II-B3. Composición y descomposición de la matriz de homografía: La homografía es una transformación proyectiva que determina la correspondencia entre dos conjuntos de puntos $\{\mathbf{x}_{i k-1}\}$ y $\{\mathbf{x}_{i k}\}$ para $i \geq 4$, pudiéndose expresar $\mathbf{x}_{i k} = H \mathbf{x}_{i k-1}$ cuando los puntos son coplanares.

Entonces, utilizando los puntos salientes de la imagen capturada en el tiempo $k-1$ y los puntos obtenidos por el flujo óptico en k , se puede determinar la matriz de homografía utilizando el algoritmo de transformación lineal directa [16] (DLT por las siglas en inglés de *Direct Linear Transformation*) como se desarrolla en [17].

Una vez obtenida la matriz H , se puede aplicar una descomposición para determinar la rotación y la traslación que sufre la cámara entre las dos imágenes que dieron lugar a la propia homografía. Se puede realizar esta descomposición mediante el algoritmo presentado en [18] y [19] en donde además de obtener la matriz R_{k-1}^k y el vector t_{k-1}^k , se obtiene el vector normal al plano donde se encuentran los puntos que se usan para formar la homografía.

En resumen, en este trabajo se utiliza el método de detección de características de Shi-Tomasi, descrito en [13] para determinar el conjunto de puntos para el tiempo $k-1$ y el flujo óptico de Lucas-Kanade desarrollado en [15], para encontrar la nueva posición de estas características en el tiempo k . Luego, con el algoritmo DLT se determina la matriz de homografía la cual es descompuesta en una matriz de rotación y un vector de traslación que relaciona las dos vistas. En el Algoritmo 2 se describe este proceso. Posteriormente a partir de esta matriz de rotación y vector de traslación R_{k-1}^k y t_{k-1}^k , respectivamente, se completa la odometría visual utilizando las ecuaciones (9) y (10), como se muestra en el Algoritmo 1.

Algoritmo 2 Estimación de la posición y la orientación relativa entre dos imágenes consecutivas

```

function POSEESTIMATION( $i_t, i_{t-1}$ )
   $\mathbf{x}_{i t-1} \leftarrow$  FEATURE_DETECTOR( $i_{t-1}$ )
   $\mathbf{x}_{i t} \leftarrow$  OPTICAL_FLOW( $i_t, i_{t-1}, \mathbf{x}_{i t-1}$ )
   $H_\lambda \leftarrow$  FINDHOMOGRAPHY( $\mathbf{x}_{i t}, \mathbf{x}_{i t-1}$ )
   $R, \mathbf{n}, \mathbf{T}/d \leftarrow$  GETRTN( $H_\lambda$ )
  return  $R, \mathbf{n}, \mathbf{T}/d$ 
end function

```

III. IMPLEMENTACIÓN DEL SISTEMA PROPUESTO

Esta sección presentan los detalles de implementación del sistema propuesto para minimizar los errores en la estimación de la posición y la orientación.

III-A. Descripción general

El sistema propuesto está formado por dos etapas principales. Primero un EKF con un vector de estados de cinco dimensiones para determinar la orientación y la altura lo que permite al vehículo mantener un vuelo estabilizado, ya sea estacionario o no. En una capa de hardware diferente, se cuenta con un módulo encargado de realizar la odometría visual mediante detección de puntos característicos en cada imagen, el flujo óptico y la homografía.

El vínculo entre estas dos etapas, que en este trabajo se denomina *doble acoplamiento*, se debe a que el EKF, implementado con acelerómetros y giróscopos, en una dirección utiliza información del módulo de odometría visual para determinar la rotación alrededor del eje z del vehículo. Además, completando el acoplamiento en el otro sentido, el EKF provee al módulo visual de información de orientación y altura. Con esta información es posible quitar la distorsión perspectiva a las imágenes capturadas y corregir el error inducido.

En la subsección III-B se trata el problema de la estimación de la orientación y el problema de la determinación de la altura con un telémetro de forma integrada. Esto es necesario debido a la relación que existe entre la lectura del telémetro (sonar, láser, infrarrojo, etc.) y la orientación del sistema de referencia del cuerpo al que está acoplado. Debido a que con las lecturas del acelerómetro y el giróscopo no se puede determinar el ángulo de guiñada, se utiliza información del módulo de visión para corregirlo.

En la subsección III-C se presenta la propuesta de quitar la distorsión perspectiva de modo que las imágenes analizadas sean coplanares, lo que minimiza los errores debidos al error de escala sin aumentar significativamente el tiempo de computo.

III-B. Filtro de Kalman extendido para determinar orientación y altura

Se plantea utilizar un filtro de Kalman extendido donde, como se sugiere al inicio de la sección, el vector de estados tiene cinco dimensiones $\mathbf{x} = [q_0 \ q_1 \ q_2 \ q_3 \ p_z]^T$, los cuatro escalares del cuaternión que representa la orientación más un escalar para la altura. Los principales detalles se presentan a continuación

III-B1. Predicción: Modelos cinemático de cuaternión y de altura constante: Para determinar la matriz de transición de estado, del tiempo $k-1$ al k , se puede separar la etapa de predicción en dos partes usando distintos modelos. Por un lado, partiendo de la derivada temporal del cuaternión unitario

$$\dot{\mathbf{q}} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}) \mathbf{q} \quad (11)$$

desarrollada en [20] junto con su versión discreta

$$\mathbf{q}_k = \left(\frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}_k) \Delta t + I \right) \mathbf{q}_{k-1} \quad (12)$$

para obtener la matriz de transición vinculada al cuaternión

$$F_{\mathbf{q}} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}_k) \Delta t + I \quad (13)$$

con

$$\boldsymbol{\Omega}(\boldsymbol{\omega}_k) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}. \quad (14)$$

Por otro lado, se considera que el telémetro se actualiza a mayor velocidad de lo que varía la altura, por lo que se trabaja bajo la hipótesis de altura constante ($\dot{h} = 0$). La altura se discretiza mediante $h_k = F_s h_{k-1}$ con $F_s = 1$.

La etapa de predicción queda formada entonces por la ecuación de proceso $\mathbf{x}_k = F\mathbf{x}_{k-1}$ donde

$$F = \begin{bmatrix} F_q & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & F_s \end{bmatrix} \quad (15)$$

y por la actualización de la matriz de covarianza del estado estimado $P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q$ con $Q = E[\mathbf{w}\mathbf{w}^T]$. como la matriz de covarianza del ruido de proceso \mathbf{w} , el cual está relacionado con el ruido del giróscopo dado su efecto en el cuaternión que forma el vector de estado.

III-B2. Actualización: modelos de observación del acelerómetro, el ángulo de guiñada y el telémetro: La etapa de actualización se divide en tres partes: dos para corregir o actualizar el cuaternión y la restante para actualizar la altura de vuelo. Para la actualización del cuaternión se utiliza la medición dada por el acelerómetro o el ángulo ψ , mientras que para la actualización de la altura se utiliza el telémetro.

Cuando el VANT está en vuelo estacionario, o en vuelos con velocidad constante como por ejemplo en vuelos de reconocimiento, las mediciones del acelerómetro son útiles para determinar la inclinación del vehículo respecto del suelo. Esto se debe a que a pesar de su nombre, los acelerómetros no miden aceleraciones, sino que en realidad miden las fuerzas específicas actuando sobre el sensor. Por ejemplo, si se deja un acelerómetro en reposo sobre una superficie estática, se puede decir que el sensor no está sometido a ninguna aceleración, sin embargo la lectura en este caso es de 1g en el eje vertical debido a la fuerza ficticia de la superficie que evita la caída libre del sensor. Se puede utilizar este fenómeno para modelar la respuesta del acelerómetro en estas condiciones en función de la orientación. En importante destacar que la respuesta del acelerómetro es independiente del ángulo de guiñada ya que cambios en este ángulo no afectan la verticalidad del sensor.

Si se considera al vector gravedad en el sistema de referencia de navegación $\mathbf{g}^n = [0 \ 0 \ g]^T$, siendo g la aceleración debida a la gravedad (se aproxima a 9.81m/s²) desde el sistema de referencia del sensor la lectura ideal del acelerómetro será $\mathbf{g}^b = R_n^b \mathbf{g}^n$, siendo \mathbf{g}^b la acción de la gravedad referida al sistema de referencia del VANT.

La matriz de cambio de marco de referencia R_n^b en función de un cuaternión unitario $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$ es

$$R_n^b = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_3q_0) & 2(q_1q_3 - q_2q_0) \\ 2(q_1q_2 - q_3q_0) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_1q_0) \\ 2(q_1q_3 + q_2q_0) & 2(q_2q_3 - q_1q_0) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

por lo que la definición de \mathbf{g}^b se simplifica a

$$\mathbf{g}^b = g \begin{bmatrix} 2(q_1q_3 - q_2q_0) \\ 2(q_2q_3 + q_1q_0) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (16)$$

debido a que el vector gravedad \mathbf{g}^n solo tiene la componente en el eje z distinta de cero. La ecuación (16) es considerada como modelo de observación del acelerómetro a bordo del VANT respecto del cuaternión unitario que representa la orientación del vehículo.

En cuanto al ángulo de guiñada, si se cuenta un módulo de visión como en [21] se podrían corregir los errores de

orientación dados por el giróscopo para complementar las lecturas del acelerómetro.

Dado el cuaternión unitario \mathbf{q} con componentes $[q_0, q_1, q_2, q_3]$, que representa la orientación de un cuerpo rígido, se puede demostrar [22] que el ángulo ψ de la representación en ángulos de Euler de la orientación del mismo cuerpo es

$$\psi = \arctan \left(\frac{2(q_0q_3 + q_1q_2)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \right) \quad (17)$$

Para fines prácticos, no es posible utilizar la relación (17), ya que presentaría discontinuidades cuando el vehículo realice un giro completo. Este no es un simple problema de representación de la orientación; puede ser un problema mayor cuando se calcula la innovación en un filtro de Kalman, ya que el valor de corrección puede ser tan grande como 2π en el límite del rango cuando en realidad es de un diferencial de ángulo.

Por este motivo, en lugar de usar el ángulo ψ para corregir el cuaternión, se propone utilizar un vector ampliado ψ_a (ecuación (18)) que si bien aumenta el tamaño del problema, evita la aparición de discontinuidades.

$$\psi_a = \begin{bmatrix} \cos \psi \\ \sin \psi \end{bmatrix} \quad (18)$$

Además, utilizando las identidades trigonométricas $\cos(\arctan(\gamma)) = \frac{1}{\sqrt{\gamma^2 + 1}}$, $\sin(\arctan(\gamma)) = \frac{\gamma}{\sqrt{\gamma^2 + 1}}$ y $\gamma = \frac{2(q_0q_3 + q_1q_2)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}$, la ecuación (18) resulta

$$\psi_a = \begin{bmatrix} \frac{1}{\sqrt{\gamma^2 + 1}} \\ \frac{\gamma}{\sqrt{\gamma^2 + 1}} \end{bmatrix} \quad (19)$$

La ecuación (19) es la considerada para modelar la observación del ángulo ψ en función del cuaternión unitario que representa la orientación del VANT.

En el caso del telémetro, como se mencionó anteriormente, cuando su eje de sensibilidad no forma una perpendicular con el plano que forma el obstáculo, su lectura no se corresponderá con la distancia a la que se encuentra el sensor. Este efecto se muestra (ampliado) en la Figura 3 donde se ve que la distancia medida por el telémetro será mayor en magnitud que la distancia que lo separa del plano y dependerá de la orientación del sistema de referencia del sensor además de la distancia.

Se puede modelar la distancia medida por un telémetro, s^s , como la distancia entre dos puntos dada por la norma del vector formado con dichos puntos de modo que $s^s = \|\mathbf{p}_i - \mathbf{p}_s\|$, donde \mathbf{p}_s es el origen de coordenadas del sistema de referencia del telémetro y \mathbf{p}_i es el punto de intersección entre la recta L definida por el eje de sensibilidad y el plano Π representando el piso.

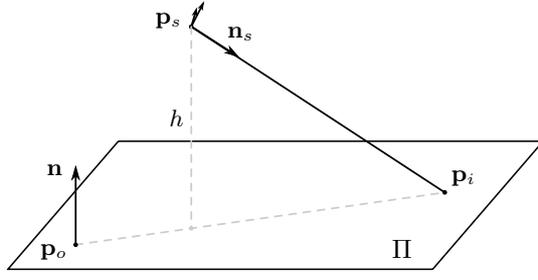


Figura 3. Modelo de observación de la altura usando un telémetro.

Encontrando la intersección de la recta L , definida por el punto \mathbf{p}_s y el vector \mathbf{n}_s , y el plano Π se puede expresar a la lectura del telémetro como la norma del segmento s

$$\begin{aligned} s &= \|\mathbf{s}\| = \sqrt{s_i^2_x + s_i^2_y + s_i^2_z} \\ &= \frac{p_{s_z}^n}{n_{s_z}^n} \sqrt{(n_{s_x}^n)^2 + (n_{s_y}^n)^2 + (n_{s_z}^n)^2} \end{aligned}$$

donde el superíndice n indica que el sistema de referencia en el cual se expresan los valores es el sistema de navegación (coincidente con el plano Π). Debido a que el cambio de coordenadas desde el sistema de referencia del sensor al sistema de referencia de navegación preserva la norma, eligiendo el vector normal como $\mathbf{n}_s^s = [0, 0, 1]^T$ (el cual tiene norma unitaria), finalmente la lectura del telémetro se puede expresar como

$$s = \frac{p_{s_z}^n}{n_{s_z}^n}. \quad (20)$$

Por lo tanto, la lectura del telémetro no solo dependerá de la altura a la que se encuentre el mismo sino también del tercer elemento de la tercera columna de la matriz de rotación que relaciona el sistema de referencia del mismo con el sistema en el que se mide la posición.

Para el caso de representación de orientación mediante cuaternión usando la matriz R_n^b , la ecuación (20) resulta

$$s = \frac{p_{s_z}^n}{q_0^2 - q_1^2 - q_2^2 + q_3^2}, \quad (21)$$

considerada como la ecuación de observación del telémetro.

III-B3. Implementación del autopiloto: Con los modelos cinemáticos y de observación desarrollados en las Secciones III-B1 y III-B2, y las ecuaciones de Kalman formuladas en la Sección II-A se implementa el autopiloto. La Figura 4 presenta el esquema utilizado. Con cada nueva lectura del giróscopo comienza la etapa de predicción utilizando las ecuaciones de la (1) a la (3) para obtener el estado predicho $\hat{\mathbf{x}}_{k|k-1}$ que se utiliza también en la predicción de la observación.

Cuando el acelerómetro, el módulo de visión o el telémetro devuelven una lectura \mathbf{z}_k , esta se utiliza para corregir los posibles errores en la predicción de la orientación o altura, comparándose con la predicción de la observación dada por $\hat{\mathbf{z}}_{k|k-1} = h(\hat{\mathbf{x}}_{k|k-1})$ obteniéndose la innovación $\mathbf{y} = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}$. En cada caso, la función $h(\cdot)$ será la ecuación de observación que corresponda a cada sensor (modelos dados por las ecuaciones (16), (19) y (21)), como se vio en la Sección III-B2.

El proceso de estimación se completa con la etapa de actualización utilizando las ecuaciones (4–8), presentadas en la Sección II-A.

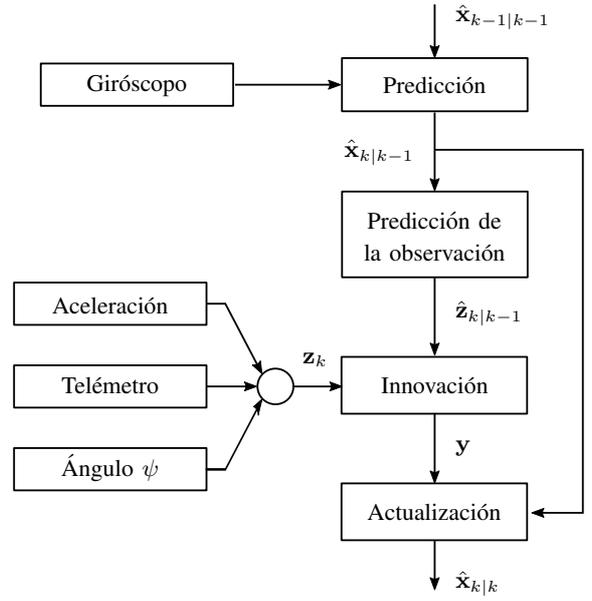


Figura 4. Esquema de filtrado para determinar la orientación y la altura

III-C. Odometría visual usando sensores inerciales

Como se vio en Sección II-B, se puede determinar la posición y la orientación de un vehículo a partir de una secuencia de imágenes capturadas usando una cámara montada rígidamente en el mismo. Estas imágenes pueden ser procesadas con distintos algoritmos como los vistos en la Sección II-B2 para detectar y seguir características entre imágenes consecutivas. Con los puntos correspondientes en cada imagen, se puede determinar la homografía y a partir de esta, obtener traslación y rotación. Finalmente, mediante odometría visual se puede componer la posición y orientación del vehículo en donde está montada la cámara respecto de un sistema de referencia, usando las ecuaciones vistas en la Sección II-B3. Como se detalla en diversas secciones, este trabajo se centra en la determinación de la posición y orientación de vehículos voladores, en particular los multirrotores, en los cuales la cámara se encuentra generalmente apuntando hacia abajo.

Si bien cuando el Algoritmo 2 da como resultado la rotación y la traslación entre las imágenes, debido a que la construcción de la homografía es realizada mediante puntos expresados en píxeles obtenidos a partir del flujo óptico, es necesario afectar a la traslación obtenida con la escala. Por otro lado, cuando un VANT se desplaza, su inclinación hace que la escala de cada píxel sea diferente, lo que genera un error en la determinación de la traslación. Para mitigar este error, en este trabajo se propone una ortogonalización de la imagen utilizando información de los sensores inerciales, previa a la determinación del flujo óptico, para luego ajustar la escala con una medida de altura uniforme.

Conocida la orientación y la altura del vehículo, utilizando la información de la geometría de la cámara, se pueden obtener los puntos que corresponden a la intersección de los vértices de la imagen con el plano que forma el suelo, usando como punto central de la cámara a $c_0 = [0 \ 0 \ h]^T$, siendo h la altura estimada de la cámara. Los valores de la matriz de calibración de la cámara que se utilizan para determinar los puntos del plano de la imagen pueden ser encontrados usando algoritmos como el presentado en [23]. De esta manera se modela la superficie del trapecio en el suelo que se proyecta al plano imagen cuando se produce la captura. En la Figura 5 se muestra este modelo, incluyendo el punto central, el plano imagen, y la superficie formada por la intersección de los puntos vértices del plano imagen en el plano del suelo.

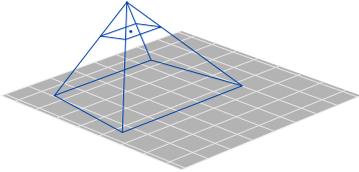


Figura 5. Modelo esquemático de la cámara y los puntos de protección en el suelo para el tiempo $k - 1$.

En el caso de ejemplo presentado en la Figura 5 se utiliza un ángulo de alabeo de 10° para exagerar el efecto, sin embargo en vuelos reales es más usual una inclinación máxima de 5° . Luego, usando el mismo punto central y matriz de calibración, pero con el eje óptico orientado perpendicularmente respecto del plano del suelo, se proyectan los puntos del plano imagen al plano del suelo, formando un rectángulo, como se presenta en la Figura 6.

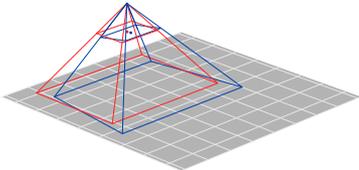


Figura 6. Modelo de la cámara en posición ortogonal y puntos de proyección en el suelo.

La imagen original debe proyectarse desde el trapecio a los de los puntos calculados del rectángulo produciendo lo que se conoce como ortogonalización de la imagen o rectificación ortogonal. La superficie útil que queda de la imagen en $k - 1$ es entonces la resultante de mapear los puntos del rectángulo en el trapecio. Tal como ejemplifica la Figura 7, se toma la intersección de las dos figuras geométricas y se completa el resto del rectángulo que representa la imagen con negro.

Cuando la nueva imagen esta disponible se repite el proceso anterior. La Figura 8 muestra el esquema de la cámara en el tiempo k y su rectificación ortogonal. También se puede ver la porción útil de la imagen en $k - 1$.

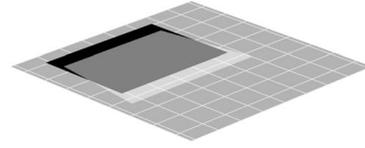


Figura 7. Superficie útil de la imagen en el tiempo $k - 1$.

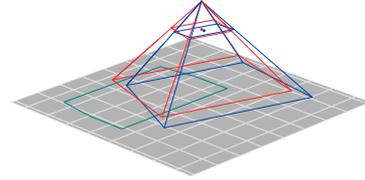


Figura 8. Modelo esquemático de la cámara, su proyección ortogonal en el tiempo k y la porción de suelo correspondiente a $k - 1$.

Finalmente, la Figura 9 muestra las dos superficies útiles sobre las cuales se realizan la detección de características y la estimación de movimiento mediante flujo óptico.

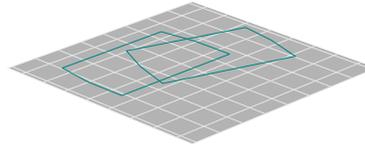


Figura 9. Porciones de suelo para el tiempo k y $k - 1$ utilizados para hacer el registro de imágenes 2D a 2D.

La Figura 10 muestra el resultado de la rectificación de una imagen mediante el método propuesto y el flujo óptico obtenido.

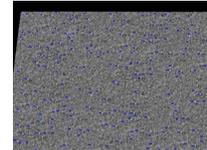


Figura 10. Rectificación ortogonal y posterior flujo óptico de una imagen capturada desde una cámara apuntando al suelo.

IV. ANÁLISIS EXPERIMENTAL

La validación del algoritmo desarrollado se realizó utilizando la herramienta de simulación *Robotics Toolbox* [24], compuesta de *scripts* de MATLAB para distintos modelos de robots, entre ellos un VANT multirrotor. El simulador retorna el valor verdadero del estado del VANT y las mediciones obtenidas por sensores con características configurables. Por otro lado, el algoritmo propuesto se implementó en el hardware disponible a bordo del QA3, un multirrotor desarrollado en el Centro de Investigación en Informática para la Ingeniería [25]. Se evaluó también la eficiencia computacional del algoritmo completo en las dos capas de hardware del QA3 (ARM Cortex-M3 para el autopiloto e Intel i5 para el módulo visual).

IV-A. Generación de datos para análisis

El vehículo simulado puede ser programado con trayectorias a seguir mientras se observan los valores de aceleración lineal y velocidad angular que sufre el vehículo. Con estos valores, se pueden obtener lecturas de sensores inerciales simulados dependiendo de la trayectoria preconfigurada así como los valores verdaderos de orientación y altura. La trayectoria elegida para presentar en este trabajo fue con el VANT partiendo de una posición en vuelo estacionario a 3.5m de altura hacia una altura final de aproximadamente 4m. En el proceso realiza dos giros de alrededor de 1m de radio. Esta trayectoria curva, obliga al vehículo a inclinarse para poder mantener el radio de giro, lo que aumenta la distancia al suelo dada por lectura del telémetro.

Para evaluar el módulo visual, se generaron un conjunto de vistas simuladas a partir de una gran imagen compuesta de ruido, simulando así algo semejante a vistas capturadas en un vuelo sobre una alfombra. Cada imagen fue formada utilizando la matriz de parámetros intrínsecos de la cámara y los valores verdaderos de posición y orientación, haciendo una proyección desde el suelo hacia el plano de la imagen teniendo en cuenta la orientación simulada.

En cuanto a la evaluación de la velocidad de ejecución, se utilizaron valores reales tomados de los sensores y se midió el tiempo de procesamiento en las plataformas disponibles.

IV-B. Análisis del autopiloto

Esta subsección presenta el análisis y los resultados obtenidos con la implementación del filtro de Kalman extendido propuesto en la Sección III-B.

La Figura 11 reporta las componentes del cuaternión verdadero y estimado que representan la orientación del vehículo simulado junto con la altura de vuelo. Con la línea de trazos negra se representa la orientación y la altura simulada, mientras que con la línea continua roja se muestra el resultado del filtro de Kalman extendido.

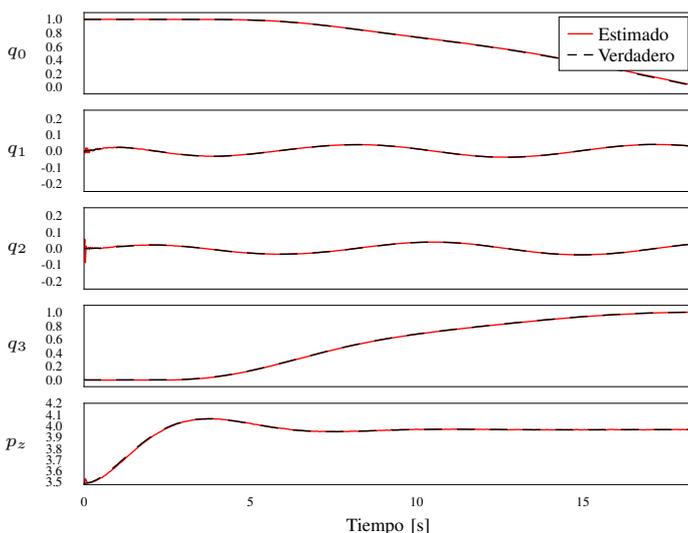


Figura 11. Estimación del cuaternión de orientación y la altura del VANT usando el EKF propuesto.

Si bien la Figura 11 muestra que existe una pequeña inestabilidad al inicio de la estimación, se puede ver que el filtro converge rápidamente y se estabiliza en el valor simulado. Salvo en los instantes iniciales, se puede ver que tanto los valores del cuaternión y la altura estimada son coincidentes con el valor verdadero durante toda la simulación.

El filtro de Kalman extendido propuesto fue implementado en lenguaje C y embebido en la computadora de abordo del QA3. Esta computadora, también llamada autopiloto, no tiene gran capacidad de procesamiento y solo puede llevar a cabo un número limitado de tareas.

El procesador central es un microcontrolador LPC1769 basado en un ARM Cortex-M3 con capacidad de funcionar a frecuencias de hasta 120MHz. Además cuenta con una gran cantidad de periféricos lo que lo hace útil para funcionar como procesador central en un autopiloto. La mayor dificultad que tiene este microcontrolador es que no cuenta con una unidad de punto flotante (FPU por *Floating Point Unit*). Esto significa que todas las operaciones de números de punto flotante se realizan por software y no por hardware como la mayoría de los procesadores. La diferencia es que cuando estas operaciones se realizan mediante módulos de hardware simplemente tardan un ciclo de reloj para su resolución, mientras que cuando se realizan por software pueden tardar decenas de ciclos.

La implementación en lenguaje C fue compilada con el gcc-arm-none-eabi en su versión 4.9.3 y para los cálculos matriciales fue utilizada la biblioteca estándar CMSIS (*Cortex Microcontrolles Software Interface Standard*) versión 2.0, la cual está optimizada para su utilización en esta plataforma. La biblioteca CMSIS cuenta con la ventaja adicional de poder usarse en arquitecturas que cuenten con FPU de manera transparente. De esta forma, en caso de reemplazarse el microcontrolador por uno con FPU, no hay que reescribir el código donde están implementadas las operaciones matriciales.

La Figura 12 presenta el tiempo demandado por cada etapa cuando son implementadas en la plataforma descrita en el párrafo anterior. Se puede hacer un análisis cualitativo del tiempo que demora cada etapa observando el largo de cada barra horizontal, la cual representa la media, mientras que las pequeñas barras verticales a los bordes representan la desviación estándar. La media y la desviación estándar fueron calculadas a partir de un experimento de 30 segundos obteniendo nuevos datos a razón de 100 veces por segundo.

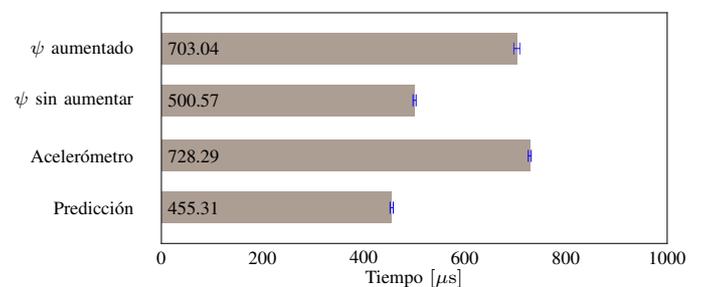


Figura 12. Tiempos de cómputo para cada etapa de los filtros implementado.

El análisis del tiempo de ejecución indica que cada una de las etapas que componen el EKF puede ser ejecutada en menos de $0.75\mu s$. La figura muestra también que el hecho de ampliar el vector de observación del ángulo ψ incrementa menos del 50% el tiempo de cómputo de la actualización. El tiempo total demandado por el EKF es menor a 2.4ms lo que permite controlar el VANT a una velocidad de 400Hz.

IV-C. Análisis del módulo visual

Se evaluó el algoritmo propuesto utilizando el conjunto de imágenes generadas usando los datos de la simulación más los datos obtenidos en la estimación de orientación y altura con el filtro de Kalman extendido de la Sección IV-B.

La Figura 13 presenta los resultados obtenidos utilizando odometría con las ecuaciones descritas en la Sección II-B y la comparación con el método propuesto para la posición en x y en y y para el ángulo ψ . Las líneas de trazos discontinuos negros grafican el valor verdadero obtenido a partir de la simulación. En líneas azules se grafica el valor de la odometría visual a partir de imágenes en donde no se usa información de la unidad inercial, esto es imágenes de las que no fue removida la distorsión perspectiva, mientras que las líneas rojas representan la posición y ángulo de ψ obtenidos a partir de la odometría sobre las imágenes ortogonales. Los resultados indican que el uso de imágenes ortogonales para hacer el seguimiento de las características logra estimar la posición de forma correcta, ya que la línea de trazos negros está superpuesta con la línea roja. Por el contrario, la línea azul tiene un error en la estimación cuando se produce el cambio de dirección, tanto en el eje x como en el eje y . En cuanto a la estimación del ángulo de ψ , ambos métodos muestran certeza en la estimación, aunque con cierta deriva debido al propio proceso de integración.

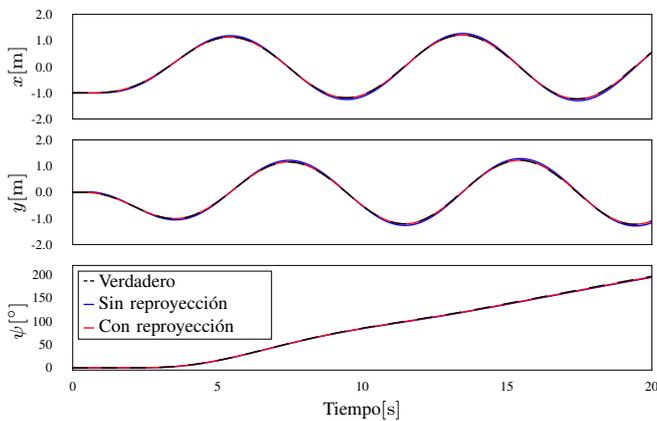


Figura 13. Comparación de odometría visual con y sin imágenes ortogonales.

La Figura 14 reporta el error en valor absoluto, esto es la diferencia entre el valor verdadero y la estimación, para el caso cuando se rectifica la imagen (rojo) y para el caso tradicional (azul). Los resultados indican que usando el método de imágenes ortogonales el error es menor que en el caso tradicional, siendo para este caso de cerca de 2.5cm para el método propuesto contra 8cm del original.

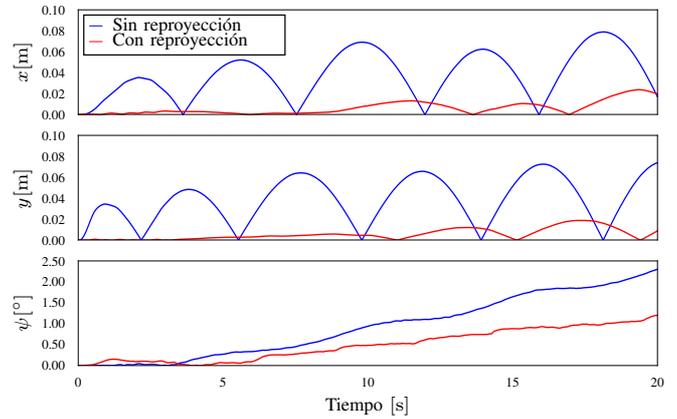


Figura 14. Error absoluto en la odometría visual.

Los resultados obtenidos implican que el error porcentual es de 0.18% para el algoritmo propuesto, tomando en cuenta que la trayectoria total recorrida en valor absoluto es cerca de 13.8m. Se logra mejorar significativamente el error que se tiene al utilizar odometría con imágenes sin preprocesar, que es del 0.57%.

La Figura 15 reporta el tiempo demandado por las dos funciones que requieren mayor poder de cómputo. Los algoritmos fueron implementados en C++ usando la biblioteca OpenCV, compilados con g++-4.8 con la bandera de optimización O2. La evaluación de tiempos de ejecución se realizó en un procesador Intel i5 de 3ra generación corriendo a 3.4GHz con 8GB de RAM.

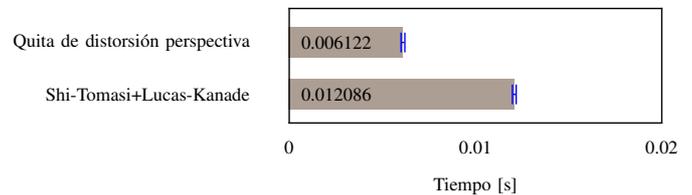


Figura 15. Análisis de velocidad de procesamiento usando características de intensidad.

La barra superior en la Figura 15 reporta el tiempo de ejecución de la función que se encarga de quitar la perspectiva a la imagen utilizando la información proveniente del autopiloto. La barra inferior indica el tiempo demandado por el detector de características de intensidad más el tiempo tomado por el algoritmo de flujo óptico. En ambos casos, se reporta la desviación estándar en líneas azules.

El análisis de eficiencia computacional indica que quitar la perspectiva incrementa el tiempo de procesamiento en aproximadamente 6ms. Asimismo, ambas funciones pueden realizarse en aproximadamente 18ms. Estos resultados demuestran que es posible procesar las imágenes provenientes de una cámara de hasta 50 cuadros por segundo en tiempo real.

V. CONCLUSIONES Y TRABAJO FUTURO

Este trabajo presenta el diseño y la implementación de un sistema para determinar la posición y la orientación de un VANT. El sistema está formado por dos capas de hardware, una para lograr la estabilización del VANT y otra para procesar las imágenes. Entre ambas existe un acoplamiento en las dos direcciones, para minimizar los errores en la estimación. El primer acoplamiento consiste en corregir el cuaternión de orientación calculado con sensores inerciales, utilizando información del ángulo de guiñada proveniente del módulo visual. El segundo acoplamiento consiste en el uso de información de orientación y altura proveniente del autopiloto para minimizar el error en el cálculo de la odometría visual. El método consiste en la quita de perspectiva de las imágenes para luego calcular la odometría.

Los algoritmos propuestos fueron evaluados usando simulaciones para analizar su validez. A continuación se analizó la eficiencia computacional de las implementaciones realizadas en hardware real, con el objetivo de determinar si es posible utilizar ambos algoritmos en tiempo real, procesando la información y obteniendo resultados antes de que las nuevas lecturas estén disponibles.

De acuerdo a los resultados obtenidos, se concluye que utilizando los algoritmos propuestos se puede estabilizar el VANT a una frecuencia de 400Hz, la cual es superior a la tasa de muestreo de los sensores involucrados. Además, el controlador disponible funciona a 100Hz, por lo que se puede asegurar que esta parte del sistema trabaja en tiempo real. Por otro lado, las imágenes pueden ser procesadas a una frecuencia de 50 cuadros por segundo, lo cual es suficiente para las cámaras disponibles ya que pueden registrar el entorno a 30 cuadros por segundo. Todos estos resultados demuestran que se puede lograr el control del VANT en tiempo real aplicando el algoritmo propuesto.

La principal línea de trabajo futuro, con el objetivo de reducir los costos de manufactura, se orienta a la integración de los dos módulos descritos en este trabajo en un solo microprocesador con capacidad de aplicar cómputo en paralelo. La arquitectura objetivo es la que integra en el mismo circuito microprocesadores ARM con FPGA (*Field Programmable Gate Array*). De esta manera, se prevé una reducción no solo en los costos, sino también en el gasto energético, lo que derivará en un aumento de la autonomía del VANT.

REFERENCIAS

- [1] H. Lim, J. Park, D. Lee, and H. Kim, "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 33–45, Sept 2012.
- [2] Y. Lu, D. Macias, Z. Dean, N. Kreger, and P. Wong, "A UAV-mounted whole cell biosensor system for environmental monitoring applications," *IEEE Transactions on NanoBioscience*, vol. 14, no. 8, pp. 811–817, 2015.
- [3] M. Erdelj and E. Natalizio, "UAV-assisted disaster management: Applications and open issues," in *International Conference on Computing, Networking and Communications*, 2016, pp. 1–5.
- [4] A. Samad, N. Kamarulzaman, M. Hamdani, T. Mastor, and K. Hashim, "The potential of unmanned aerial vehicle (UAV) for civilian and mapping application," in *System Engineering and Technology*, 2013, pp. 313–318.
- [5] F. Hoffmann, N. Goddemeier, and T. Bertram, "Attitude estimation and control of a quadcopter," in *Intelligent Robots and Systems*, 2010, pp. 1072–1077.
- [6] C. Paz, G. Perez, and M. Pucheta, "Acoplamiento en la estimación de la orientación y la altura mediante filtro extendido de Kalman," in *Mecánica Computacional Vol XXXIV*, 2016, pp. 3419–3442.
- [7] Y. Zhang, T. Wang, Z. Cai, Y. Wang, and Z. You, "The use of optical flow for UAV motion estimation in indoor environment," in *IEEE Chinese Guidance, Navigation and Control Conference*, 2016, pp. 785–790.
- [8] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.
- [9] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition*, 2004, pp. 652–659.
- [10] H. Moravec, "Visual mapping by a robot rover," in *6th International Joint Conference on Artificial Intelligence*, 1979, pp. 598–600.
- [11] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, 2000.
- [12] C. Harris and M. Stephens, "A combined corner and edge detection," in *4th Alvey Vision Conference*, 1988, pp. 147–151.
- [13] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition*, 1994, pp. 593–600.
- [14] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [15] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *7th International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.
- [16] A. Aziz and H. Karara, "Direct linear transformation into object space coordinates in close-range photogrammetry," in *ASP Symposium on Close-Range Photogrammetry*, 1971, pp. 1–18.
- [17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [18] E. Malis and M. Vargas, "Deeper understanding of the homography decomposition for vision-based control." *INRIA Research Report 6303*, 2007.
- [19] Y. Ma, S. Soatto, J. Kosecká, and S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. New York, NY: Springer, 2010.
- [20] J. Solà, "Quaternion kinematics for the error-state KF." [Online]. Available: <http://www.iri.upc.edu/files/scidoc/1773-Quaternion-kinematics-for-the-error-state-Kalman-filter.pdf>, accessed April 2017
- [21] G. Araguás, C. Paz, D. Gaydou, and G. P. Paina, "Quaternion-based orientation estimation fusing a camera and inertial sensors for a hovering UAV," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 1, pp. 37–53, 2015.
- [22] W. Phillips, C. Hailey, and G. Gebert, "Review of attitude representations used for aircraft kinematics," *Journal of Aircraft*, vol. 38, no. 4, pp. 718–737, 2001.
- [23] O. Faugeras and G. Toscani, "Camera calibration for 3d computer vision," in *International Workshop on Machine Vision and Machine Intelligence*, 1987, pp. 240–247.
- [24] P. Corke, *Robotics, Vision and Control*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2011, vol. 73.
- [25] D. Gaydou, G. Suarez, C. Paz, G. Paina, and G. Araguás, "Robot volador no tripulado QA3. Diseño y construcción de un cuatrirrotor para experimentación," in *VIII Jornadas Argentinas de Robótica*, 2014.