

## CARACTERIZACIÓN DE SENSOR DE FLUJO ÓPTICO PARA APLICACIÓN EN NAVEGACIÓN INTEGRADA

Concia Bernardo, Zumarraga Augusto, Mitidieri Pedro

Departamento de Aeronáutica - Facultad de Ingeniería  
Calle 116 e/47 y 48 - (1900) La Plata  
e-mail: bernardo.concia@ing.unlp.edu.ar

### 1 - INTRODUCCIÓN

El siguiente trabajo presenta el desarrollo y caracterización de un sensor de flujo óptico. Se plantean también las ecuaciones necesarias para ser utilizado en un algoritmo de navegación integrada de 6 grados de libertad.

Los avances tecnológicos en procesamiento y sensado que ofrece la microelectrónica amplió significativamente las capacidades de navegación autónomas de vehículos no tripulados (UAV). Ha permitido, entre otros, el uso de algoritmos de procesamiento digital de imágenes en tiempo real y a bordo. Motivados por estos avances este trabajo se encuentra enmarcado en el desarrollo del módulo de sensado y navegación para un UAV.

Los sensores de imagen obtienen una proyección plana de un escenario dentro de un cierto ángulo de visión. Cualquier movimiento relativo, entre el sensor y los objetos en el campo de visión, tendrá una relación directa con el desplazamiento detectado por el sensor. Si se logra calcular el movimiento entre dos imágenes consecutivas estamos en condiciones de calcular la magnitud y dirección de cambio de una región de interés en el plano del sensor. Conociendo el tiempo entre las imágenes podemos ver la magnitud y dirección como un desplazamiento diferencial en el tiempo, es decir, una velocidad.

El campo de velocidades en toda la imagen es denominado *Flujo óptico*. Algunos algoritmos calculan el movimiento de cada uno de los píxeles de la imagen mientras que otros sólo calculan puntos de interés. En este caso desarrollamos un sensor de flujo óptico que solo calcula el movimiento de puntos claves particulares del tipo esquina[1].

Se desarrollarán los pasos principales del algoritmo utilizado en la detección de los puntos claves. Luego se presentará el algoritmo de seguimiento que calcula el desplazamiento de cada uno de estos puntos claves entre cuadro y cuadro. Ambos métodos son integrados luego en el algoritmo final utilizado para el cálculo de flujo óptico. Debido a que el sensor será utilizado para navegación, se exhibe la ecuación de salida del mismo para ser utilizado en el algoritmo de navegación integrada. Finalmente se presentan los resultados obtenidos en diferentes ensayos y se caracteriza su rendimiento.

### 2 - DESARROLLO DEL ALGORITMO

#### 2.1. Detección de Features

Existen diversos métodos para encontrar características en imágenes como líneas, circunferencias, formas geométricas específicas, objetos etiquetados y esquinas. A partir de los resultados obtenidos en [1] se resolvió utilizar puntos esquinas gracias a que son invariantes a los movimientos de la cámara y a cambios no uniformes en la intensidad de los píxeles. Además son puntos claves que dan un buen rendimiento en el algoritmo de seguimientos elegido.

En la literatura se encuentran múltiples algoritmos para la detección de esquinas como por ejemplo Harris o FAST[2]. Otros algoritmos además generan descriptores de cada punto clave encontrado. Esto permite identificar un mismo punto clave en dos imágenes diferentes, sin importar la ubicación de los mismos. Generalmente estos algoritmos son invariantes a los cambios de escala y rotación entre las imágenes. En nuestro sensor no es necesario utilizar descriptores. Sin embargo, desarrollar el sensor de flujo óptico con puntos claves sumado a descriptores y otros métodos permiten elaborar un algoritmo de posición relativa en vez de velocidad. Esta es una de las razones por la cual se decidió utilizar puntos claves esquinas a diferencia del más utilizado sensor de flujo óptico que es calculado en todos los píxeles de la imagen.

El algoritmo seleccionado fue ORB[3], sin calcular los descriptores de los puntos claves. Este algoritmo utiliza FAST [2]. Para ordenar los puntos por probabilidad de ser esquina utiliza el detector de Harris[4]. Gracias a ello permite la elección de la cantidad de puntos a mostrar y solo devuelve la cantidad deseada de puntos esquina. Esto es de gran utilidad ya que el costo computacional debe mantenerse bajo para una mayor velocidad de cómputo.

El método FAST encuentra esquinas analizando los píxeles alrededor del punto a evaluar. Para discernir entre un punto que es una esquina y otro que no, el algoritmo busca punto a punto en toda la imagen. En cada píxel de estudio crea un círculo de otros píxeles que tienen al primero como centro y realiza una comparación de intensidades. Si existe un arco ininterrumpido de una cierta cantidad de píxeles cuyas intensidades se encuentran por arriba o abajo de la intensidad del centro con cierto umbral, entonces dicho píxel es una esquina. El algoritmo es muy eficiente debido a que utiliza un algoritmo de machine learning como es árbol de decisión para evaluar de manera eficiente cada uno de los puntos y descartar rápidamente con menos pasos los puntos que no son esquinas.

## 2.2. Seguimiento

El algoritmo que se utilizó para el seguimiento de los puntos es 'Lucas-Kanade'[5] con una implementación con pirámides. El método permite encontrar con bajo costo computacional el desplazamiento de los puntos seleccionados.

Para encontrar el flujo óptico, el método considera que la intensidad de los píxeles entre una imagen y la siguiente se mantiene constante y que los gradientes son constantes en la cercanía de los puntos evaluados. A partir de las dos imágenes se obtiene el desplazamiento del punto en cuestión. En general se coloca una cota al límite de movimiento esperado de un punto clave, aplicando el algoritmo solo en una subregión de la imagen en las inmediaciones del punto clave a evaluar. Este punto a evaluar llamado  $\mathbf{u}$  con coordenadas  $(u_x, u_y)$  en la imagen  $\mathbf{I}$ , es buscado en la nueva imagen  $\mathbf{J}$  dada por  $\mathbf{v} = (u_x + d_x, u_y + d_y)$ . Para ello se deben hallar los valores de  $d_x$  y  $d_y$  que minimicen la siguiente función residual  $\epsilon$ :

$$\epsilon(\mathbf{d}) = \epsilon(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (I(x, y) - J(x + d_x, y + d_y))^2 \quad (1)$$

La región de búsqueda es una subregión de la imagen con valores  $w_x$  y  $w_y$ ; el flujo óptico es  $d_x$  y  $d_y$ . Para encontrar estos valores de flujo óptico se utiliza un método iterativo para encontrar ceros de funciones similar al procedimiento de Newton-Raphson. El primer paso es computar la diferencia entre las imágenes con un valor de  $\mathbf{d}_{inicial} = 0$ . Luego se computa el vector de diferencia  $\mathbf{b}_k$ . Este vector representa que tan similares son ambas imágenes, ponderada con

las derivadas direccionales de intensidad de la imagen.

$$b_k = \sum_{x=u_x^L-w_x}^{u_x^L+w_x} \sum_{y=u_y^L-w_y}^{u_y^L+w_y} \begin{bmatrix} \delta I_k(x, y) I_x(x, y) \\ \delta I_k(x, y) I_y(x, y) \end{bmatrix} \quad (2)$$

Para cada imagen se calculan las derivadas de las intensidades en direcciones  $I d_x$ ,  $I d_y$  y  $I d_{xy}$  que luego son utilizadas para obtener la matriz gradiente  $G$ .

$$G = \sum_{x=u_x^L-w_x}^{u_x^L+w_x} \sum_{y=u_y^L-w_y}^{u_y^L+w_y} \begin{bmatrix} I d_x^2 & I d_x I d_y \\ I d_x I d_y & I d_y^2 \end{bmatrix} \quad (3)$$

Una vez obtenida la matriz  $G$  y el vector diferencia  $b$ , se aproxima con la siguiente ecuación el valor  $\eta$ .

$$\eta = G^{-1} b \quad (4)$$

Finalmente, se calcula la nueva aproximación de  $d_x$  y  $d_y$ , que es utilizada como punto de partida para la siguiente iteración.

$$d_k = d_{k-1} + \eta_k \quad (5)$$

Las iteraciones finalizan cuando  $\eta^k$  es menor que un umbral elegido, y el valor de flujo óptico es encontrado. En caso de no converger a un valor lo suficientemente aceptable el punto bajo evaluación es descartado y marcado como perdido.

Nuestra implementación es realizada con pirámides para alcanzar mayor velocidad. Las pirámides son versiones sucesivas de menor resolución de las imágenes originales. El flujo óptico es calculado en el último nivel de la pirámide para obtener una estimación aproximada de la ubicación del punto clave. Con esta estimación se pasa al siguiente nivel de la pirámide y se repite el procedimiento. La diferencia es que se utiliza la aproximación del desplazamiento hallado en el nivel anterior  $d_{ini}^L = d_{final}^{L+1}$ .

### 2.3. Implementación del Algoritmo para Cálculo de Flujo Óptico

La estimación del flujo óptico es obtenida a partir del siguiente algoritmo y es el que ha sido utilizado para la caracterización del sensor. Es un método cíclico donde se evalúan imágenes consecutivas en tiempo de muestreo  $t_{n-1}$  y  $t_n$ .

1. Tomar imagen en  $t_{n-1}$ .
2. Corregir distorsión de imagen  $t_{n-1}$ .
3. Detectar los puntos claves en la imagen en  $t_{n-1}$ .
4. Tomar imagen en  $t_n$ .
5. Corregir distorsión de imagen  $t_n$ .
6. Seguimiento de los puntos en imagen  $t_{n-1}$  a la imagen  $t_n$  con algoritmo Kanade-Lucas.
7. Sumar los puntos encontrados ( $N$ ). Si  $N < 10$  ir a punto 3.
8. Promediar del movimiento de los puntos en unidad de píxeles.

El algoritmo es repetido desde el punto 4 a la frecuencia deseada.

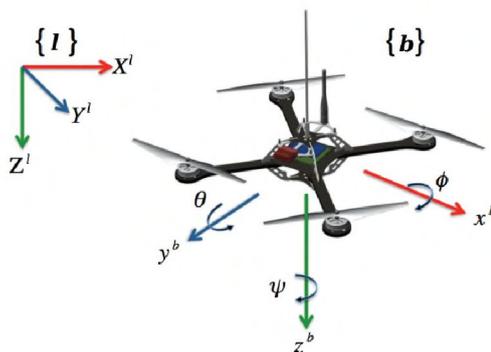


Fig. 1: Sistemas de Coordenadas - b=body - l=local

### 3 - ECUACIÓN DE SALIDA PARA NAVEGACIÓN INTEGRADA

La integración de la medición en el algoritmo de navegación integrada[6] se realiza a partir de la ecuación que relaciona los estados, en el caso de nuestra aplicación, del quadricóptero y la medición de flujo óptico. Se define para ello un sistema de coordenadas local geográfico (terna- $l$ ), fijo en el suelo y el sistema de coordenadas móvil solidario con la aeronave y al sensor óptico (terna- $b$ ). Los cuales se pueden observar en la figura 1. El centro de la terna- $b$  es colocada en el sensor. Para la proyección del espacio en el plano de foco se utiliza el modelo de la cámara estenopeica. En la literatura se encuentran implementaciones del sensor de flujo óptico para navegación de quadricópteros que no tienen en cuenta la medición que genera la rotación de la cámara[7]. En la ecuación de salida presentada se tiene en cuenta tanto la traslación como la rotación del sensor.

- $\mathbf{p}_j^l$  Posición del punto clave  $j$  proyectado en terna- $l$
- $\mathbf{c}^b$  Posición de la cámara (en terna- $l$ )
- $\mathbf{d}_j^b$  Distancia relativa entre la cámara y el punto clave  $j$
- $\mathbf{v}_j^b$  Velocidad relativa entre la cámara y el punto clave  $j$
- $\mathbf{C}_l^b$  Matriz de rotación de la terna  $l$  a la terna  $b$
- $\mathbf{p}_j^o$  Proyección de  $\mathbf{d}_j^b$  en el plano de la cámara ( $xy^b$ ), mm/s
- $\mathbf{v}_j^o$  Velocidad de la proyección de  $\mathbf{d}_j^b$  en el plano de la cámara, en pixeles por segundo
- $\mathbf{v}^o$  Flujo óptico

La distancia relativa es  $\mathbf{d}_j^l = \mathbf{p}_j^l - \mathbf{c}^l$ , que puede proyectarse en la terna móvil mediante la matriz de rotación correspondiente:  $\mathbf{d}_j^b = \mathbf{C}_l^b \mathbf{d}_j^l$ . En una imagen se computa la proyección de la distancia relativa en el plano de la cámara, y su magnitud depende de los factores de perspectiva  $f/h$  en cada eje:

$$\mathbf{p}_j^o = \frac{1}{h} \begin{bmatrix} fx & 0 & 0 \\ 0 & fy & 0 \end{bmatrix} \mathbf{d}_j^b \quad (6)$$

donde  $h = \mathbf{d}_j^b \cdot \mathbf{z}^b$  es la distancia entre el plano de la cámara y el plano paralelo a este que contiene al punto clave; que debe determinarse independientemente.

La medición del sensor es la variación entre la posición del punto clave en dos cuadros consecutivos, lo cual se corresponde con la velocidad de la proyección del punto clave en el plano de la cámara:

$$\mathbf{v}^o = \frac{1}{N} \sum_{j=1}^N \dot{\mathbf{p}}_j^o = \frac{1}{h} \begin{bmatrix} fx & 0 & 0 \\ 0 & fy & 0 \end{bmatrix} \frac{1}{N} \sum_{j=1}^N \left( \mathbf{v}_j^b - \frac{\dot{h}}{h} \mathbf{d}_j^b \right) \approx \frac{1}{h} \begin{bmatrix} fx & 0 & 0 \\ 0 & fy & 0 \end{bmatrix} \frac{1}{N} \sum_{j=1}^N \mathbf{v}_j^b \quad (7)$$

en donde se descarta el segundo término dentro de la sumatoria asumiendo que los puntos

clave están distribuidos de forma homogénea alrededor del centro de la imagen. Por otra parte:

$$\mathbf{v}_j^b = \dot{\mathbf{d}}_j^b = [\mathbf{C}_l^b \mathbf{v}_j^l + \dot{\mathbf{C}}_l^b \mathbf{d}_j^l] \quad (8)$$

Sustituyendo:

$$\mathbf{v}^o \approx \frac{1}{h} \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \end{bmatrix} [\mathbf{C}_l^b \bar{\mathbf{v}}^l + \dot{\mathbf{C}}_l^b \bar{\mathbf{d}}^l] \quad (9)$$

donde  $\bar{\mathbf{d}}$  y  $\bar{\mathbf{v}}^l$  son la distancia y velocidad media respecto de los puntos clave; y además se cumple que  $h = |\bar{\mathbf{d}}|$ .

#### 4 - CARACTERIZACIÓN EXPERIMENTAL DEL SENSOR

Se realizaron una serie de ensayos para caracterizar el sensor. Los mismos se realizaron con una iluminancia en el plano horizontal de la cámara de 250lx, un valor de iluminación general de una vivienda[8]. Se tomaron los mejores 50 puntos en el algoritmo de detección. Con seguimiento a todos los puntos claves el algoritmo demora no mas de 6ms, con un valor medio de 4ms. La máxima frecuencia es de alrededor de 200 Hz, siempre y cuando el valor de iluminancia permita un buen contraste en la imagen. Debido a los requerimientos de navegación se utilizó una frecuencia de 30 Hz y una resolución de  $160px \times 120px$ . Se realizaron pruebas satisfactorias a 90Hz con una iluminancia de 700lx. Se utilizó una computadora de placa reducida modelo Raspberry Pi 3B+ equipada con una cámara CCD Camera Module V2, cuyo campo de visión es de  $62,2 \pm 0,13^\circ$  horizontalmente ( $\alpha_h$ ) y  $48,8 \pm 0,11^\circ$  verticalmente ( $\alpha_v$ ).

##### 4.1. Ruido

Se evaluó el ruido del sensor en estado de reposo. Se realizaron ensayos de 120 segundos con puntos claves en un plano a 1 metro y a 2 metros del sensor. Se utilizó la misma iluminancia en todos los ensayos, 250lx. La medición de flujo óptico fue convertida a velocidad relativa entre la cámara y los puntos claves de forma aproximada utilizando la ecuación 10 que se obtiene a partir de aproximar la ecuación 9.

$$\dot{p}_{x,y} = \frac{\mathbf{v}_{x,y}^o f_{x,y}}{h} \quad (10)$$

Los coeficientes  $f_x$   $f_y$  dan la relación entre una distancia en la imagen en píxeles y la misma distancia en el plano proyectado, en función de la distancia entre el sensor y el plano;  $f_x = 132,7$  y  $f_y = 132,3$ ;  $h$  es la distancia entre el eje de rotación y el plano perpendicular donde se encuentran los puntos claves;  $\dot{p}_{x,y}$  es la velocidad relativa que esta en mm/seg. y  $\mathbf{v}_{x,y}^o$  es el flujo óptico en píxeles/seg.

En la figura 2a y 2b se encuentran los resultados para varias corridas a 1 metros y a 2 metros. La figura 2a muestra el error del sensor en estado estacionario, mientras que la figura 2b muestra lo que sucede en las primeras décimas de segundo donde el ruido es mayor. Una posible explicación es que el algoritmo de seguimiento encuentra un punto con mayor linealidad y estabilidad que el detectado por el algoritmo de detección.

La varianza promedio para el caso de 1 metro fue de  $8,5e - 5mm^2$  con varianza máxima de  $1,3e - 4mm^2$ ; en cuanto a el ensayo a 2 metros la varianza promedio fue de  $3,5e - 4mm^2$  y la máxima de  $9,2e - 4mm^2$ . En la figura 3 se puede ver la densidad espectral de potencia de ruido para varios ensayos. Lejos esta de ser un ruido blanco, dado que las fluctuaciones se dan por diferentes motivos. Alguno de ellos son las diferentes texturas de las imágenes y la no

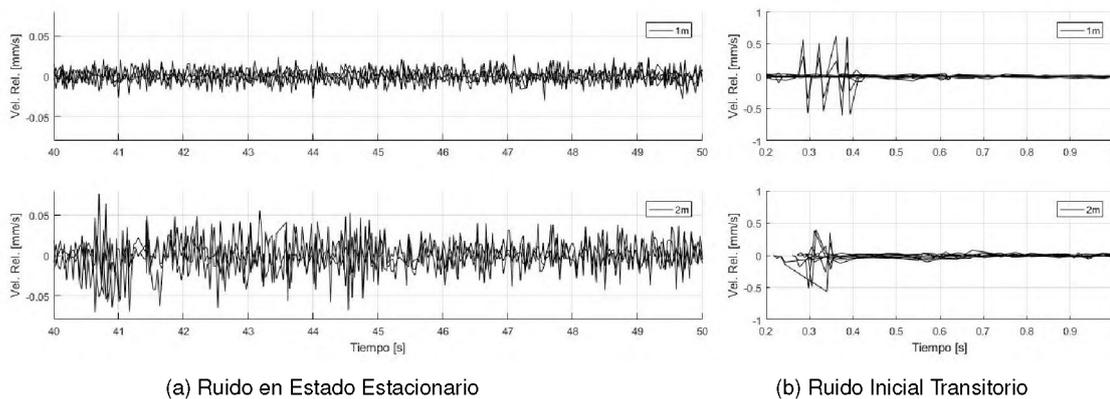


Fig. 2: Ruido de Sensor Óptico

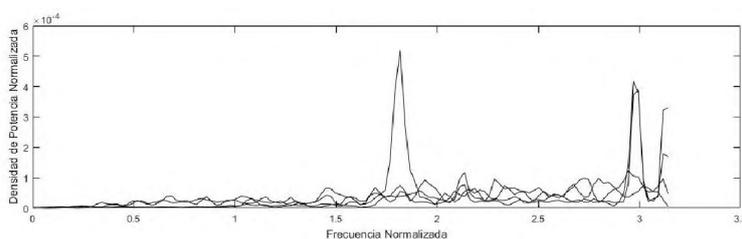


Fig. 3: Densidad Espectral de Potencia de Ruido

linealidad del gradiente de intensidad de la imagen. Las texturas de alta frecuencia generan errores en la estimación de la posición del punto clave cambiando la estimación por pequeñas variaciones de la imagen. Esto además se suma a los errores del algoritmo dado de que la variación de intensidad no es lineal en la mayoría de las imágenes tomadas.

#### 4.2. Integración del ruido (Random Walk)

Si bien el flujo óptico es una medición de velocidad, al ser integrado puede obtenerse una estimación de posición. Integrar para conseguir posición suele denominarse navegación a la estima (dead-reckoning). El método no es del todo efectivo, debido a que la integral de una señal con ruido diverge. Para que esto no suceda otros algoritmos deberán ser implementados. En la figura 4 se puede ver la estimación de posición con el sensor en estado de reposo. Nuevamente aparece un transitorio en la inicialización que suma un error considerable. En estado estacionario la integración diverge en menor cantidad. Ningún ensayo presento más de 3 milímetros de error pasados los 60 segundos para alturas de 1 y 2 metros.

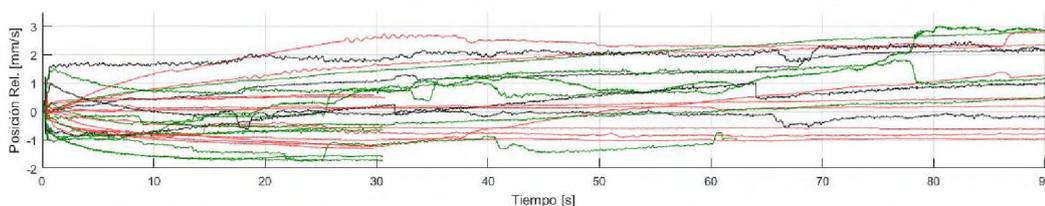


Fig. 4: Integración del Ruido de Medición (Random Walk)

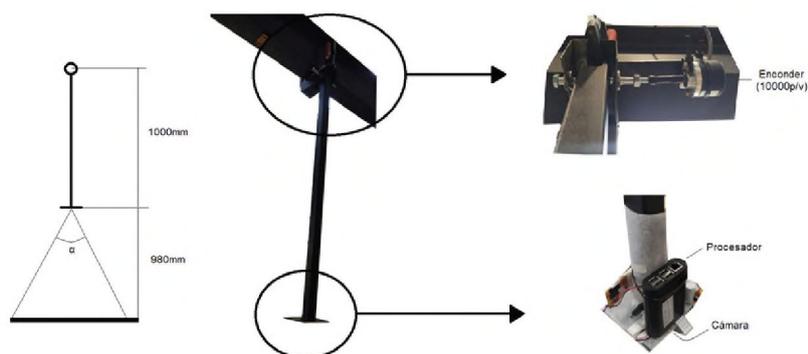
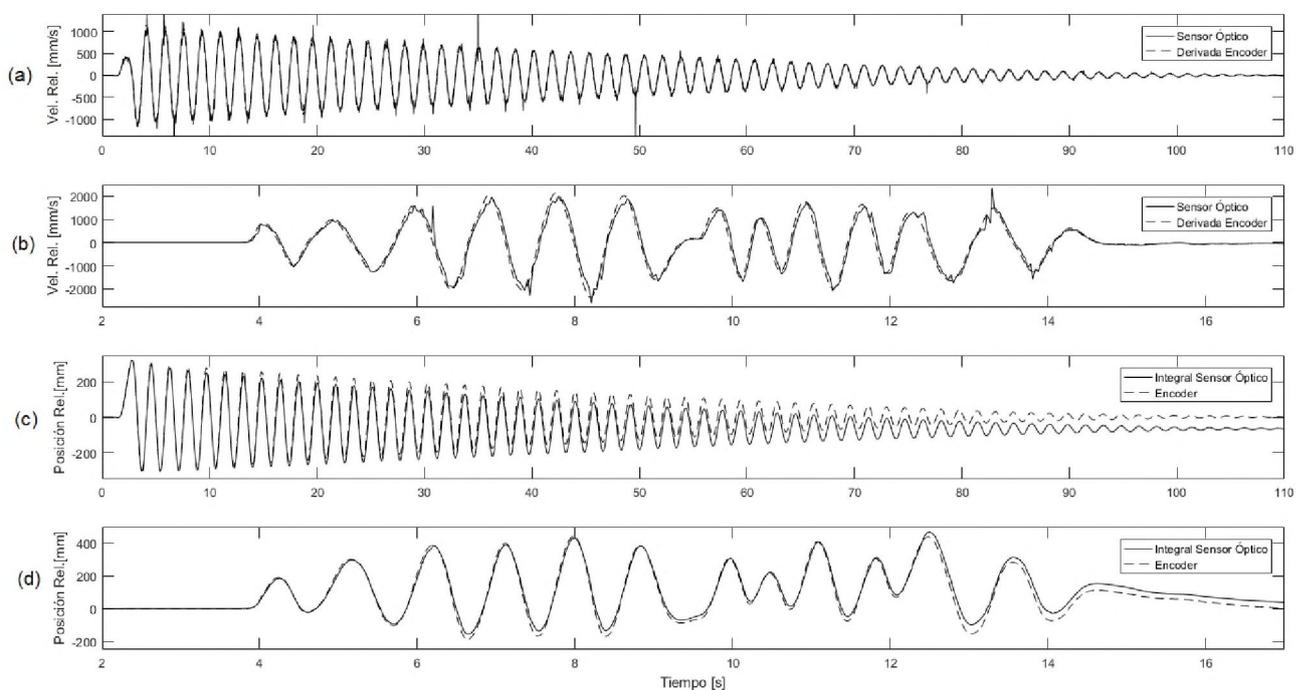
Fig. 5: Diagrama de Ensayo con Péndulo.  $\alpha_{h,v}$  = Ángulo de Apertura

Fig. 6: Ensayos de Velocidad y Posición Relativa

### 4.3. Péndulo

Para poner a prueba el sensor óptico, se construyó un péndulo plano, ya que sabiendo la longitud y el ángulo, fácilmente pueden obtenerse los datos de velocidad y posición. El mismo como se ve en la figura 5 tiene una longitud de 980mm y el eje de rotación está conectado a un encoder de 2500 pulsos por vuelta es decir 10000 puntos de resolución por vuelta. Esto da una resolución de  $0.036^\circ$ . En la base del péndulo se encuentra el sensor CCD a una distancia de 1000mm del piso. A partir de los datos obtenidos por el encoder se calculó la velocidad, para esto se derivó y alisó la medición de posición para obtener una curva suave de velocidad que pueda ser comparada con la medición del flujo óptico. Para comparar resultados se convirtieron los valores del sensor óptico medidos en píxeles/s a mm/s, a partir de la altura del péndulo y el campo de visión de la cámara 10.

En la figura 6 están los resultados obtenidos en dos ensayos con el péndulo; uno extenso con movimiento libre del péndulo y variaciones de  $\pm 350\text{mm}$  y  $\pm 1000\text{mm/s}$  (figuras 6a y 6c); el

otro ensayo es forzado y de corto tiempo con variaciones de  $\pm 450\text{mm}$  y  $\pm 2000\text{mm/s}$  (figuras 6b y 6d). Por un lado se encuentran las velocidades relativas (figuras 6a y 6b) y por el otro las posiciones relativas (figuras 6c y 6d), obtenidas de integrar la medición de flujo óptico. En el caso de la posición, la resolución del sensor óptico fue menor que la del encoder, y los errores acumulados al final del ensayo fueron de 100mm para el ensayo libre y 30mm para el forzado. En los gráficos de velocidad relativa, puede observarse que el sensor óptico tiene discontinuidades en la medición que pueden ser filtrados con facilidad. Estas discontinuidades pueden tener origen en el ruido ya explicado o también ser producto de sombras pasajeras que cambian la intensidad local en la imagen.

## 5 - CONCLUSIONES

El sensor de flujo óptico cumple los requisitos para ser implementado en un algoritmo de navegación. Además de su velocidad de cálculo, su bajo coste y peso lo hacen útil para aplicaciones in situ en pequeños UAVs.

Se encontraron características que permiten la mejora del algoritmo. Se deslumbraron mayores errores en la inicialización del algoritmo. Estos errores podrán ser evitados dando suficiente tiempo al sensor para encontrar los puntos claves mas estables. Se observaron saltos en la medición que pueden ser eliminados fácilmente conociendo las tasas de cambio máximas esperadas. Solo se logró colocar una cota mayor aproximada a la varianza del ruido del sensor, ya que este es dependiente de la escena que se esta sensando. Entre las posibles fuentes de error se encuentran imágenes con texturas de alta frecuencia, limitaciones del algoritmo de seguimiento en variaciones de intensidad no lineales y variaciones locales de luminosidad, causadas en su mayoría por sombras.

El sensor de flujo óptico puede utilizarse para estabilizar la velocidad en un algoritmo de navegación de un quadricóptero. El ruido de medición en reposo es escaso, y a medida que aumenta la magnitud de flujo óptico, aumenta tambien el ruido del sensor. Al integrarlo la medición para estimar posición con movimientos de  $\pm 300\text{mm}$  se obtienen velocidades de error de posición menores a los 50mm/minuto. En reposo, a velocidad del error es inferior a 3mm/minuto.

Entre los trabajos futuros se encuentran: La integración del sensor en un quadricóptero, implementar el algoritmo en distintos hilos de procesamiento y obtener una estimación de posición relativa basada en los puntos claves iniciales, lo que lograría una sustancial mejora de la estimación. De esta manera también se logrará estabilizar la estimación de posición incluso con variación de intensidades de regiones o objetos en movimiento, algo que este sensor no es capaz de realizar por si solo.

## 6 - BIBLIOGRAFIA

- [1] J. Shi and Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, June 1994.
- [2] D. Viswanathan, "Features from accelerated segment test (fast)," 2011.
- [3] 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.
- [4] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, 1988.

- [5] 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.
- [6] M. España, "Fundamentos de la navegación integrada," in *AADECA*, 2010.
- [7] M. S. Gageik, "An autonomous uav with an optical flow sensor for positioning and navigation," in *International Journal of Advance Robotic Systems*, 2013.
- [8] N. IRAM, "Iram-aadl j 20-14," in *Norma Argentina*, 2014.