

# Sistema de inspección y vigilancia utilizando un robot aéreo guiado mediante visión artificial

## Inspection and monitoring system using an aerial robot guided by artificial vision

**Andrés Felipe Silva Bohórquez**

Ingeniero Mecatrónico, Investigador Semillero de Investigación de Automatización, Robótica y Control-A&C  
Universidad de Pamplona, Colombia  
afsilvab@gmail.com

**César Augusto Peña Cortés**

Ph.D. Automática y Robótica  
Docente Tiempo Completo, Investigador Grupo Automatización y Control, Universidad de Pamplona, Colombia  
cesarapc@unipamplona.edu.co

**Luis Enrique Mendoza**

M.Sc. Ingeniería Biomédica  
Docente Tiempo Completo, Investigador Grupo de Ingeniería Biomédica, Universidad de Pamplona, Colombia  
luis.mendoza@unipamplona.edu.co

**Resumen—** Este artículo presenta el desarrollo de un sistema de inspección y vigilancia el cual procesa las imágenes de las cámaras abordo con el fin de realizar una navegación autónoma o semiautónoma, según como el usuario lo especifique. El dispositivo aéreo utilizado es un cuadrotor, al que se le implementaron varias estrategias de control servo-visual para determinar la orientación, la posición del mismo y así seguir una trayectoria demarcada. Las imágenes de la segunda cámara del robot aéreo son transmitidas por la red permitiendo la inspección remota del lugar que se desea confrontar. Se realizaron las verificaciones correspondientes a los controladores, demostrando que el robot aéreo sigue diversos tipos de trayectorias satisfactoriamente.

**Palabras clave—** Autónomo, inspección, navegación, robot aéreo, visión

**Abstract—** This paper presents the development of an inspection system and monitoring which processes the onboard camera images in order to perform an autonomous or semi-autonomous navigation, depending on how the user specified. The aerial device used is a quadrotor, which is implemented several strategies servo-visual control to determine the orientation, the position thereof and thus follow a path demarcated. The images of the second aerial robot camera are transmitted over the network allowing remote inspection of the place you want to confront. Checks were made corresponding to the controllers, showing that the robot follows diverse types of air paths satisfactorily.

**Keywords—** Autonomous, inspection, navigation, aerial robot, vision.

## 1. INTRODUCCIÓN

En las últimas décadas muchos vehículos te-leoperados y autónomos han sido desarrollados dentro del campo de la robótica y ampliado a otros más. Estos vehículos poseen orugas, patas y ruedas, pero a pesar de los grandes avances logrados en las últimas décadas, muchos vehículos terrestres poseen importantes limitaciones como, por ejemplo, acceder a lugares deseados debido a las características del terreno y la presencia de obstáculos que no se pueden evitar [1], [2].

En gran parte de estos casos, los vehículos aéreos son la mejor manera de acercarse al objetivo y obtener información o incluso realizar algunas acciones. La robótica aérea permite un enfoque útil para realizar tareas tales como adquisición de datos mediante imágenes, localización de objetivos, detección de incendios y áreas afectadas, seguimiento, publicidad, creación de mapas, defensa entre muchos otros servicios.

Los vehículos aéreos no tripulados (UAV) se han utilizado para aplicaciones militares, pero también son útiles para muchas aplicaciones civiles tales como la inspección de servicios públicos y de terrenos, prevención de desastres, vigilancia ambiental, búsqueda y rescate, cartografía aérea,

vigilancia el tráfico y la cinematografía. Sin embargo, el desarrollo de vehículos autónomos aéreos robotizados implica la resolución de varios problemas relacionados con la capacidad de carga, requisitos de seguridad, energía, resistencia de vuelo, autonomía, perturbaciones causadas por corrientes de aire, entre otros [3].

Los UAV son vehículos autopropulsados que se encuentran controlados a distancia o son capaces de realizar operaciones de forma autónoma. Los UAV en el campo de la investigación abarcan desde diseños de bajo nivel con algoritmos de control de vuelo hasta la múltiple coordinación de aeronaves de alto nivel. Durante las últimas décadas, se han dedicado importantes esfuerzos para aumentar la autonomía de vuelo, el control autónomo y la carga útil de los UAV.

La robótica aérea ha involucrado principalmente helicópteros, aviones, dirigibles y otros diseños de Vehículos de Despegue y Aterrizaje Vertical (VTOL, por sus siglas en inglés) donde la principal ventaja de estas plataformas es la maniobrabilidad, que es necesaria para muchas aplicaciones robóticas [4]. También se encuentran robots bioinspirados en aves e incluso murciélagos, los cuales vuelan gracias al movimiento vertical de sus alas (*flapping robots*). La capacidad de mantenerse en vuelo suspendido es muy importante en muchas tareas, sin embargo, son difíciles de controlar y requieren de operadores experimentados para su desarrollo y aplicación. Por otra parte, el control del vehículo totalmente autónomo no es tarea fácil y requiere el control una gran cantidad de variables.

En algunos casos la evolución de las plataformas convencionales va dirigida a los vehículos nuevos con mayor autonomía de vuelo y mayor capacidad de carga como la Yamaha R50 y Rmax [5]. La Rmax, es capaz de volar durante 1 hora con una carga útil de 24 kilogramos.

En el Laboratorio de Control Aeroespacial del Instituto Técnico de Massachusetts (MIT, por sus siglas en inglés) [6], [7] desarrolla un cuadrotor con inclinación variable, capaz de realizar maniobras acrobáticas que se extienden más allá de las capacidades de los modelos actuales de inclinación fija. En el Instituto Técnico de Zürich (ETH, por sus siglas en alemán) se realiza el proyecto de cuadrotor cooperativos, en donde estos realizan varias acción en conjunto. Estos vehículos experi-

mentan complejas fuerzas que los empujan a los límites de sus capacidades dinámicas.

La empresa alemana Festo ha desarrollado un vehículo aéreo no tripulado con forma de libélula, BionicOpter [8] cuenta con cuatro alas que operan de forma independiente, construidas en fibra de carbono, poliéster y aluminio, con ello consiguen que el conjunto pese menos de 175 gramos (el cuerpo está constituido por plástico ABS, aluminio y fibra). El dispositivo es capaz de volar hacia delante y hacia atrás, moverse lateralmente, flotar en un lugar, e incluso realizar vuelo sin motor.

Para el desarrollo de este proyecto se seleccionó el robot aéreo AR.Drone (Fig. 1) desarrollado por la empresa Parrot, el cual posee 4 hélices de alto desempeño (diseñadas específicamente para este producto), 4 motores sin escobillas (a 3500 rpm con una potencia de 15 W), estructura tubular de fibra de carbono, una batería de polímero de litio de 3 celdas (11,1 V, 1000 mAh) que brinda una autonomía de vuelo de aproximadamente 18 minutos mientras que su tiempo de carga es de 90 minutos.

Fig. 1. AR.DRONE



Fuente: Autor del proyecto

Posee dos cámaras, una frontal de lente gran angular a 93° (diagonal) con sensor CMOS, tiene una resolución de 640x480 pixeles y posee una detección en distancia de 30 cm a 5 m, transmitiendo video a 30 fps. La cámara vertical posee un lente a 64° (diagonal) con sensor CMOS, una resolución de 176x144 pixeles, transmitiendo video a 60 fps.

Incluye un altímetro por ultrasonido con una frecuencia de emisión de 40 KHz que le permite alcanzar hasta 6 metros de estabilidad vertical.

En el sistema informático integrado, el AR.Drone cuenta con un procesador ARM 9 a 468 MHz, una memoria DDR de 128 Mb a 200 MHz, genera una red Wi-Fi b/g, un puerto USB de alta velocidad y todo esto bajo el mando del sistema operativo Linux. El cuanto al sistema de orientación inercial posee un acelerómetro de 3 ejes, un girómetro de 2 ejes y un girómetro de precisión de viraje en un eje.

Este artículo se encuentra organizado de la siguiente forma: En la sección 2 se presentan los algoritmos de control los cuales incluyen el procesamiento digital de imágenes, el control de altitud, posición, orientación, la planeación de trayectorias y la interfaz. En la sección 3 se presentan un análisis de resultados de las pruebas experimentales y por último se presentan las conclusiones.

## 2. ALGORITMOS DE CONTROL

Las maniobras que realiza el dispositivo (Fig. 2) se obtienen mediante el cambio gradual de los ángulos Roll (giros en el eje X), Pitch (giros en el eje Y) y Yaw (giros en el eje Z) [9], [10].

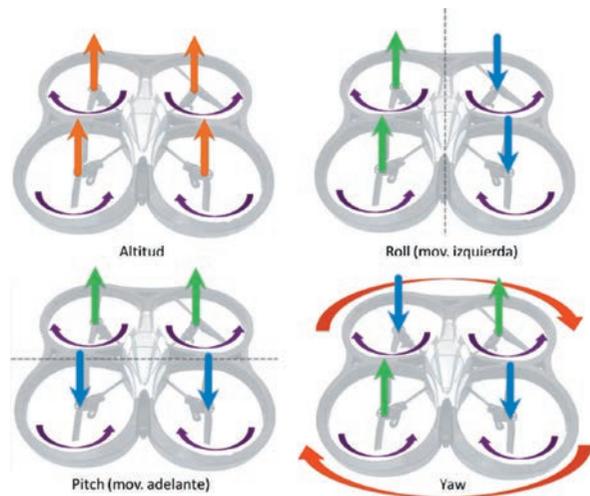
Fig. 2. ÁNGULOS DE NAVEGACIÓN



Fuente: Autor del proyecto

Para lograr estos giros, el robot varía las velocidades de sus rotores generando pares opuestos. En la Fig. 3 se aprecian los giros del robot y las flechas verticales indican la diferencia de velocidad con respecto a un valor fijo predeterminado para los rotores y que corresponde a la velocidad para la sustentación del robot [11], [12].

Fig. 3. MOVIMIENTOS DEL UAV



Fuente: Autor del proyecto

Al variar la velocidad de los rotores de la izquierda y de la derecha, se generan los movimientos de Roll, permitiéndole al robot avanzar hacia la izquierda o derecha. Al variar la velocidad de los rotores delanteros y traseros se producen los movimientos de Pitch haciendo que el robot avance o retroceda. Por último al variar la velocidad de los rotores opuestos se generan los movimientos de Yaw, permitiéndole girar sobre la vertical.

### 2.1. Funcionamiento general

Para especificar la trayectoria que debe seguir el robot se plantea el uso de señales o marcas en forma de círculo ubicadas en el suelo. Una vez se tiene indicado el camino se da inicio al software desarrollado el cual verifica la comunicación entre el cuadrotor y el ordenador, vía Wi-Fi, el nivel de batería, para dar inicio al despegue.

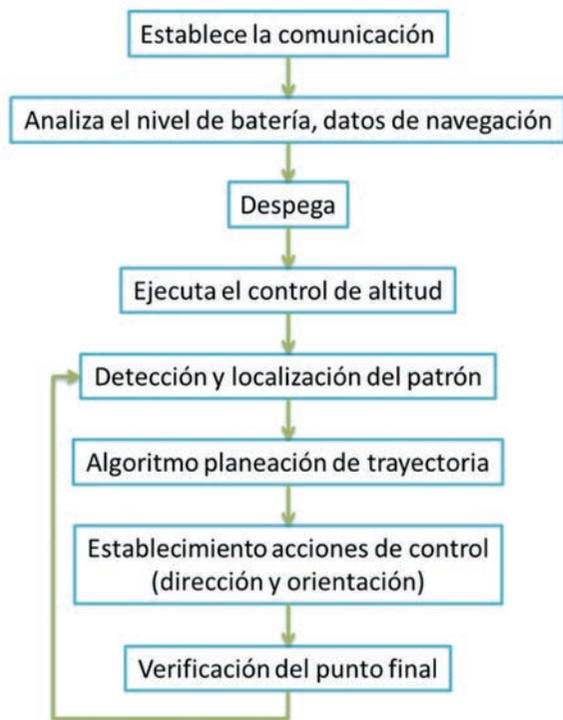
El cuadrotor puede iniciar con cualquier orientación, sin embargo es requerimiento que este se encuentre cerca de las marcas. Al despegar se ejecuta el controlador de altitud que eleva al robot a una altura adecuada.

Este controlador se basa en la medición del radio de las marcas por medio del procesamiento digital de las imágenes enviadas por la cámara inferior (entre más elevado este, las marcas se verán más pequeñas) y el procesamiento de los datos de los sensores de ultrasonido ubicados en parte inferior del robot. Se implementó este nivel de redundancia debido a que los obstáculos (plantas, sillas, escritorios, equipos) ubicados cerca de una

trayectoria típica, suelen afectar constantemente los sensores ultrasónicos.

La idea general es que el dispositivo varíe su altura hasta que el radio de las marcas (en píxeles) logre un valor idóneo indicando que el robot se encuentra a la altura adecuada.

Fig. 4. DIAGRAMA DE FUNCIONAMIENTO



Fuente: Autor del proyecto

Una vez la altitud se encuentra establecida (Fig. 4), se procede a la localización de los patrones y al cálculo de la distancia con respecto al dispositivo. Con el fin de dar inicio a los algoritmos de planeación de trayectorias, donde se hace que el robot se ubique sobre el patrón más cercano o se elige el patrón inicial y el final para realizar la trayectoria.

El algoritmo de planeación de trayectorias permite que el objetivo se encuentre desfasado tanto en orientación como en dirección lo cual corrige por medio de los controladores, el único requerimiento es que siempre se vea al menos una las marcas.

Sin importar la trayectoria que se le plante o se le exija al robot por medio de la disposición de los patrones, el dispositivo está en la capacidad de

resolver y ejecutar el circuito, ya sea que incluyan líneas rectas o curvas.

## 2.2. Detección y localización de los patrones

El proceso digital de imágenes se realiza mediante la utilización de la cámara vertical del AR.Drone, donde esta detecta un patrón específico. El algoritmo de procesamiento de imágenes se implementó haciendo uso de las librerías de OpenCV [13]. En primer lugar se realiza un algoritmo filtrado basado en color para conservar solamente los píxeles similares al color del patrón (en este caso círculos de color rojo), posteriormente se elimina el ruido con un filtro Gaussiano y se aplica la transformada de Hough [14] para la detección del patrón (Fig. 5), esta función debe ser sintonizada con la intensidad de luminosidad del ambiente en el que se encuentra con el fin de mejorar el rendimiento del algoritmo.

## 2.3. Transformada de Hough

La Transformada de Hough (TH), se define como una herramienta matemática para encontrar o clasificar contornos en una imagen. Esta técnica es muy útil en la detección de líneas, círculos y elipses, aunque puede ser implementada para la detección de cualquier figura paramétrica. La TH generalizada, está definida solo para trabajar en imágenes a nivel de gris y matemáticamente plantea una transformación paramétrica de la imagen. Por ejemplo, la ecuación de una recta está definida por (1), la idea general de TH es concentrarse en los parámetros  $(a, b)$  y no en los puntos de la imagen  $(y_i, x_i)$ .

$$y_i = a \cdot x_i + b \quad (1)$$

Dado que pueden existir líneas verticales, los parámetros pueden quedar indefinidos, por lo cual se suele trabajar en coordenadas polares tal como se presenta en (2)

$$x \cdot \cos(\varnothing) = \varrho \quad (2)$$

Teniendo en cuenta esto, se puede asociar a cada recta de la imagen, unos parámetros  $\varnothing$  y  $\varrho$  los cuales son únicos si:  $\varnothing \in [0, \pi)$  y  $\varrho \in \mathbb{R}$  o  $\varnothing \in [0, 2\pi)$  y  $\varrho \geq 0$ . Donde,  $\varnothing$  se define como el ángulo que forma la recta perpendicular a la recta formada por la ecuación 2 y  $\varrho$  es la distancia del punto  $(0,0)$  al punto  $(x,y)$  de la recta.

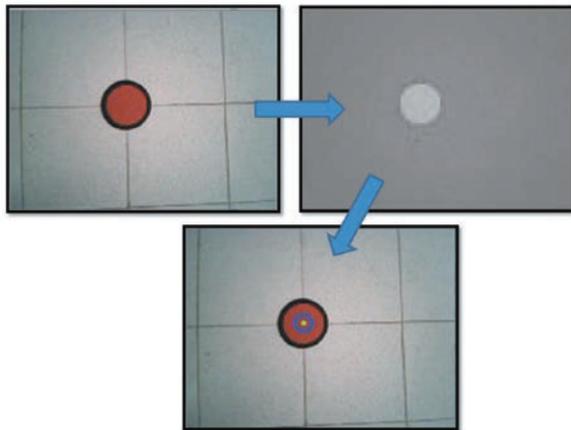
La TH se encarga de contar el número de repeticiones de cada uno de los parámetros correspondiente a cada punto. Los parámetros que posean un mayor número significativo de repeticiones indicaran una recta en la imagen.

Para el caso de las circunferencias se utiliza (3), donde  $a, b$  representa el centro del círculo y  $r$  el radio. Por lo tanto, este caso se utiliza un acumulador de tres dimensiones  $(a, b, r)$ .

$$(x-a)^2 + (y-b)^2 = r^2 \quad (3)$$

En la Fig. 5 se presenta el filtrado de la imagen y la detección de uno de los patrones utilizados indicando su centro por medio de una marca compuesta por un punto y círculo concéntrico de menor tamaño.

Fig. 5. DETECCIÓN E IDENTIFICACIÓN DEL PATRÓN



Fuente: Autor del proyecto

## 2.4. Planeación de trayectorias

Una vez detectados los patrones se procede a evaluar la acción a tomar según la cantidad que se detecten. Las reglas implementadas son las siguientes:

Al encontrar un solo patrón, el robot aéreo toma como objetivo posarse sobre el mismo, en caso de perturbaciones (ejemplo: viento) que muevan el robot, el controlador calcula la distancia al objetivo y si supera un umbral establecido (máximo error permitido), se reactivan las acciones de control corriendo el error.

En el caso de detectar dos patrones, se selecciona uno como posición inicial de la trayectoria (patrón más cercano a la parte inferior de la imagen) y el otro como posición final u objetivo.

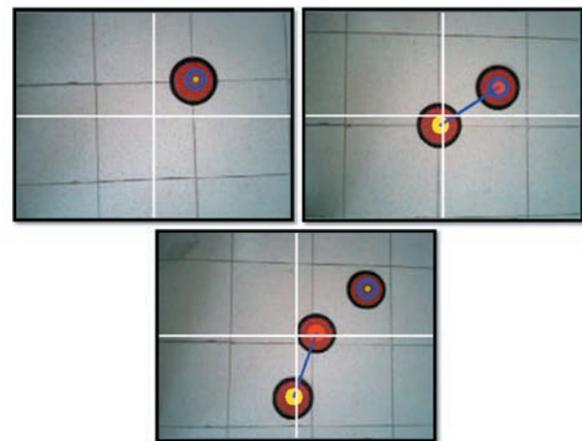
El controlador guiará el robot hacia el objetivo (control de posición). Adicionalmente, al existir más de un patrón se puede ejecutar el control de orientación, de tal forma, que el robot además de posicionarse, avanza con una orientación predefinida (por defecto la orientación se fija para que la cámara frontal coincida con el sentido de la trayectoria).

El control de orientación toma como referencia el número de columnas en la imagen que separan la posición objetivo con la inicial. Si el número es cercano a cero significa que el robot posee la orientación correcta, de lo contrario tendrá que girar sobre el eje  $z$  (Yaw) hasta lo logre.

Al detectar 3 patrones o más patrones, toma los dos más cercanos como referencia para controlar la orientación y el tercero lo selecciona como punto objetivo para el control de posición.

En la Fig. 6 se ilustran los tres casos. El círculo azul representa el objetivo del control de posición, el círculo amarillo, el círculo rojo y la línea azul la referencia en orientación del robot. El robot debe girar hasta que la línea azul este casi vertical.

Fig. 6. SELECCIÓN DE LA TRAYECTORIA SEGÚN LOS PATRONES DETECTADOS



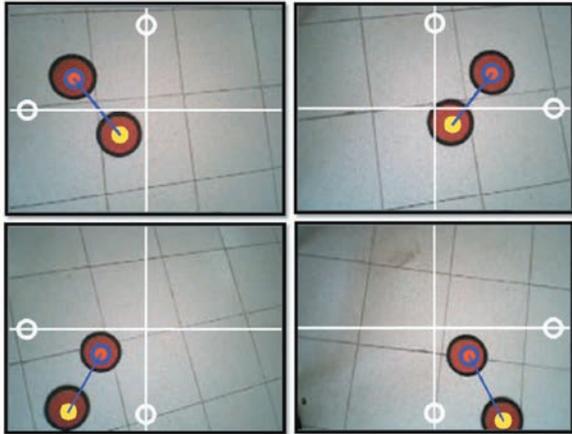
Fuente: Autor del proyecto

## 2.5. Establecimiento de las acciones de control

La corrección de la posición está dada por la ubicación de los patrones detectados. Dependiendo del cuadrante en el que se encuentre el patrón (objetivo) identificado por el algoritmo de detección, se ejecutan las acciones de control las cuales pueden ser: avanzar, retroceder, ir a izquierda o derecha.

En la Fig. 7 se ilustran cuatro casos, donde el robot tiene que desplazarse en dos direcciones simultáneamente (avanzar-retroceder e ir a la izquierda-derecha). Lo círculos en blanco indican el sentido de la acción de control.

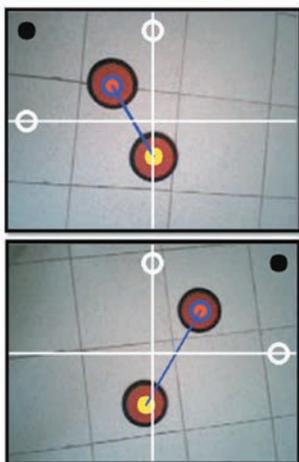
Fig 7. CORRECCIÓN DE LA POSICIÓN



Fuente: Autor del proyecto

En la Fig. 8 se puede apreciar el uso simultáneo del control de posición y orientación del robot. Donde es importante recalcar que el círculo negro en la parte superior de la imagen indica el sentido de la acción de control para corregir la orientación del robot. Si está a la derecha indica que el robot está girando en sentido horario y si está a la izquierda lo hará en sentido anti-horario. La toma de esta decisión se hace teniendo en cuenta la posición horizontal relativa (número de columnas a derecha o izquierda) entre el centro del patrón inicial (punto de partida) y el patrón objetivo.

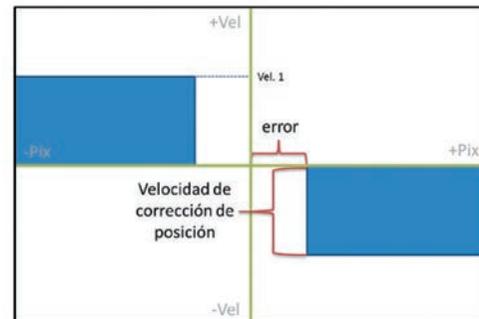
Fig. 8. CORRECCIÓN DE LA ORIENTACIÓN



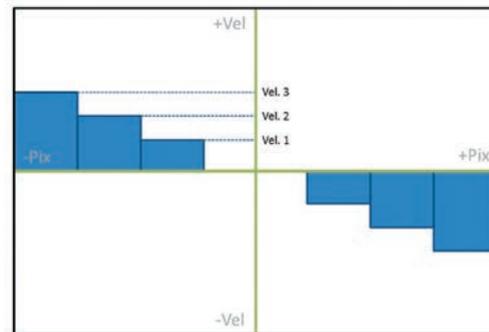
Fuente: Autor del proyecto

Para la realización de los movimientos del robot tanto en posición como en orientación, se implementaron tres tipos de controladores: on-off, escalonado y proporcional [15]. En la Fig. 9 (a). Se aprecia el con on-off donde básicamente se aplica una velocidad constante positiva o negativa para realizar el movimiento deseado. Se puede apreciar una zona donde no se aplica acción de control la cual corresponde con la zona de error máximo permitido.

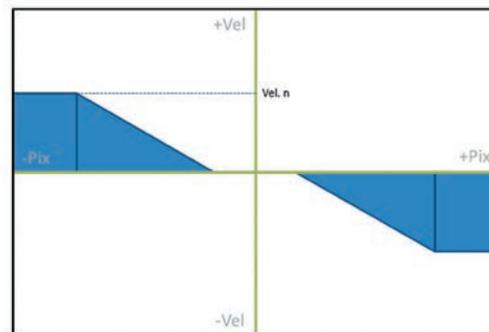
Fig. 9. A) CONTROLADOR ON-OFF, B) CONTROLADOR ESCALONADO Y C) CONTROLADOR PROPORCIONAL



a)



b)



c)

Fuente: Autor del proyecto

En la Fig. 9 (b), se ilustra un control escalonado donde en vez de aplicar un solo valor de velocidades, se aplican tres valores diferentes, según la

magnitud del error. Por último en la Fig. 9 (c) se muestra el control proporcional, el cual varía la velocidad del movimiento de forma proporcional al error. Cabe aclarar que este controlador también cuenta con una zona de máximo error permitido y una velocidad máxima.

Con estos controladores el robot adquiere la habilidad de seguir la trayectoria que se encuentre especificada en el suelo, sin importar la forma o la complejidad en que se implemente, ya sea una línea recta, un círculo, un zigzag, etc.

## 2.6. Interfaz de usuario

La aplicación fue diseñada bajo el lenguaje de C++/CLI, a través de la cual el AR.Drone envía la información de navegación y los paquetes de video al ordenador. Este realiza el procesamiento digital de imágenes y envía los comandos de estado para que el dispositivo realice una determinada acción.

Consiste en una interfaz sencilla e intuitiva que maneja información en tiempo real (Fig. 10). Esta interfaz está compuesta por los siguientes paneles:

- **Comunicación:** Inicializa y finaliza la comunicación entre el AR.Drone y el ordenador.
- **Visualización:** Muestra la imagen capturada por la cámara frontal del AR.Drone y a su vez el procesamiento digital de imágenes que se realiza a partir de la cámara vertical inferior.
- **Datos de navegación:** Muestran los valores de altitud, nivel de batería, ángulos de navegación y velocidades de navegación en cada eje.
- **Modo autónomo:** El AR.Drone realiza el circuito de forma autónoma según la trayectoria definida por los patrones que se encuentren en el suelo.
- **Modo manual:** El operador puede controlar el dispositivo mediante el teclado del ordenador o los botones dispuestos en la interfaz.
- **Paro de emergencia:** En caso de una falla o eventualidad, el usuario puede activar el paro de emergencia para bloquear las hélices y aterrizar el AR.Drone de forma inmediata, no se recomienda su uso, salvo en casos ineludibles.

Fig. 10. INTERFAZ DE USUARIO



Fuente: Autor del proyecto

## 3. ANÁLISIS DE RESULTADOS

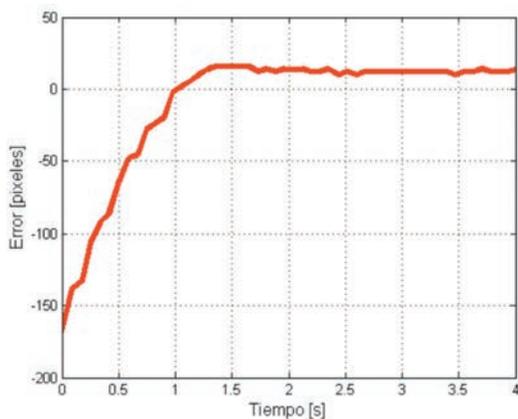
Para la verificación experimental de los controladores se realizaron diversas pruebas que incluyen el posicionamiento del robot sobre un patrón (control de posición), el cambio de orientación usando al menos dos patrones (control de orientación) y el seguimiento de trayectorias completas, obteniendo resultados muy satisfactorios a pesar de las perturbaciones tales como el viento y los cambios de iluminación.

En la Fig. 11 se ilustran los resultados de unas pruebas, aplicando el control de orientación del robot. Para estas pruebas se dispuso de tres patrones ubicados en línea recta y se ponía a despegar el robot con una orientación inicial de 90 grados con respecto a la trayectoria.

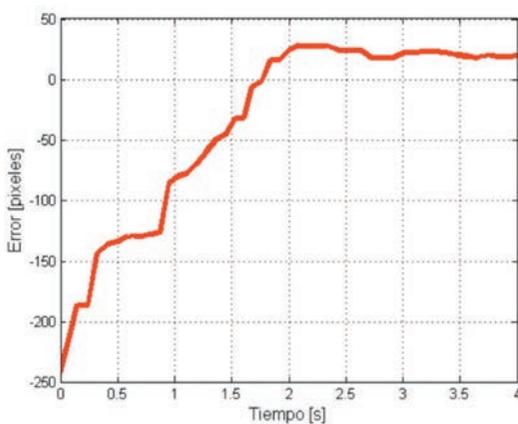
En la Fig. 11 (a) se aprecia el resultado utilizando en controlador on-off. Debe recordarse que este controlador tiene como realimentación la diferencia entre la posición horizontal (en píxeles) de los patrones que estén más cerca del robot. Se aprecia que el robot se orienta rápidamente, sin embargo existe un error en estado estable.

En la Fig. 11 (b) se ilustran el resultado haciendo uso del control escalonado, donde se aprecia que el robot también se orienta de forma rápida. Obsérvese que el error inicial es mayor que en el caso anterior. Debe tenerse en cuenta que el error inicial es variable debido a que cuando el robot despegue este cambia su posición dependiendo de las perturbaciones (viento, nivel de batería entre otros).

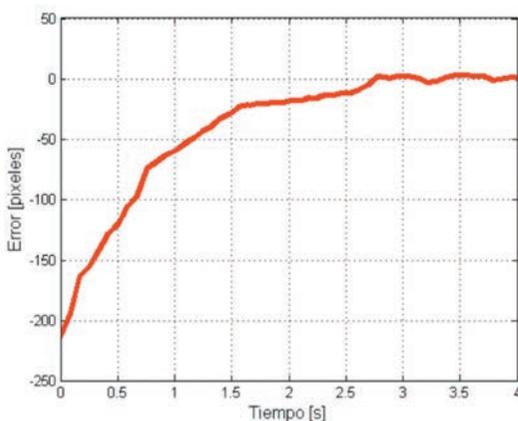
Fig. 11. A) CONTROLADOR ON-OFF, B) CONTROLADOR ESCALONADO Y C) CONTROLADOR PROPORCIONAL



a)



b)



c)

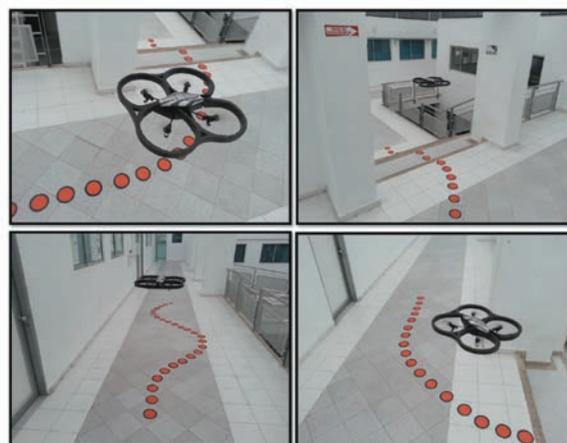
Fuente: Autor del proyecto

En la Fig. 11 (c) se presenta el resultado obtenido con el control proporcional del robot. En este se puede apreciar como el robot se orienta con mayor precisión. Esto se debe a que la zona de

máximo error permitido se puede reducir significativamente.

De forma análoga y con resultados muy similares se realizaron las pruebas del control de posición, en el cual se hacía despegar el robot a una distancia fija de un patrón y se examinaba el posicionamiento del robot sobre el mismo. Por lo cual se optó por el uso del control proporcional.

Fig. 12. PRUEBAS SEGUIMIENTO TRAYECTORIA



Fuente: Autor del proyecto

En cuanto a recorridos completos, el cuadrotor desarrolló todas las trayectorias que se implementaron con los patrones, tanto en línea recta, como en forma de zigzag, subiendo o bajando las escaleras e incluso atravesando por el marco de una puerta, en la Fig. 12, se ilustran algunos de estos recorridos.

#### 4. CONCLUSIONES

Se logró diseñar un sistema de inspección y/o vigilancia que hace uso de un cuadrotor, para que un operario lo controle de forma remota o funcione autónomamente y envíe las imágenes capturadas con sus dos cámaras.

Se determinó que es posible establecer un algoritmo seguimiento de rutas para un robot aéreo haciendo uso de patrones circulares y visión artificial.

A pesar que las características de las cámaras abordo del dispositivo son muy limitadas, fueron suficientes para realizar el proceso de visión artificial, identificando patrones y brindando la información suficiente para que el proyecto fuera viable.

El algoritmo propuesto de planeación de trayectorias demostró ser bastante robusto y muy fá-

cil de implementar, lo que implica un bajo requerimiento computacional para su desarrollo.

La interfaz de usuario desarrollada se caracterizó por un diseño muy intuitivo y sencillo de emplear por personas con pocos conocimientos en el área.

Se escogió el controlador tipo proporcional para la corrección de la posición y orientación debido a que la respuesta fue satisfactoria y mucho más estable en comparación con el controlador on-off y escalonado. Aunque es posible implementar controladores más avanzados este cumplió ampliamente con los requerimientos establecidos para aplicación, lo cual se evidencia con el desarrollo de las diversas trayectorias a las cuales fue sometido.

### REFERENCIAS

- [1] R. A. Sasongko, T. Mulyanto, A.H. Wijaya, "The development of an autonomous control system for a small UAV: Waypoints following system," Asian Control Conference 7th, pp.308,313, 27-29 Aug. 2009.
- [2] H. Dong-in, K. Jong-Hun, L. Dae-Woo, C. Kyum-rae, C. Sung-jin, "Autonomous flight test using angle of UAV's velocity vector," Control, Automation and Systems. International Conference on, vol., no., pp.312,315, 14-17 Oct. 2008.
- [3] M. Byoung-Mun, L. Il-Hyung, H. Tae-Won, H. Jin-Sung, B. Hyo-choong, T. Min-Jea, "Unmanned autonomous helicopter system design and its flight test," Control, Automation and Systems. International Conference on, vol., no., pp.2090,2095, 17-20 Oct. 2007.
- [4] Z. Zhiyang, H. Fei, L. Jian, "Autonomous flight control system designed for small-scale helicopter based on approximate dynamic inversion," Advanced Computer Control (ICACC). 3rd International Conference on, vol., no., pp.185,191, 18-20 Jan. 2011.
- [5] A. Bogdanov, E. Wan, G. Harvey, "SDRE flight control for X-Cell and R-Max autonomous helicopters," Decision and Control, 2004. CDC. 43rd IEEE Conference on, vol.2, no., pp.1196,1203 Vol.2, 14-17 Dec. 2004.
- [6] K. Peng, M. Dong, M. Chen Ben, G. Cai, Y. Lum Kai, T.H Lee, "Design and Implementation of a Fully Autonomous Flight Control System for a UAV Helicopter," Control Conference, 2007. CCC 2007. Chinese, vol., no., pp.662,667, July 26 2007-June 31 2007.
- [7] T. Matsukawa, S. Arai, K. Hashimoto, "Autonomous flight of small helicopter with real-time camera calibration," System Integration (SII), 2010 IEEE/SICE International Symposium on, vol., no., pp.242,247, 21-22 Dec. 2010.
- [8] C.D.Hendrix, M.J Veth, R.W Carr, "LQG control design for a hovering micro air vehicle using an optical tracking system," Aerospace conference, 2009 IEEE, vol., no., pp.1,14, 7-14 March 2009.
- [9] S. Saito, B. Yue, T Mochizuki, "Autonomous flight control for RC helicopter using camera image," SICE, 2007 Annual Conference, vol., no., pp.1536,1539, 17-20 Sept. 2007.
- [10] C. Longsheng, J. Yang, W. Changkun, "Adaptive visual servo control of UAV Ground-Target-Autonomous-Tracking System," Intelligent Control and Automation (WCICA), 2012 10th World Congress on, vol., no., pp.133,137, 6-8 July 2012.
- [11] J. Zhou, X.M. Wang, X.Y Wang, "Automatic Landing Control of UAV Based on Optical Guidance," Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on, vol., no., pp.152,155, 23-25 Aug. 2012.
- [12] J. Stowers, M. Hayes, A. Bainbridge-Smith, "Beyond optical flow – Biomimetic UAV altitude control using horizontal edge information," Automation, Robotics and Applications (ICARA), 2011 5th International Conference on, vol., no., pp.372,377, 6-8 Dec. 2011.
- [13] V. Grabe, H.H Bulthoff, P.R Giordano, "On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow," Robotics and Automation (ICRA), 2012 IEEE International Conference on, vol., no., pp.491,497, 14-18 May 2012.
- [14] C. D. Hendrix, M.J Veth, R.W Carr, "LQG control design for a hovering micro air vehicle using an optical tracking system," Aerospace conference, 2009 IEEE, vol., no., pp.1,14, 7-14 March 2009.
- [15] A. Cesetti, E. Frontoni, Q. Mancini, P. Zingaretti, "Autonomous safe landing of a vision guided helicopter," Mechatronics and Embedded Systems and Applications (MESA), 2010 IEEE/ASME International Conference on, vol., no., pp.125,130, 15-17 July 2010.