

Interfaz para monitoreo y navegación controlada de un robot aéreo tipo cuatrimotor

Luis Alejandro Zetina de la Cruz, Antonio Benitez Ruiz, Jorge de la Calleja Mora, María Auxilio Medina, Ismael Everardo Bárcenas
Universidad Politécnica de Puebla, Departamento de Posgrado, 3er Carril del Ejido Serrano S/N, San Matéo Cuanalá, Juan C. Bonilla, Puebla CP 72640 e-mail: {Luis.zetina, Antonio.benitez, Jorge.delacalleja, ismael.barcenas}@uppuebla.edu.mx, mauxmedina@gmail.com

Resumen

La robótica aérea es de gran relevancia por su versatilidad de aplicaciones. Este trabajo presenta el desarrollo de una interfaz gráfica de usuario para el control de navegación de un cuatrimotor. Este desarrollo es la base de un sistema de exploración autónoma y mapeo de objetos de interés en un ambiente controlado. Así en éste artículo se propone una interfaz que permitirá: recuperar la información de los sensores con que cuenta un robot aéreo tipo cuatrimotor, proporcionar funcionalidades para controlar la locomoción del robot, disponer de un área para el despliegue de las imágenes capturadas a través de la cámara y finalmente aproximar su localización en un ambiente tridimensional

1. Introducción

La investigación y desarrollo de la robótica se ha incrementado debido al interés en su amplio campo de aplicación, que incluye la milicia, aplicaciones civiles como vigilancia, búsqueda y rescate, y de investigación como la exploración y mapeo. Una de las aéreas de reciente aplicación es la relacionada con la robótica aérea. La robótica aérea tiene como objetivo brindar acceso rápido a lugares remotos y mayor rapidez en exploración de grandes extensiones de áreas. La aeronave no tripulada o UAV (“Unmanned Aerial Vehicle”) es capaz de realizar una misión sin necesidad de contar con un piloto a bordo para su navegación, pero sin que esto restrinja la posibilidad de que se cuente con un piloto o controlador de vuelo desde tierra. Existe también la aeronave capaz de realizar una misión sin necesidad de alguna interacción, llamada aeronave autónoma o sistema aéreo autónomo (AAS: Autonomus Aerial System) [1].

Por otro lado, los robots aéreos se clasifican de acuerdo al tipo de despegue [1]: vertical y no vertical.



Figura 1. Robot AR.Drone: Robot aéreo tipo cuatrimotor utilizado en este proyecto [6].

Dentro de los de despegue vertical se tienen los de hélice (helicópteros y cuatrimotores) y los auto-sustentados (dirigibles y globos). En los de despegue no vertical se encuentran los de ala flexible (parapente y ala delta) y los de ala fija (aeroplanos). El cuatrimotor es el más utilizado para los desarrollos de aplicaciones civiles, debido a las ventajas que presenta, como las mencionadas por Hoffmann et al. [2] entre las que destacan la mejor maniobrabilidad en espacios pequeños, que es muy limitada en aeronaves de ala fija, como por ejemplo, la inspección de edificios y puentes para la detección de grietas, la detección de municiones sin detonar, baliza de seguimiento de rescate. Como menciona Toledo et al. [9] su versatilidad principalmente se debe a que puede realizar rotaciones y desplazamientos en cualquiera de los tres ejes. El objetivo de este trabajo es desarrollar una interfaz gráfica de usuario (GUI del inglés: *Graphical User Interface*) que permita la comunicación, la recuperación de datos y el envío de comandos hacia un robot aéreo tipo cuatrimotor a través de un sistema de cómputo basado en GTK y plataforma Linux.

Para esto, el artículo se presenta de la siguiente manera: en la sección 2, se describen las características del robot aéreo utilizado; en la sección 3, se presentan las herramientas de software utilizadas para programar el cuatrimotor, además la interfaz propuesta se describe en la sección 4. Por último en la sección 5, se presentan los resultados preliminares obtenidos en este trabajo y las conclusiones.

2. Robot aéreo tipo cuatrimotor

El sistema está compuesto de un robot aéreo que cuenta con sensores y actuadores. También se desarrolló una GUI con comandos básicos para el funcionamiento del robot. En este trabajo, se utilizó el robot aéreo de tipo cuatrimotor de radiocontrol llamado AR.Drone como el que se muestra en la Figura 1, de la empresa francesa Parrot [6]. Este robot dispone además, con una interfaz de monitoreo de sensores, la cual se encuentra disponible dentro de los códigos ejecutables del kit de desarrollo de software o SDK (Software Development Kit).

El AR.Drone es un cuatrimotor de radiocontrol desarrollado por la empresa Francesa Parrot y fue presentado en la feria International Consumer Electronics Show (CES) en Las Vegas en 2010 y la nueva versión fue presentada en la misma del 2012 con buenas expectativas de los visitantes, como expresa la revista Muy interesante en su versión en línea [8]; éste cuenta con un microprocesador y un conjunto de sensores, al igual que un conector Wi-fi para la comunicación con dispositivos móviles con sistemas operativos IOS, Android o Linux, que le permitan controlarlo y recibir los datos de lectura de los sensores del AR.Drone. En la Tabla 1 se representan de manera general las características técnicas del AR.Drone.

Tabla 1. Características técnicas del AR.Drone [3]

Procesador	ARM9 RISC 32 Bits a 468 MHz	128 MB RAM
Conexiones	Inalámbrica Wi-Fi (b/g)	Universal Serial Bus (USB)
Sensores	3 acelerómetros	2 giroscopios
	1 giroscopio yaw de precisión	Altímetro ultrasónico rango de 6 m
Cámaras	Frontal de lente angular	Vertical de lente angular de 93°
	Resolución 640 x 480 pixeles	Resolución 176 x 144 pixeles
	15 fps	60 fps
Sistema Operativo	Linux	

2.1 Características y sensores

Este robot contiene diversos sensores de movimiento que se encuentran por debajo del casco central. El AR.Drone cuenta con seis grados de libertad, tiene integrado sistemas de orientación inercial con microsistema electromagnético (MEMS), que incluye un acelerómetro de tres ejes, un girómetro de dos ejes y un girómetro de precisión de viraje de un eje. También dispone de un altímetro ultrasónico que proporciona medidas de altitud para la estabilización automática de la altitud y ayuda al control de la velocidad vertical. Una cámara apuntando hacia el suelo proporciona medidas de velocidad de avance para el vuelo. La estructura mecánica del AR.Drone comprende cuatro rotores unidos a los extremos de una cruz en la que se unen la batería y el hardware (Figura

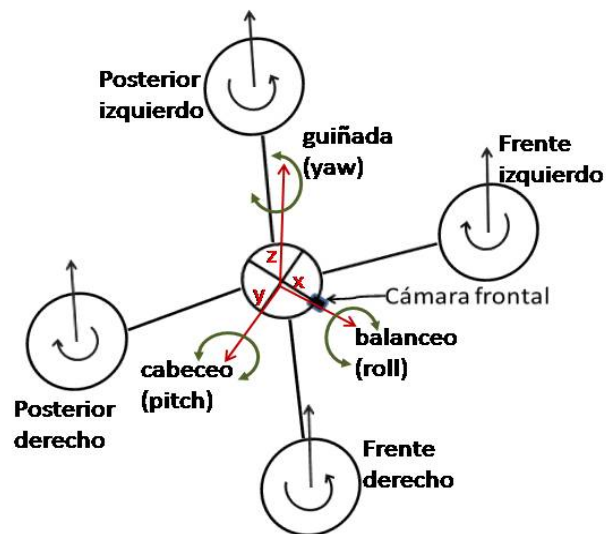


Figura 2. Rotación de rotores en pares; posicionamiento de los ejes cartesianos con respecto al cuatrimotor.

1). Cada par de rotores opuestos está girando de la misma manera. Un par está girando en sentido horario y el otro en sentido contrario, como se muestra en la Figura 2.

Dependiendo las velocidades de los rotores, el robot puede hacer movimientos en los tres ejes dimensionales. Los vehículos aéreos pueden realizar rotaciones en los tres ejes dimensionales [4], como se puede apreciar en la Figura 3. Estos movimientos se pueden describir:

- *Yaw Axis* (eje de guiñada): es perpendicular al plano de las alas (en su caso). Un movimiento de guiñada es un movimiento del frente de la aeronave de lado a lado.
- *Pitch Axis* (eje de cabeceo): es perpendicular al eje de guiñada y es paralelo al plano de las alas con su origen. Un movimiento de cabeceo es un

movimiento hacia arriba o hacia abajo del frente de la aeronave.

- *Roll Axis* (eje de alabeo o balanceo): es perpendicular a los otros dos ejes. Un movimiento de balanceo es un movimiento hacia arriba y abajo de los costados de la aeronave.

Los movimientos se logran variando la velocidad de los rotores. Para el movimiento de balanceo las aceleraciones de los rotores de la izquierda y derecha se hace de manera diferente. En el caso de la operación para avanzar o retroceder los rotores a los que se les varía la aceleración son los delanteros y traseros; y en el caso del giro a la izquierda o derecha (giro de guiñada) se varían las aceleraciones de los rotores en pares opuestos, la Tabla 2 especifica los rotores actuadores que llevan a cabo cada movimiento de navegación del AR.Drone.

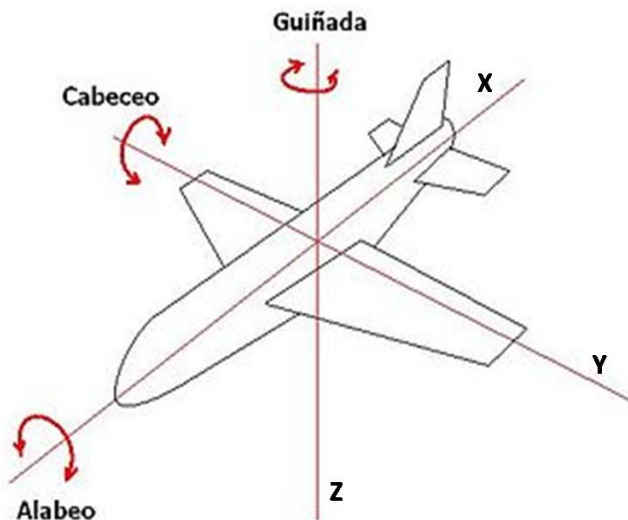


Figura 3. Rotaciones de aeronaves [3]: Guiñada (Yaw) en el eje Z, cabeceo (Pitch) en el eje Y y balanceo (Roll) en el eje X.

3. Herramientas de software

Una ventaja importante de este cuatrimotor es su capacidad de ser programable. Así, este robot puede ser programado a través del SDK especialmente desarrollado para este propósito y GTK, bajo el sistema operativo Ubuntu basado en Linux. El SDK [3] de código abierto que se encuentra disponible permite a cualquier desarrollador de software, desarrollar y distribuir nuevos juegos basados en el AR.Drone. El SDK incluye lo siguiente [3]:

- Documento que explica cómo utilizar el SDK y describe los protocolos de comunicación del AR.Drone.

- Biblioteca AR.Drone (ARDroneLIB), que proporciona las API necesarias para comunicarse y configurar el AR.Drone.
- La biblioteca herramienta AR.Drone (ARDroneTool), que proporciona un cliente dron completamente funcional donde los desarrolladores sólo tienen que insertar su código de aplicación específica [especifica](#).

Tabla 2. Relación de los movimientos que puede realizar el AR.Drone, sobre que eje actúa y que rotores (véase Figura 2), incrementan su velocidad para realizar el movimiento.

Eje de acción	Movimiento	Rotores actuadores
Balanceo	Avance derecha	Frente derecho-posterior derecho
	Avance izquierda	Frente izquierda-posterior izquierda
Cabeceo	Avance adelante	Frente izquierdo-frente derecho
	Avance atrás	Posterior izquierdo-posterior derecho
Guiñada	Giro derecha	Frente derecho-posterior izquierdo
	Giro izquierda	Frente izquierdo-posterior derecho
Velocidad vertical (eje z)	Ascenso	Los cuatro rotores incrementan su velocidad
	Descenso	Los cuatro rotores disminuyen su velocidad
	Estacionario	Los cuatro rotores a la misma velocidad

Para el desarrollo de la interfaz se utiliza GTK (“GIMP Tool Kit”) o kit de herramientas GIMP (“GNU Image Manipulation Program”) como software de desarrollo, la cual es una herramienta multiplataforma para crear interfaces gráficas de usuario [7]. Estas herramientas permiten construir una GUI, la cual se convierte en el medio por el cual una persona o usuario de un sistema puede comunicarse con los dispositivos o máquinas que permita el mejor aprovechamiento del sistema y la explotación de sus funcionalidades.

Para el desarrollo de la GUI fue necesario considerar las herramientas de desarrollo disponibles, la información necesaria a desplegar, las funcionalidades y los medios de interacción. Se utilizó una multiplataforma GTK, con una API (Interfaz de Programación de Aplicaciones), lo que ayuda a acelerar el tiempo de desarrollo. Esta biblioteca está escrita en C, pero fue diseñada para apoyar una amplia gama de lenguajes como C/C++, Perl y Python, C#,

JAVA y PHP, entre otros. GTK es una herramienta de software libre, parte del proyecto GNU [5].

Un elemento más que forma parte de GTK es GTK-Cairo, que es la biblioteca que brinda las herramientas necesarias para poder realizar dibujo de líneas y áreas utilizando diferentes colores, también se puede utilizar para el manejo de video. Esta biblioteca es utilizada en la sección de los botones de mando de la interfaz para dibujar botones que sean intuitivos para el usuario. La herramienta GTK es compatible con el SDK del AR.Drone realizado en lenguaje C. La interfaz está integrada por un conjunto de pequeñas aplicaciones llamados Widget, como ventanas, botones, etiquetas, entre otras, manejados por GTK.

4. Diseño de interfaz

El API SDK que se describió en la sección anterior utilizado para desarrollar la interfaz, considera las funcionalidades necesarias para la operación y monitoreo del cuatrimotor desde la interfaz propuesta mostrada en la Figura 4 y a continuación se describen las secciones consideradas para su desarrollo:

- Comandos de locomoción: Para la interfaz es necesario contar con comandos, que permitan realizar operaciones elementales de navegación con el AR.Drone, para lograrlo se implementó una botonera en la interfaz, ordenando los botones en pares de tal forma que cada par correspondan a movimientos realizados sobre el mismo eje:
 - Botones de desplazamiento vertical (Figura 4, sección a): estos botones modifican la altura de vuelo del drone actuando sobre el eje z . Envían el comando de aceleración o desaceleración de la velocidad vertical produciendo la elevación del cuatrimotor o

el descenso.

- Botones de desplazamiento bidireccional (Figura 4, inciso b): realizan el desplazamiento hacia adelante o hacia atrás del drone. Estos botones modifican la velocidad del drone produciendo un movimiento sobre el eje x .
- Botones de desplazamiento lateral (Figura 4, inciso c): estos botones envían el comando de locomoción para modificar la velocidad que actúa sobre el eje y para realizar un movimiento lateral hacia la izquierda o derecha.
- Botones de giro (Figura 4, inciso d): envía la instrucción para realizar giro a la derecha o a la izquierda, el giro actúa sobre el eje z , ocasionando un movimiento de torque.
- Botón stop (Figura 4, inciso e): envía el comando de locomoción con los parámetros que le indiquen al drone realizar un vuelo estacionario.
- Botón de emergencia (Figura 4, inciso e): este botón envía al drone el comando para realizar un aterrizaje inmediato y reiniciar el sistema.

- Información sensorial: Es necesario contar con una sección donde se visualicen algunas de las lecturas de los sensores como las velocidades en los tres ejes x , y , z , la elevación del robot y el porcentaje de carga de la batería, ya que de ésta depende el buen funcionamiento del robot.
- Autocalización del robot: Debido a que el cuatrimotor puede navegar en un ambiente en tres dimensiones, se debe tener presente los movimientos que el robot puede realizar en cada uno de ellos; así, se requiere una sección donde se visualicen los valores de posición (x , y , z) y la orientación, estos datos son recuperados de

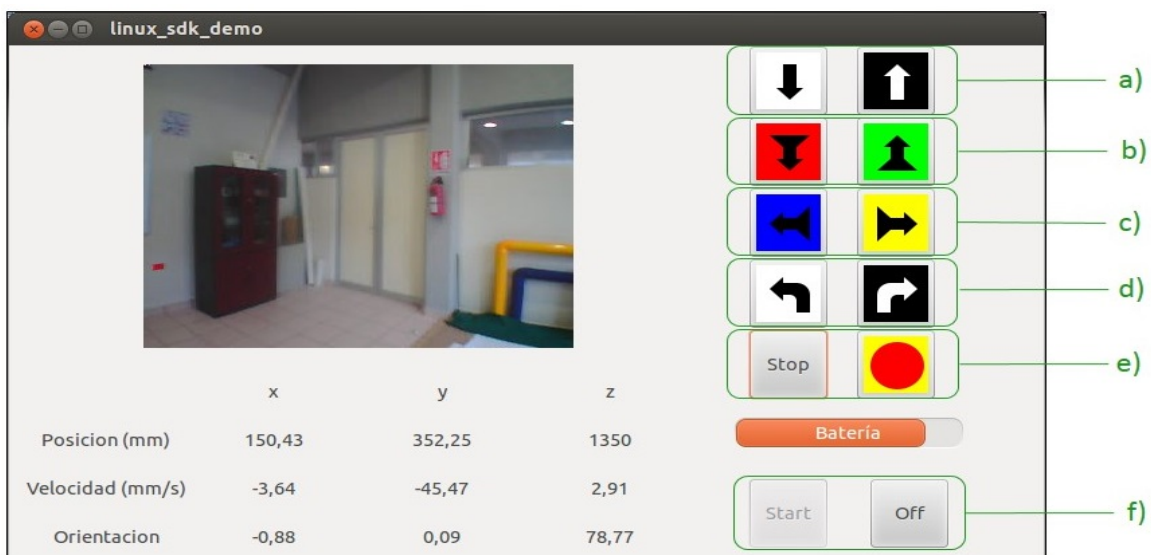


Figura 4: Interfaz para el monitoreo y control del AR.Drone, que incluye: botones para la locomoción del robot, video recuperado desde la cámara y sección en donde se muestra el estado de los sensores.

los sensores con los que cuenta el dron.

- Despliegue de imágenes recuperadas: Por último se requiere una sección donde se desplieguen las imágenes recuperadas de la cámara que tiene al frente el AR.Drone.
- Sección de inicio-fin de operación del AR.Drone (Figura 4, inciso f): estos botones start-off, inician el sistema del dron lo que permite enviar las lecturas de los sensores y también permite que los comandos de locomoción enviados por la interfaz sean ejecutados por el dron o en su caso realizar el aterrizaje del dron, desactivar la operación de los motores y de cualquier comando de locomoción recibido.

5. Resultados y conclusiones

Se probó el cuatrimotor tanto en términos de locomoción (movimiento) como de la recuperación de la información que se genera a través de los sensores, lo cual incluye la cámara. Puesto a punto el robot, se diseñó e implementó una interfaz de usuario. Esta interfaz incluye un área en donde se puede manipular el robot, una en la que se pueden verificar los valores de las lecturas de los sensores y una más en donde se presenta la imagen que el robot recupera con la cámara frontal.

5.1 Resultados

En el área de sensores las lecturas están calculadas para cada uno de los ejes del plano cartesiano. Se tiene una sección que visualiza la velocidad que detectan los acelerómetros dadas en milímetros por segundo (mm/s). La orientación es dada en grados, siendo el punto de partida la orientación que el cuatrimotor tenía al momento de iniciar su sistema. La posición para los ejes x , y es calculada en base al avance que calcula un algoritmo implementado por la compañía desarrolladora que utiliza las imágenes tomadas por la cámara ventral; para el caso de la posición en el eje z , se refiere a la altitud del dron y es calculada por un sensor ultrasónico, las medidas de desplazamiento están dadas en milímetros. Todos los datos mostrados en la sección de sensores de la interfaz y el video, son actualizados constantemente, tomando las actualizaciones de las lecturas de los sensores que el SDK registra.

5.2 Conclusiones

Así la interfaz propuesta se muestra en la Figura 4. La interfaz permite realizar un control de navegación más interactivo para el usuario y visualizar algunos de

los datos recuperados por los sensores, así como tener monitoreado en todo momento las velocidades, el desplazamiento y orientación del AR.Drone. Esta interfaz se ha probado en ambientes cerrados obteniéndose buenos resultados, la imagen de la cámara es visualizada en todo momento y se pueden observar regiones de difícil acceso o que requieran del vuelo estacionario del robot. La información relacionada con la autolocalización del robot aun presenta errores de medición, sin embargo actualmente se trabaja en depurar este proceso.

Esta interfaz es el punto de partida para desarrollar tareas de navegación, exploración y mapeo, ya que forma parte de un proyecto a mayor escala en el cual, además de controlar el robot a través de esta interfaz se busca que se reconozcan objetos de interés, agregando algoritmos y técnicas de aprendizaje automático a este proyecto.

6. Referencias

- [1] Barrientos A., Del Cerro J., Gutiérrez P., San Martín R., Martínez A., Rossi C. "Vehículos aéreos no tripulados para uso civil. Tecnología y aplicaciones", Grupo de Robótica y Cibernética, Universidad Politécnica de Madrid, 2007.
- [2] Hoffmann G. M., Waslander S. L., Tomlin C. J. Quadrotor Helicopter Trajectory Tracking Control. AIAA Guidance, Navigation and Control Conference and Exhibit 18 – 21, Honolulu, Hawaii. Agosto 2008.
- [3] Piskorski S., Brulez N., Eline P. AR.Drone Developer Guide, SDK 1.7. 17 de mayo de 2011.
- [4] <http://www.grc.nasa.gov/WWW/K-12/airplane/rotations.html>. 29 de marzo de 2013.
- [5] <http://www.gtk.org>. Consultado 10 de Diciembre de 2012.
- [6] <http://www.parrot.com/es/>. Consultado el 04 de abril de 2013.
- [7] <http://www.gtk.org>. Consultado 10 de Diciembre de 2012.
- [8] <http://www.muyinteresante.es/parrot-presenta-la-nueva-version-de-su-famoso-cuadricoptero>. 28 de noviembre de 2012.
- [9] Toledo, L. Acosta, M. Sigut, and J. Felipe. Stabilisation and altitude tracking of a four-rotor microhelicopter using the lifting operators. Control Theory Applications, IET, 3(4):452–464, 2009.