

## Control de altitud de un cuadricóptero utilizando lógica difusa

Javitt Higmar Nahitt Padilla Franco<sup>1</sup>, Jorge Arturo Pérez Venzor<sup>1</sup>, Iván Alberto Mendoza Encinas<sup>1</sup>,  
Denisse Daniela Segovia Hernandez<sup>1</sup>, David García Chaparro<sup>1</sup>

<sup>1</sup>Universidad Autónoma de Ciudad Juárez.

### Resumen

En el presente artículo se diseña el prototipo de controlador de altitud difuso para un cuadricóptero modelo V262 Cyclone. Que contiene un sistema de comunicación Bluetooth punto a punto; un circuito de sensado y adquisición de datos; un entorno de visualización para el circuito de sensado y adquisición; un circuito adaptador de mando RC; la simulación del algoritmo difuso en Matlab; y un programa de lógica difusa para mantener la altitud. Todo fue integrado al cuadricóptero y se fijó en una base para realizar las pruebas de funcionamiento. Estas consistieron en vuelos de 50 segundos donde se buscó que alcanzara la altitud establecida de 50 centímetros y la prueba se consideraba exitosa si lo lograba en el límite de tiempo. Dichas pruebas se realizaron dentro de un ambiente controlado y se crearon tablas para presentar dichos resultados y poder obtener conclusiones.

**Palabras clave:** Lógica difusa, cuadricóptero, Matlab.

### Introducción

Hoy en día los vehículos aéreos no tripulados han tomado notoriedad debido a los múltiples usos que pueden llegar a tener. Sin embargo, esto no hubiera sido posible sin los avances en los sistemas modernos de control y a la miniaturización de los componentes electrónicos. Aún son muchos los retos que hay vencer para lograr que estos vehículos puedan desplazarse de manera autónoma y es de gran importancia encontrar nuevos métodos para ello, haciendo sistemas funcionales que cumplan su objetivo de una manera eficaz y den una alternativa para los métodos de control actuales [1].

La mayoría de los sistemas de control de los cuadricópteros utilizan para su implementación modelos matemáticos que pueden llegar a ser muy complejos, debido a la naturaleza de los vehículos aéreos no tripulados (UAVs por sus siglas en inglés: Unmanned Aerial Vehicle). Aunque se han implementado ya diversas técnicas de *soft computing* para este propósito, el control de los cuadricópteros, en especial el de la altitud, sigue siendo un reto, por lo que aún existe mucho campo de investigación para el modelado y creación de sistemas flexibles, rápidos y robustos para el control de estos vehículos.

## Construcción del prototipo

La construcción del prototipo consta de:

Un sistema de comunicación Bluetooth punto a punto.

Un circuito de sensado y adquisición de datos.

Entorno de visualización para el circuito de sensado y adquisición.

Circuito Adaptador de Mando RC.

Simulación del algoritmo en Matlab™.

### Sistema de comunicación Bluetooth punto a punto.

En las comunicaciones punto a punto, también llamadas comunicaciones transparentes, es donde el sistema de comunicación se comporta como un cable invisible entre los 2 puntos. Una de las partes vitales del proyecto fue lograr una comunicación entre los módulos HC06 y HC-05 para el envío y recepción de los

datos. A pesar de que los módulos Bluetooth son físicamente idénticos, no poseen las mismas características y posibilidades. El módulo HC-05 puede funcionar como esclavo o maestro mientras que el HC-06 solo funciona como esclavo, esto es debido a que poseen diferente firmware.

Estos módulos se configuran mediante comando AT enviados a través de un terminal hacia el puerto serial del mismo, ya que la información sobre estos es escasa, muchos fabricantes colocan su propio firmware y esto provoca que los comandos no sean los mismos, pero en general los módulos pueden formar una conexión punto a punto siempre y cuando se tenga un de ellos configurado como esclavo y el otro como maestro. El maestro deberá tener el PIN y la dirección MAC del dispositivo esclavo, mientras que el esclavo solo deberá estar disponible para emparejarse con el dispositivo maestro.

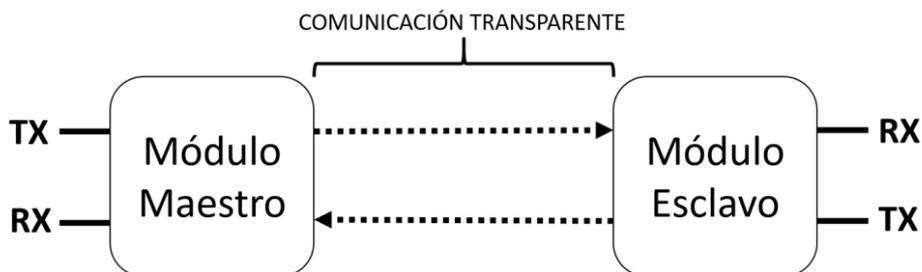


Figura. 1 Representación gráfica de comunicación serial transparente.

Circuito de sensado y adquisición

Este es el primer circuito que se desarrolló con el objetivo de conocer la altitud a la

que se encuentra el cuadricoptero, para luego mandarla a través de un transmisor inalámbrico, además el circuito debería ser lo suficientemente ligero y lo más compacto posible para que se integre exitosamente al cuadricóptero y este pudiese cargarlo con facilidad en vuelo.

Las dimensiones y peso ideales son los siguientes:

Peso: > 50 gramos.

Dimensiones: 7 cm x 7 cm.

Los componentes que integran el sistema son los siguientes:

Arduino Nano.

LM7805.

Conector de batería de 2 terminales.

Sensor ultrasónico HC-SR04.

Módulo Bluetooth HC-06.

Se llegó a la conclusión de que el circuito debería verse similar al boceto de la figura 2 respetando las restricciones de dimensiones y peso.

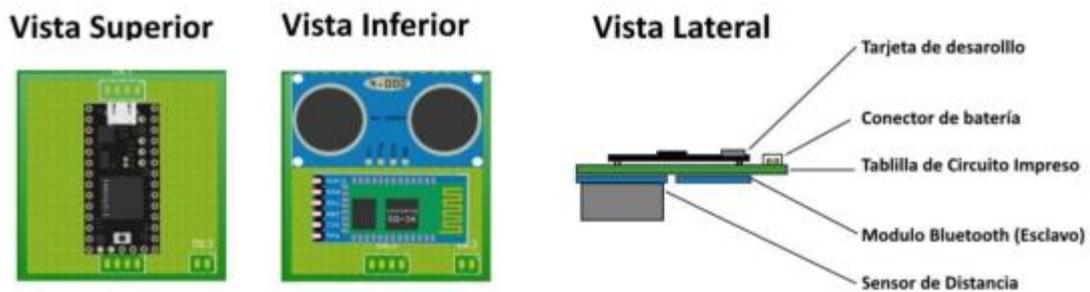


Figura 2. Boceto de circuito de sensado y adquisición.

No todos los componentes que integran este sistema funcionan bajo el mismo nivel de alimentación. La tarjeta de desarrollo puede funcionar con un voltaje de entrada de 5 a 12 volts mientras que los 2 módulos solo soportan un máximo de 5 volts, la batería de alimentación es de 7.4 volts por lo que el voltaje deberá ser regulado a 5 volts para evitar que los módulos funcionen y no se dañen. El integrado LM7805 nos permite regular un voltaje de entrada de 5 a 30 volts a una salida constante de 5 volts, por lo que lo es ideal para regular el voltaje que alimentara el sistema entero procedente de

la batería. Se realizaron una serie de pruebas básicas en el laboratorio para comprobar la operación del integrado previo a su integración al sistema general.

Para el envío de los datos de manera inalámbrica se utilizó el módulo HC-06 configurado de manera previa, que funciona en configuración esclavo, este se conectó de manera cruzada en el puerto serial del microcontrolador como se puede ver en la figura 3. Se modificó el programa del microcontrolador para que el dato de altitud enviado por el puerto serial sea en un byte, esto implica que el dato no podrá ser

visualizado directamente en un terminal serial [4].

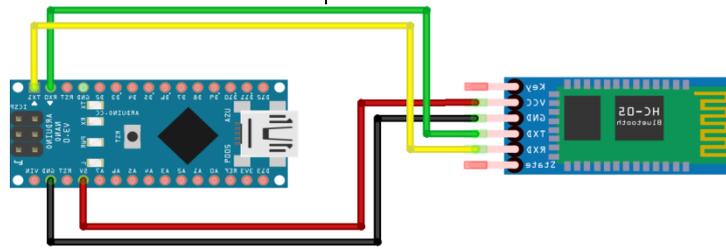


Figura 3. Conexión cruzada entre Arduino Nano R3 y módulo HC-06.

### Circuito Adaptador de Mando RC

Una de las características de la lógica difusa es que puede aplicarse a sistemas de control sin tener que tener en cuenta el modelo matemático del sistema que se desea controlar, lo que en este caso significa que no es necesario contemplar el modelo aerodinámico de cuadricóptero para realizar el controlador e implica que no se requiere de un cuadricóptero programable o controlable por software. El uso de un vehículo de juguete implica una mayor asequibilidad por su bajo costo y una mejor situación para probar que el controlador difuso puede estabilizar el sistema en altitud, pero también implica el problema de poder controlar este a través de software,

es por ello que fue necesario crear este sistema que consta de un circuito que nos permite controlar la velocidad de los motores por medio de una variable en software.

La conexión con el mando con la computadora es el primer obstáculo que se presentó, y se decidió empezar por estudiar el mando y ver cómo funciona. Como era ya evidente la potencia de los motores está controlada por un potenciómetro, este puede verse en la figura 4, tras un análisis sencillo se obtuvo el rango de voltaje que varía (0-3.3 volts). De momento se definió que para variar las direcciones sería necesario variar entre voltajes de 0 a 3.3 volts en la salida del potenciómetro de las direcciones que se quisiera controlar.

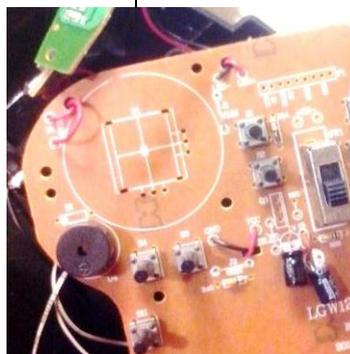


Figura 4. Potenciómetro dentro del mando RC.

### Módulo RF con comunicación SPI.

Se encontró que el emisor RF presente en el mando y con el cual se envían las órdenes al cuadricóptero funciona bajo el protocolo

SPI, el módulo puede verse en la figura 5. Esto abre la posibilidad de poder controlar el vehículo a través de este módulo RF que, de estar ligado con el otro en el

cuadricóptero, solo sería cuestión de enviar una cadena de bytes con los valores a los que se desea esté la potencia en cada una de las direcciones [5].

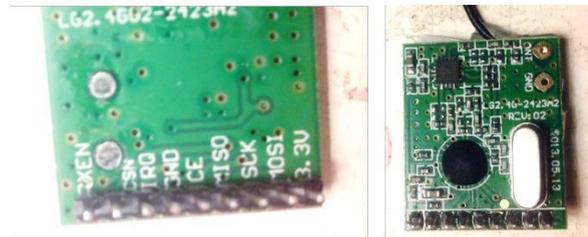


Figura 5. Módulo emisor RF del cuadricóptero.

DAC para variar voltaje.

El análisis para controlar la potencia utilizando un “potenciómetro virtual” controlado por software se lleva en el microcontrolador fue programado para

generar una señal PWM que en teoría variaría el voltaje, en el laboratorio se pudo observar que el voltaje variaba, pero los cambios de flanco eran lo suficientemente lentos para ser detectados por el microcontrolador presente en el mando RC.

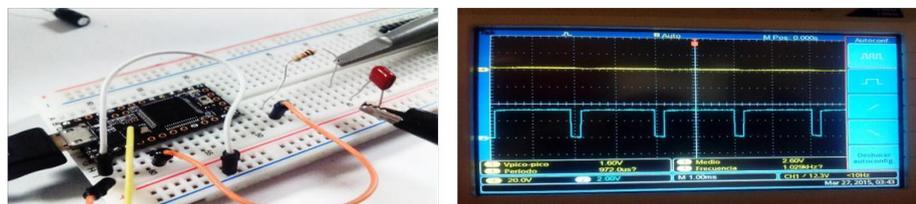


Figura 6. Pruebas al filtro RC.

Se crean filtros pasivos para filtrar la señal PWM generada por el microprocesador y se hacen pruebas con filtros pasivos RC de primer orden como el mostrado en el la figura 6. Tras varias pruebas se determinó que el tiempo de respuesta era lento.

El circuito se probó conectando la salida del DAC directamente sobre la salida del potenciómetro que regula la potencia de los motores, lo cual se puede ver en la figura 7. Se varió el voltaje a razón de 100 mV y se pudo observar que la variación de la potencia se realizaba perfectamente.



Figura 7. Salida del DAC conectada a la terminal de salida del potenciómetro en el radio control.

### Comunicación serial USB.

Para ver el comportamiento de la entrada y salida en este sistema los datos se envían a través del Serial0 que se encuentra direccionado al puerto USB en la tarjeta Teensy 3.1, lo que permite observar en la pantalla del terminal, los datos recogidos en

el puerto Serial1 (Bluetooth) y la salida que se está escribiendo en el DAC [6].

Se utilizó una tablilla perforada de 5 cm x 5 cm para montar ahí la tarjeta Teensy 3.1, un indicador LED, el módulo Bluetooth y un cable con conector DB9 macho. Figura 8.

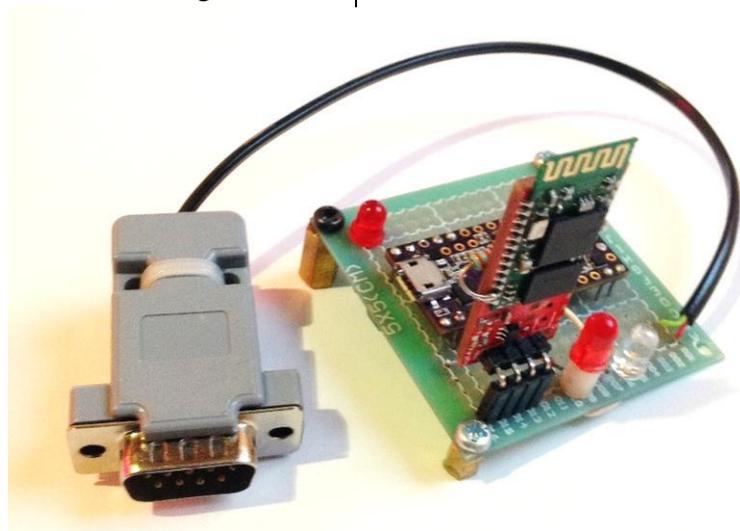


Figura 8. Circuito Adaptador de Mando RC.

Programa de lógica difusa para el control de altitud.

La creación del programa de lógica difusa que toma las decisiones sobre la altura del cuadricóptero fue sin duda una parte de vital importancia para el proyecto [7]. Se decidió crear en el lenguaje de programación C para que fuera fácilmente adaptable a diferentes plataformas. Además, se utilizó la librería eFLL que

funciona para controladores difusos, la cual es de software libre. Para la creación del algoritmo previo a la codificación se tomaron en cuenta las siguientes características:

Número de entradas .- Para el número de entradas se definió que el único dato relevante es la altura que tenía el cuadricóptero en un momento dado con

respecto a la referencia, a esta entrada se le llamo "Altitud".

Referencia .- La referencia la definimos como la altitud que marca el usuario a la cual debe mantenerse el cuadricóptero en el eje Z.

Número de salidas .- El número de salidas es la acción que tomará el algoritmo después de tomar la decisión de acuerdo a las reglas. Para el proyecto la única salida relevante es la velocidad de los motores, de la cual depende si el cuadricóptero sube o baja la velocidad para igualmente subir o bajar la altitud, a esta salida se le llamo

"vMotores", lo cual representa la velocidad de los motores.

Tipo de conjunto difuso .- El tipo de conjunto seleccionado es el trapezoidal ya que es uno de los más sencillos y adaptables en la teoría de la lógica difusa, y es ampliamente utilizado en microcontroladores.

Número de variables lingüísticas .- Para elegir el número de variables lingüísticas adecuadas fueron necesarias varias pruebas para comprobar el comportamiento del cuadricóptero con menos o con más variables. Esto se debe a que con pocas el efecto de rebote sería muy grande y no podría alcanzar la altura adecuada, y por el contrario muchas variables podrían ser innecesarias. Finalmente se decidió que el número de variables con las que el funcionamiento del sistema es correcto se estableció en 29 del conjunto de entrada y 29 del conjunto de salida.

Voltaje de operación de los motores .- El voltaje de operación de los motores fue

establecido entre un 0% y un 80% de la velocidad total ya que por los requerimientos de la batería es el máximo que se puede alcanzar.

Rango de altitud .- El rango de altitud en la que opera el controlador difuso para tomar sus decisiones es de 1 metro. Si el rango de altitud supera la distancia, o está por debajo de esta, el algoritmo tomará la decisión de la última regla pertinente, que es subir o bajar la velocidad de los motores rápidamente, hasta llegar a una altura que entre en el rango de la referencia.

Funciones de pertenencia de las variables lingüísticas de entrada .- Debido a que el número de variables lingüísticas del conjunto difuso de entrada "Altitud" se marcaron en 29, y el rango de altitud es un metro, se hizo una división del rango de altitud para asignarle rangos a los conjuntos difusos y así obtener los trapecios de cada conjunto. Las funciones de pertenencia se verían como en la figura 9.

Funciones de pertenencia de las variables lingüísticas de salida .- Para el conjunto de variables de salida, se dividió la potencia total de los motores entre las 29 posibles salidas de acuerdo a las reglas.

Número de reglas. El número de reglas es proporcional al número de variables lingüísticas de entrada ya que cada entrada contendrá una salida, por lo tanto, el número de reglas es un total de 29.

Tipo de sistema difuso .- El sistema difuso es de tipo Mamdani, el cual utiliza el tipo de reglas si-entonces.

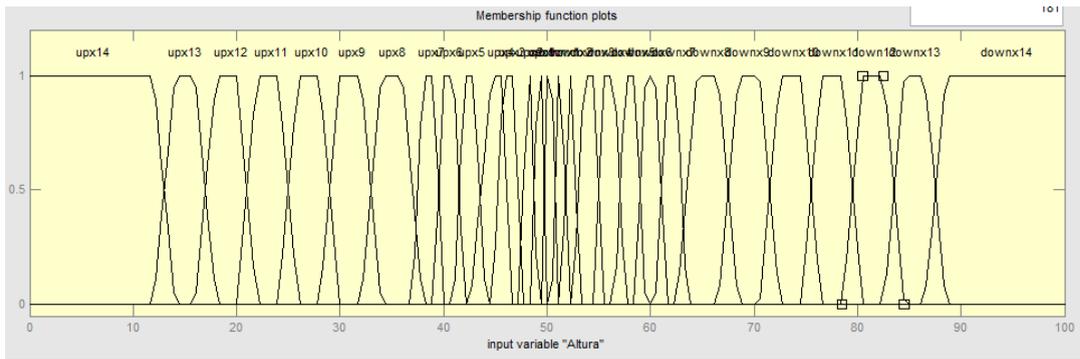


Figura 9. Gráfica que representa las funciones de pertenencia de la altura.

El circuito de sensado y adquisición toma la altura actual con respecto a la referencia de entrada.

El algoritmo decide cual es el grado de pertenencia a los conjuntos difusos definidos.

Según la pertenencia de la entrada decide que acción debe tomar de acuerdo a la base de reglas.

El microprocesador le envía la velocidad de los motores al control y este aplica la acción en el cuadricóptero.

## Conclusiones

Durante el desarrollo del proyecto, una vez que se construyó el prototipo y el entorno de pruebas, se realizaron diferentes versiones del controlador, observando y obteniendo datos para su análisis y así determinar de qué manera se podría mejorar hasta alcanzar el objetivo propuesto. Las características bajo las que se realizaron las pruebas fueron las siguientes:

Periodos de tiempo de vuelo de 50 segundos.

El número de pruebas por versión de controlador fue de 15.

En el ambiente de pruebas la altura definida para mantener es de 50 cm.

El rango de precisión se definió entre los -10cm a +10cm a partir de la altura que se indicó, por lo que si el cuadricóptero logró mantener por ese periodo de tiempo la altitud se consideró un caso exitoso.

El tiempo de vuelo se midió con un cronómetro.

El sensor fue el encargado de tomar las medidas de altitud, lo que indicó si pudo mantener la altitud el periodo de tiempo establecido.

Las pruebas se hicieron dentro de un ambiente controlado donde se trató de reducir al máximo el ruido y las perturbaciones que pudieran alterar los datos obtenidos, como las corrientes de aire.

El cuadricóptero se insertó en una base de madera de 1 metro de altura para controlar la desestabilización que se generó a causa del peso agregado en el eje x.

Al final de las pruebas, si el 80% de estas fueron exitosas, el proyecto se consideraría exitoso.

Un video de cómo funciona esta versión del controlador siendo sometida a perturbaciones puede observarse en la siguiente liga:

<https://www.youtube.com/watch?v=N-9J5IDEJAE>

Al observar las versiones de los controladores se puede ver que, al inicio, cuando en la primera versión del controlador existían pocas reglas, el sistema era bastante inestable al grado que no mantenía altitud alguna y por el contrario, variaba rápidamente entre 0 y 100 centímetros golpeándose contra los extremos de la base, por lo que debió ser suspendida la prueba del controlador para garantizar la integridad del sistema construido y del mismo cuadricóptero.

En la versión número dos del controlador donde aumentó en 4 el número de reglas, se puede empezar a observar que el sistema en algunas de las pruebas funcionó medianamente de la forma deseada, ya que, aunque no mantuvo una altitud fija, en 4 de las pruebas logró mantenerse oscilando entre los valores considerados como exitosos (entre 40 y 60 centímetros). Por otra parte, las oscilaciones que tuvo el sistema en las pruebas no exitosas ya no fueron tan grandes como en el controlador

anterior, por lo que se continuaron sin ningún temor a comprometer el sistema. Esta versión fue determinante en comprobar que al aumentar el número de reglas el sistema tenía un mejor comportamiento.

En la siguiente versión del controlador se puede observar un aumento significativo en la tasa de éxitos ya que prácticamente todas las pruebas fueron exitosas exceptuando dos, por lo que se volvió a comprobar que el aumento de reglas le añade al sistema rapidez para estabilizarse, la oscilación entre los extremos es menor, pero sin embargo aún se observa movimiento después de que el sistema se encuentra en el rango considerado exitoso.

La versión 4 con 21 reglas, es la primera en presentar el 100% de las pruebas como exitosas, el sistema se estabiliza con rapidez en comparación de las versiones anteriores y la oscilación una vez estable el sistema es prácticamente nula. Además de esto es la primera versión que permite someter al sistema a perturbaciones externas y se pudo comprobar que este reacciona bien y compensa las perturbaciones con fluidez, volviendo a la posición deseada sin problemas.

La versión número 5 con 29 reglas funcionó de una forma muy similar a su antecesora, mantuvo una altitud estable prácticamente sin oscilación y el tiempo de estabilización fue adecuado y su funcionamiento óptimo.

Gracias a estas pruebas se puede concluir que entre mayor número de reglas el funcionamiento del sistema construido es mejor, sin llegar a escribir más reglas de las necesarias. Además de esto, se demuestra que de la manera que fue planteada inicialmente la solución, es posible

controlar por software la altitud de un cuadricóptero considerado como un juguete y que no tiene función alguna para esto, con una técnica de control con lógica difusa. También se puede deducir que si es posible manejar un sistema con estas características sería posible aplicar esta técnica para el control de otros sistemas igualmente complejos.

Después del periodo de construcción, simulación y prueba de los elementos

utilizados, podemos concluir que el proyecto fue un éxito, ya que se pudo demostrar que fue posible controlar de una manera satisfactoria la altitud de un cuadricóptero modelo Cyclone V262, que no contaba con función alguna para mantener la altitud, por medio de un controlador difuso para la toma de decisiones.

## Referencias

[1] Jamshidi, M., Vadiiee, N. and Ross, T. “*Fuzzy logic and control: software and hardware applications*”, Englewood Cliffs, N.J.: Prentice Hall, 1993, p.339.

[2] S. Sheikhpour, and S. B. Shouraki. “A model-based fuzzy controller using the parallel distributed compensation method for quadrotor attitude stabilization”, in *21st Iranian*

*Conference on Electrical Engineering (ICEE)*, Mashhad, 2013.

[3] A. Nemes, “Synopsis Of Soft Computing Techniques Used In Quadrotor Modelling And Control”, *Interdisciplinary Description of Complex Systems*, Vol. 13, 2015,

[4] M. Fatan, B. Sefidgari, and A. Barenji. “An adaptive neuro PID for controlling the

altitude of quadcopter robot “, *18th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Miedzydroje, 2013.

[5] Austing, T., “*Unmanned Aircraft Systems: UAVS Design, Development and Deployment*”, Chinchester: WILEY, 2010.

[6] Blasco, X., García-Nieto, S., Reynoso-Meza, G. “*Control autónomo del seguimiento de*

*trayectorias de un vehículo cuatrirrotor*”. Comité Español de Automática, 2012.

[7] Cuesta, F., y Ollero A. “*Intelligent Mobile Robot Navigation*”, Berlin: Springer, 2005.