

Control Neuronal en Línea para Regulación y Seguimiento de Trayectorias de Posición para un Quadrotor

Hugo Yañez-Badillo^a, Ruben Tapia-Olvera^{b, *}, Omar Aguilar-Mejía^a, Francisco Beltran-Carbajal^c

^a Departamento de posgrado, Universidad Politécnica de Tulancingo, Ingenierías #100, Col. Huapalcalco, 43629, Tulancingo, Hgo., México.

^b Departamento de Ingeniería Eléctrica, Universidad Nacional Autónoma de México, Av. Universidad 3000, Cd. Universitaria, Delegación Coyoacán, 04510 Ciudad de México, México.

^c Departamento de Energía, Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Av. San Pablo 180, Col. Reynosa Tamaulipas, Delegación Azcapotzalco, 02200 Ciudad de México, México.

Resumen

Los sistemas de control automático día a día se han convertido en elementos importantes en la vida cotidiana, en tal sentido, se deben proponer nuevas y mejores formas de incorporar modelos matemáticos y algoritmos de control adaptativos para superar la gran cantidad de cambios técnicos y físicos a los que se enfrentan para su utilización. En este artículo se realiza el control de posición y seguimiento de trayectorias para un Quadrotor. Debido a la naturaleza no lineal de este sistema subactuado, se propone el empleo de un controlador adaptativo basado en redes neuronales B-spline que permita determinar las señales de control mediante un entrenamiento dividido en dos etapas: a) uno inicial fuera de línea y; b) uno continuo en línea. Esta forma de aprendizaje permite al Quadrotor extender un desempeño satisfactorio ante diferentes condiciones operativas y seguimiento de los valores de referencia. Los resultados de simulación verifican la aplicabilidad del controlador propuesto y el impacto que se tiene en el desempeño del sistema minimizando la necesidad de contar con un modelo matemático no lineal detallado, así como el conocimiento exacto de los valores de los parámetros del Quadrotor.

Palabras Clave:

Control Neuronal, Control Libre de Modelo, Aprendizaje Automático, Sistemas Subactuados.

1. Introducción

El helicóptero con cuatro rotores, conocido en la literatura como Quadrotor, es un sistema subactuado con seis grados de libertad (GDL) y cuatro actuadores independientes (entradas de control). Además, este tipo de sistemas pueden estar sometidos a incertidumbres endógenas y exógenas durante su operación. Se enfrentan a un medio altamente cambiante, como consecuencia de la fricción con el viento, velocidades de viento variables y cambio en la humedad del entorno. Se observan características dinámicas complejas, un comportamiento no lineal entre las diversas variables de interés e incertidumbre paramétrica. En tal sentido, utilizar técnicas de control adaptativas y robustas representan una alternativa de solución al problema de regulación y seguimiento de trayectorias en lazo cerrado para un Quadrotor, para contar con un sistema eficiente de navegación y estabilización. La técnica de control apropiada depende principalmente de los objetivos de aplicación que se le designen al Quadrotor. Se han propuesto diversos controladores lineales y no lineales, buscando como un objetivo principal conseguir un alto nivel de estabilidad (Chee,

Zhong, 2013; Boudjedir, Bouhali, Rizoug, 2014; Meng, Ming, Po-Leen, 2012; Fatan, Sefidgari, Barenji, 2013; Salih, Moghavvemi, Mohamed, Gaied, 2010).

Entre las principales técnicas de control resaltan controladores proporcional derivativo (PD), proporcional integral derivativo (PID), backstepping, H_∞ , estimadores de estado basados en filtros de Kalman, regulador lineal cuadrático (LQR por sus siglas en inglés), redes neuronales, algoritmos difusos (Chee *et al.*, 2013; Boudjedir *et al.*, 2014; Johnson, Leang, 2013; Erginer, Altug, 2007; Argentim, Rezende, Santos, Aguiar, 2013). Existen controladores implementados para diferentes modelos del Quadrotor, cada uno definido con sus respectivas particularidades dependiendo de los objetivos. Esto propicia que algún controlador presente ventajas con las particularidades del caso. No obstante, éstos pueden exhibir alguna degradación en su desempeño, para condiciones de operación distintas para las cuales fue diseñado. Más aún, información exacta de los parámetros constantes o variables del sistema podría no estar disponible para la implementación de algún controlador. El modelo no lineal es de gran utilidad, debido a que proporciona mayor información del comportamiento físico del sistema (Boudjedir *et al.*, 2014). Sin embargo, el modelo linealizado se emplea ampliamente en la literatura, debido a que en general presenta buenos resultados de regulación alrededor de un punto de equilibrio. Este modelo se utiliza con condiciones definidas de vuelo (hover) siendo estables

* Autor en correspondencia.

Correos electrónicos: dophthugo.yanez@upt.edu.mx (Hugo Yañez-Badillo), rtapia@fi-b.unam.mx (Ruben Tapia-Olvera), omar.aguilar@upt.edu.mx (Omar Aguilar-Mejía), fbeltran@azc.uam.mx (Francisco Beltran-Carbajal).

y efectivos sólo en rangos de operación que corresponden a ángulos razonablemente pequeños. Dentro de las ventajas de estos diseños basados en controladores lineales se puede mencionar: a) presentan desempeño aceptable; b) tienen una estructura simple y; c) pueden ser sintonizados con un modelo lineal aproximado. Para implementaciones en tiempo real, es importante considerar una ley de control robusta que considere un desempeño aceptable para diversas condiciones operativas, un controlador basado en el modelo linealizado reduce su aplicabilidad (Meng *et al.*, 2012).

El controlador PID funciona calculando los errores o diferencia entre la salida medida y el valor de referencia asignado, definiendo las entradas de control para ajustar la velocidad de los motores. Aunque este método de control ha sido propuesto inicialmente para sistemas lineales, también se ha utilizado para el control de sistemas no lineales (Fatan *et al.*, 2013). En (Meng *et al.*, 2012; Salih *et al.*, 2010; Johnson *et al.*, 2013; Erginer *et al.*, 2007; Argentim *et al.*, 2013; Bouabdallah, Noth, Siegwart, 2004) los estudios se enfocan en determinar un control óptimo a través de algoritmos de control convencionales PD y PID, cada trabajo tiene su propia metodología de sintonización. En (Johnson *et al.*, 2013) se presenta un algoritmo PD para mejorar la respuesta transitoria principalmente ante diferentes perturbaciones, la actitud del Quadrotor se analiza mediante el uso de cuaterniones para evitar singularidades Gimbal lock.

En (Salih *et al.*, 2010) se utiliza un algoritmo PID aplicado a un modelo matemático derivado a partir de la aproximación de LaGrange, en este estudio no se consideran los efectos giroscópicos, obteniendo un modelo relativamente simple, esto permite contar con un primer acercamiento en el modelado, simulación e implementación de un Quadrotor de estas características. En (Dikmen, Arisoy, Temeltas, 2009) se considera una versión del modelo incluyendo efectos giroscópicos, realizando un análisis dinámico con controladores PD, backstepping y control inverso, el último tiene como objetivo el control del sistema a través de dinámica inversa.

El algoritmo LQR es una forma de control óptimo lineal que tiene como objetivo reducir la magnitud de la entrada de control sin afectar su desempeño. En (Meng *et al.*, 2012) se emplea este regulador para obtener los parámetros de ajuste que minimizan las alteraciones no deseadas (altitud), y al mismo tiempo limitar la energía de la acción de control usando un algoritmo que minimice una función de costo o índice de desempeño con factores de ponderación. En (Castillo, García, Lozano, Albertos, 2007) se compara el desempeño del controlador LQR contra un controlador no lineal basado en el análisis de Lyapunov, el modelo se desarrolla a partir de la formulación LaGrange-Euler. En (Argentim *et al.*, 2013) se encuentran algoritmos PID y LQR, primero se presentan las constantes del controlador PID derivadas de la teoría LQR, posteriormente se sintoniza el algoritmo PID con un procedimiento utilizando como función la integral del tiempo por el error absoluto (ITAE).

De igual manera los controles convencionales se utilizan como complementos de controladores más complejos y robustos, en (Mian A., Mian I., Daobo, 2008; Mian A., Daobo, 2008) se fundamenta el diseño de controladores no lineales backstepping. En (Bouabdallah, Siegwart, 2005) se implementa un controlador backstepping especificando los parámetros empleados en la etapa de simulación, donde se proponen condiciones iniciales adversas para el vuelo estable del Quadrotor, los resultados muestran un desempeño adecuado ante perturbaciones. En (Benallegue, Mokhtari, Fridman, 2008) el algoritmo de modos deslizantes se utiliza como observador de estado para estabilizar el sistema,

adicionalmente se diseña un observador-estimador que pretende desprestigiar las perturbaciones inherentes del sistema con el menor número de sensores, esto se logra mediante una estrategia de linealización en la retroalimentación.

Es importante citar que en la literatura donde se desarrollan algoritmos complejos de control, su desempeño se compara bajo diversas circunstancias con controladores convencionales analizando su capacidad de controlabilidad. En (En-Hui, Jing-Jing, Ji-Liang, 2014; Jing-Jing, En-Hui, 2014) se implementa una estrategia de modos deslizantes y posteriormente se discuten beneficios del controlador frente a las características de operación que entrega un controlador PID. En (Luque-Vega, Castillo-Toledo, Loukianov, 2012) se contempla el uso de un controlador por modos deslizantes, proporcionando detalles de las simulaciones que validan su desempeño considerando algunos objetivos como la estimación de variables aerodinámicas y el seguimiento de trayectorias. Por su parte en (Chee *et al.*, 2013; Fatan *et al.*, 2013; Dierks, Jagannathan, 2010; Nicol, Macnab, Ramirez-Serrano, 2008; Emran, Yesildirek, 2014) se optimizan controladores convencionales con algoritmos adaptativos, en (Boudjedir *et al.*, 2014) se presentan consideraciones del entrenamiento de la red neural, tanto en línea como fuera de línea, definiendo una sola capa oculta para estimar parámetros desconocidos en el modelo. En (Fatan *et al.*, 2013) un neuro-controlador PID realiza el cálculo de los coeficientes óptimos para un adecuado desempeño, se enfatiza una correcta caracterización de los valores primarios por ejemplo la regla de aprendizaje, en este tipo de algoritmos se busca disminuir la demanda de operaciones en la etapa de procesamiento. En (Argentim *et al.*, 2013) se lleva a cabo un control adaptativo con una estrategia complementada utilizando modos deslizantes.

Las diferentes representaciones del modelo dinámico para esta clase de sistemas, convergen a cierto nivel de complejidad dependiendo de los factores que se reconocen y afectan al sistema. Algunos trabajos enfocan sus esfuerzos en estudios más detallados, como en (Estelles, Tomas-Rodriguez, 2015), donde se proponen controladores PVA (posición-velocidad-aceleración) basados en observaciones de las velocidades del Quadrotor y se evalúan en un modelo denominado flexible. En el modelo se analizan las vibraciones del cuerpo asociadas al movimiento de los rotores, y otro modelo simplificado que es equivalente al descrito en gran parte de los trabajos relacionados al control de este vehículo. Por otra parte, en (Pipatpaibul, Ouyang, 2013) se comparan dos controladores basados en el aprendizaje mediante iteraciones (ILC), para verificar el desempeño de los esquemas se realizan diversas simulaciones para el seguimiento de trayectorias planificadas variables. En (Bouabdallah, Siegwart, 2007) se hace un análisis de contribuciones previas (Bouabdallah *et al.*, 2004; Bouabdallah *et al.*, 2005; Bauer, Ritzinger, Soumelidis, Bokor, 2008) concluyendo su desarrollo con un mayor número de variables que describen la dinámica del sistema, además, describen algunas de las consideraciones que se deben tomar en cuenta en la etapa de implementación del control al sistema Quadrotor.

Finalmente, se puede apreciar que el uso de estimadores y observadores de estado es una opción importante para este tipo de tareas de control. En (Bresciani, 2008) se tiene el diseño con un filtro de Kalman lineal, e incluye una versión no lineal del filtro del Kalman extendido, diseñado mediante una estrategia lineal cuadrática para mejorar el desempeño del primero y comparar los resultados. Algunas plataformas experimentales como el Starmac (Hoffmann *et al.*, 2008) y el OS4 (Bouabdallah, Siegwart, 2005) han permitido evaluar el seguimiento de trayectorias mediante algunos de los esquemas de control citados, con resultados

satisfactorios tanto en ambientes interiores como exteriores. Las diversas técnicas de control tienen un desempeño satisfactorio pero en general quedan limitadas a algunas condiciones operativas y condiciones de diseño, en tal sentido, es necesario mejorar el desempeño del Quadrotor contando con un controlador adaptativo que permita hacer frente a variaciones paramétricas, perturbaciones del sistema y operaciones desconocidas, con características importantes para una implementación en tiempo real.

De forma particular un inconveniente que tienen algunos esquemas de control robustos y adaptativos reportados en la literatura es que al realizar su diseño se necesita conocer su modelo, la magnitud de las perturbaciones internas o externas, ocasionando que el procedimiento de análisis y diseño sea muy complejo y a la vez difícil (Boudjedir *et al.*, 2014; Dydek, Annaswamy, Lavretsky, 2013).

En este artículo se realiza la propuesta de un controlador basado en redes neuronales B-spline (RNBS) que regula la posición y trayectoria de un Quadrotor. A diferencia de los controladores existentes, con la propuesta se tiene la posibilidad de que el sistema mantenga un desempeño satisfactorio para diversas condiciones operativas, se realice una sintonización de los controladores de manera simultánea sin la necesidad de contar con un modelo matemático exacto y considerar las no linealidades inherentes para este tipo de sistemas. Finalmente, a partir de la propuesta es mínima la cantidad de operaciones que se requieren para el cálculo de las señales de control, esto permite observar que es una estrategia atractiva para su implementación en tiempo real. Los resultados se comparan con los obtenidos con un esquema de control no lineal basado en modos deslizantes.

2. Modelado del Quadrotor

En este trabajo se emplea la formulación Newton-Euler para determinar el comportamiento dinámico. En general, se considera al Quadrotor como un cuerpo rígido en un espacio tridimensional que cuenta con 6 GDL (véase la Figura 1). Las ecuaciones de fuerza neta, \mathbf{F}_B , y momentos actuantes, $\boldsymbol{\tau}_B$ sobre el Quadrotor en el marco de referencia móvil son,

$$\sum \mathbf{F}_B = m \dot{\mathbf{v}}_B + \boldsymbol{\Omega}_B \times m \mathbf{v}_B \quad (1)$$

$$\sum \boldsymbol{\tau}_B = \mathbf{J} \dot{\boldsymbol{\Omega}}_B + \boldsymbol{\Omega}_B \times \mathbf{J} \boldsymbol{\Omega}_B \quad (2)$$

donde \mathbf{J} es el tensor de inercia; m es la masa del Quadrotor; \mathbf{v}_B y $\boldsymbol{\Omega}_B$ son respectivamente los vectores de velocidad traslacional y angular del Quadrotor, en el marco de referencia móvil O_{ABC} . Aquí, el subíndice B se utiliza para denotar el marco de referencia móvil.

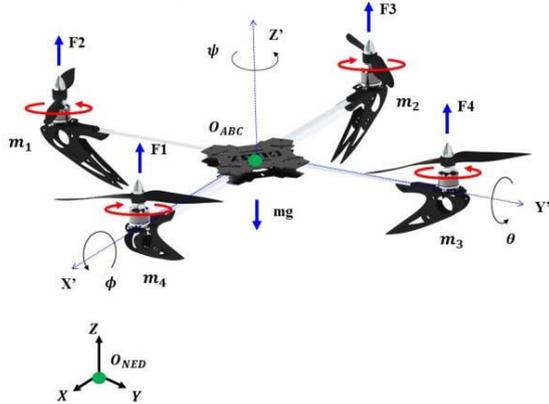


Figura 1: Diagrama esquemático de un Quadrotor para la obtención de su modelo dinámico.

El movimiento traslacional respecto al marco de referencia inercial o fijo y el movimiento rotacional en el marco de referencia móvil se pueden también describir como,

$$\begin{bmatrix} m \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_e \\ \dot{\boldsymbol{\Omega}}_B \end{bmatrix} = \begin{bmatrix} \mathbf{F}_e \\ \boldsymbol{\tau}_B \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \boldsymbol{\Omega}_B \times \mathbf{J} \boldsymbol{\Omega}_B \end{bmatrix} \quad (3)$$

Entonces, el movimiento de traslación del Quadrotor está dado por (En-Hui Z *et al.*, 2014, Jing-Jing X *et al.*, 2014)

$$\begin{aligned} \ddot{x} &= \frac{1}{m} (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) u_1 - \frac{k_1}{m} \dot{x} \\ \ddot{y} &= \frac{1}{m} (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) u_1 - \frac{k_2}{m} \dot{y} \\ \ddot{z} &= \frac{1}{m} (\cos \theta \cos \phi) u_1 - g - \frac{k_3}{m} \dot{z} \end{aligned} \quad (4)$$

donde θ , ψ y ϕ son los ángulos de Euler, los cuales describen los movimientos angulares de *roll*, *yaw* y *pitch*, respectivamente (Castillo *et al.*, 2007). Los parámetros k_1 , k_2 y k_3 son los coeficientes de amortiguamiento viscoso en las direcciones x , y , z , respectivamente.

El movimiento de rotación con respecto al marco de referencia móvil se describe como sigue (En-Hui Z *et al.*, 2014, Jing-Jing X *et al.*, 2014),

$$\begin{aligned} \dot{p} &= \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{I_r}{I_{xx}} q \Omega_n + \frac{1}{I_{xx}} u_2 \\ \dot{q} &= \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{I_r}{I_{yy}} p \Omega_n + \frac{1}{I_{yy}} u_3 \\ \dot{r} &= \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{1}{I_{zz}} u_4 \end{aligned} \quad (5)$$

donde ahora p , q y r representan las velocidades angulares en ese sistema de referencia. Aquí, u_1 , es la entrada de control de fuerza, u_2 , u_3 y u_4 los pares de control. Ω_n es la suma escalar de las velocidades angulares de los rotores,

$$\Omega_n = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (6)$$

Nótese que en el modelo (5) se han incluido también los efectos giroscópicos.

Puesto que se considera una estructura simétrica, el centro de gravedad del Quadrotor es coincidente con el origen del marco de referencia móvil y el desplazamiento angular es cercano a la condición inicial de vuelo. Entonces, la relación entre los desplazamientos angulares descritos en el marco de referencia fijo con aquellos definidos en el marco de referencia móvil está dada por (En-Hui Z *et al.*, 2014, Jing-Jing X *et al.*, 2014),

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (7)$$

En algunas contribuciones de control robusto, el modelo matemático (4) se simplifica considerando en las ecuaciones (5) y (7) que para desplazamientos angulares pequeños se tiene que $\dot{\phi} \approx p$, $\dot{\theta} \approx q$, $\dot{\psi} \approx r$ (En-Hui Z *et al.*, 2014, Jing-Jing X *et al.*, 2014). Así, si se asume que el Quadrotor se encuentra en una condición

suspendido en el aire, las ecuaciones de movimiento se aproximan como (En-Hui Z *et al.*, 2014, Jing-Jing X *et al.*, 2014),

$$\begin{aligned} \ddot{\phi} &= \frac{I_{yy} - I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} - \frac{I_r}{I_{xx}} \Omega_n \dot{\theta} + \frac{1}{I_{xx}} u_2 \\ \ddot{\theta} &= \frac{I_{zz} - I_{xx}}{I_{yy}} \dot{\phi} \dot{\psi} + \frac{I_r}{I_{yy}} \Omega_n \dot{\phi} + \frac{1}{I_{yy}} u_3 \\ \ddot{\psi} &= \frac{I_{xx} - I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta} + \frac{1}{I_{zz}} u_4 \end{aligned} \quad (8)$$

No obstante, la implementación del esquema de control propuesto en este artículo se realiza en la dinámica no lineal descrita por las ecuaciones (4), (5). Además, el desempeño del enfoque de control presentado fue evaluado en el modelo matemático no lineal obtenido por el método de Euler-Lagrange, para el sistema de referencia mostrado en la Figura 1, conduciendo a los mismos resultados. Para el lector interesado, en (Castillo *et al.*, 2007) se describe en detalle la obtención del modelo matemático usando el Lagrangiano del sistema.

Las entradas de control de fuerza, u_1 , y de torque, u_2, u_3 y u_4 , se consiguen a través de las velocidades angulares de los rotores como sigue,

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (9)$$

donde u_1 considera todas las fuerzas que ayudan a superar la fuerza gravitacional; u_2 produce un cambio del ángulo roll; u_3 produce un cambio en el ángulo pitch y; u_4 relaciona el movimiento en yaw; b es el coeficiente de empuje; l es la distancia del rotor al centro de masa del Quadrotor; d es el coeficiente de arrastre y Ω_n es la velocidad angular respectiva de las aspas de cada motor.

Los coeficientes se pueden estimar mediante la teoría del elemento pala (Bresciani, 2008; Mohamed, 2010). La identificación depende de la geometría del rotor, perfil, área del disco, radio y densidad del aire. La simulación del sistema se desarrolla en el entorno de Matlab® donde se define el modelo del Quadrotor, la fuerza aplicada por cada motor y la ley de control que se define en la sección 3.

3. Esquema de control

Uno de los objetivos que se pueden conseguir mediante el control automático en un Quadrotor es proporcionarle autonomía de movimiento, lo que en muchas ocasiones implica definir una trayectoria a seguir. El modelo descrito en la sección 2 permite implementar tres controladores independientes para estabilizar los ángulos fundamentales de movimiento: ϕ, θ , y ψ . Adicionalmente al controlador de los ángulos de posición, se utiliza un controlador para el desplazamiento en el eje z. Si se establece que las entradas de referencia al sistema son para los tres ejes x_d, y_d y z_d , Figura 2, entonces es necesario determinar los valores de referencia para los ángulos a los cuales se debe mover el Quadrotor, ϕ_d, θ_d y ψ_d y se pueda seguir la trayectoria deseada.

Una estructura de control típica para tareas de seguimiento de trayectorias es mediante controladores lineales PD. Sin embargo, estos quedan limitados al diseño original que se realice y, por lo

tanto, a ciertas condiciones operativas. En este trabajo, se combinan los siguientes controladores PD con redes neuronales para mejorar el desempeño del seguimiento de la planificación del movimiento, establecido para la operación del Quadrotor.

$$\begin{aligned} u_1 &= \left[g + K_{pz}(z_d - z) + K_{Dz}(\dot{z}_d - \dot{z}) \right] \frac{m}{\cos \phi \cos \theta} \\ u_2 &= \left[K_{P\phi}(\phi_d - \phi) + K_{D\phi}(\dot{\phi}_d - \dot{\phi}) \right] I_{xx} \\ u_3 &= \left[K_{P\theta}(\theta_d - \theta) + K_{D\theta}(\dot{\theta}_d - \dot{\theta}) \right] I_{yy} \\ u_4 &= \left[K_{P\psi}(\psi_d - \psi) + K_{D\psi}(\dot{\psi}_d - \dot{\psi}) \right] I_{zz} \end{aligned} \quad (10)$$

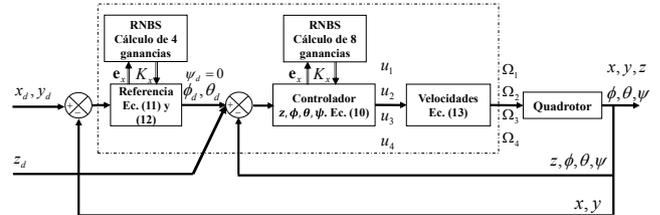


Figura 2: Diagrama de bloques del control de trayectoria de un Quadrotor.

Así, las trayectorias de referencia para los desplazamientos angulares ϕ_d y θ_d se definen con base en las trayectorias planificadas para los movimientos de traslación x_d y y_d como sigue

$$\begin{aligned} \theta_d &= \sin^{-1} \left(\frac{m}{u_1} u_x - \frac{\sin \phi \sin \psi}{\cos \phi \cos \psi} \right) \\ \phi_d &= \sin^{-1} (u_x \sin \psi - u_y \cos \psi) \end{aligned} \quad (11)$$

con

$$\begin{aligned} u_x &= K_{Px}(x_d - x) + K_{Dx}(\dot{x}_d - \dot{x}) \\ u_y &= K_{Py}(y_d - y) + K_{Dy}(\dot{y}_d - \dot{y}) \end{aligned} \quad (12)$$

Nótese que las fuerzas debidas al amortiguamiento viscoso se consideran como términos de perturbación desconocidos.

Para determinar los valores correctos de las velocidades angulares que requiere el Quadrotor para alcanzar un vuelo estable, se requiere la inversa de (9),

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & -1 \\ 4b & 0 & -2bl & -4d \\ 1 & -1 & 0 & 1 \\ 4b & 0 & 2bl & -4d \\ 1 & 1 & 0 & 1 \\ 4b & 2bl & 0 & 4d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (13)$$

Así, los controladores (10) se sustituyen en la ecuación (13) para determinar la velocidad angular para cada motor. Se puede considerar que la relación de cambio en las velocidades angulares $\dot{\phi}_d, \dot{\theta}_d, \dot{\psi}_d$ sean igual a cero. En el caso de ψ_d se considera igual a cero para que no exista un movimiento rotacional respecto al eje z, Figura 1.

El uso de técnicas de control tradicionales es suficiente para tareas básicas de estabilización de vuelo del Quadrotor, despegue y aterrizaje, en ciertos entornos de operación. Sin embargo, se han propuesto diversos algoritmos para mejorar la respuesta del sistema, optimizar los recursos computacionales, ampliar su desempeño ante nuevas tareas y operaciones de mayor complejidad.

El control de actitud del Quadrotor requiere el cálculo de u_1 , u_2 , u_3 y u_4 , en este trabajo se propone que esta tarea se realice mediante las ecuaciones (10)-(12) integrando una red neuronal B-spline por cada ganancia, K , que se requiere definir en el esquema. Por lo tanto, se propone la inclusión de doce RNBS para que se tengan controladores adaptativos, cubriendo un mayor número de condiciones operativas y escenarios no previstos en el diseño del sistema de control. La entrada para la red neuronal se define a través de señales de error que provienen de las variables de interés, en este caso ángulos y desplazamiento traslacional.

Con la estructura propuesta el controlador se adapta a la condición de diseño original y sigue aprendiendo de manera continua en línea las diversas condiciones operativas del sistema, cambios paramétricos e incertidumbres. La Figura 2 muestra un diagrama de bloques del control a lazo cerrado del Quadrotor considerando la estructura que se representa mediante las ecuaciones (10)-(13). Asimismo, se puede apreciar como la red neuronal interactúa con la ley de control determinando los mejores valores para las ganancias que se tienen que definir para una correcta operación.

4. Diseño del controlador neuronal

La propuesta consiste en incluir una red neuronal B-spline para que calcule cada una de las doce ganancias asociadas al esquema de control. La red neuronal actualiza sus pesos sinápticos en cada muestra obtenida de las variables de retroalimentación del Quadrotor, Figura 2. Para este propósito se han definido doce redes neuronales B-spline con una estructura semejante, lo que cambia para cada uno de los casos son las entradas para su aprendizaje y el índice de aprendizaje. El número de neuronas, de pesos sinápticos y funciones base es de las mismas características para todas ellas, Figura 3. Este tipo de redes neuronales son un caso particular que permiten controlar y modelar sistemas de una forma adaptativa, con la opción de llevar a cabo su tarea en línea, tomando en consideración las incertidumbres del sistema y las no linealidades inherentes como consecuencia de que aprenden de mediciones del sistema físico. En esta estrategia de control, el diseño inicial considera las condiciones operativas actuales del sistema, estructura, características de las entradas y su respuesta. Sin embargo, su naturaleza adaptativa permite que la red neuronal vaya aprendiendo del desempeño del sistema y actualizando las señales de control como una consecuencia de la posible modificación de la estructura, entradas y entorno. En este sentido, se puede considerar que a pesar de que en el diseño del controlador neuronal no se tomaron en cuenta un gran número de condiciones operativas, variaciones paramétricas en el sistema o en su entorno, el mismo aprendizaje y actualización de los pesos le permitirá considerar estos aspectos en las futuras señales de control y así garantizar un desempeño satisfactorio en un amplio rango de condiciones operativas reales.

4.1. Esquema de la red neuronal

Una función B-spline es un mapeo polinomial que está formado por una combinación lineal de las funciones base monovariante y multivariante que se definen por sus extremos (Brown, Harris, 1994). Las redes de memoria asociativa como las B-spline ajustan sus pesos iterativamente con el objetivo de reproducir una función especificada, mientras que los algoritmos fuera de línea o por lotes típicamente generan los coeficientes mediante una matriz de inversión o empleando el conjugado del gradiente. La actualización de los pesos en las redes de memoria asociativa generalmente

utiliza algoritmos de mínimos cuadrados para realizar un mapeo particular, modificando el índice de peso de cada entrada para cada una de las funciones base participantes.

A través de las redes neuronales B-spline (RNBS) se tiene la posibilidad de acotar el espacio de entrada mediante la definición de las funciones base. Generalmente, únicamente un número limitado de funciones base participan en la salida de la red, en este caso una ganancia para cada red neuronal. Por lo tanto, no todos los pesos se tienen que calcular en cada dato de la señal de entrada, de esta manera se reduce el esfuerzo computacional y el tiempo de cálculo de las señales de control.

La salida de la RNBS se puede describir mediante (Brown *et al.*, 1994),

$$y = \mathbf{aw} \quad (14)$$

$$\mathbf{w} = [w_1 \ w_2 \ \dots \ w_h]^T, \quad \mathbf{a} = [a_1 \ a_2 \ \dots \ a_h]$$

donde w_j y a_j son el j -ésimo peso y la j -ésima salida de la función base, respectivamente; h es el número de pesos sinápticos.

En este trabajo se propone que las ganancias de control se calculen a través de las salidas de las RNBS. El espacio de entradas se construye a partir de señales de error obtenidas de la diferencia entre la señal de referencia y el valor real de la variable de interés, e_x , asimismo, se emplea la derivada del error, Figura 3. Por ejemplo la diferencia en θ respecto al valor de referencia θ_d y la diferencia en su derivada se emplean como entradas para la red neuronal y determinar las ganancias (10). El resto de las ganancias se calculan de manera semejante a partir de una RNBS con la misma estructura, Figura 3, cambiando la señal de entrada que depende de las variables x , y , z , ϕ , θ , y ψ , respectivamente.

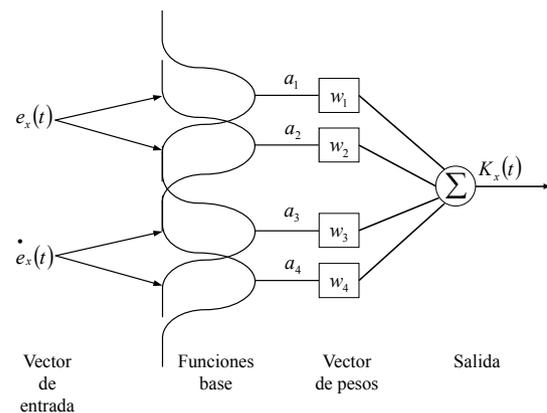


Figura 3: Estructura de la RNBS propuesta para calcular las señales de control.

La selección de las entradas se basa en la estrecha relación que existe entre los errores presentes de las variables de interés, las señales de control y la naturaleza de las redes de neuronas artificiales. Entonces, la dinámica de las ganancias se define como,

$$K_x = F(\mathbf{e}_x, \mathbf{w}) \quad (15)$$

donde F denota la red neuronal B-spline que se usa para calcular las doce ganancias; \mathbf{w} es el vector de pesos correspondiente para cada esquema propuesto; \mathbf{e}_x es el vector de entrada. La Figura 3 presenta el esquema de la red neuronal B-spline propuesta. En este trabajo se emplean doce esquemas con la misma estructura.

Un diseño apropiado de la red neuronal requiere la siguiente información previo a su definición: los valores que acotan las señales de entrada, e_x , el tamaño, la forma y traslape de las

funciones base. Esta información permite acotar la entrada de la red neuronal y mejorar la convergencia y estabilidad de la regla de entrenamiento instantáneo (Brown *et al.*, 1994). Por otra parte, con esta información la RNBS estima los mejores valores de los pesos de tal forma la señal de control consiga que el error de la variable controlada se hace igual a cero en un tiempo finito.

Conociendo esta información previa, se realizan algunos estudios fuera de línea que permitan establecer la estructura de la RNBS y las condiciones iniciales para su aprendizaje en línea. La Tabla 1 presenta valores de referencia a los cuales se sometió al Quadrotor, almacenando estos datos para entrenar la red neural definiendo estructura, funciones base, entradas. Como se puede observar y se menciona al inicio de esta sección, solo se consideran algunas condiciones operativas para el diseño inicial de la red neuronal, en este caso, algunas pruebas para seguimiento de referencias de las posiciones angulares primer columna, solo ecuación (10) fijando como entrada las referencias de ángulos y eje z, y otros experimentos para seguimiento de trayectorias en x, y, z, segunda columna ecuaciones (10)-(12), Tabla 1. Los datos del entrenamiento incluyen la respuesta transitoria y en estado estacionario del sistema, dentro de las ventajas del controlador neuronal propuesto, es que estos datos pueden provenir de un modelo matemático exacto o aproximado o por otra parte ser datos medidos del sistema real (entradas, salidas, señales de control, etc.). Lo que permite finalmente evitar la dependencia del controlador de un modelo matemático exacto, parámetros del sistema o modificaciones del entorno. Posteriormente, la red neuronal seguirá aprendiendo en línea de otras condiciones operativas y adaptándose para que la señal de control cumpla con el desempeño deseado.

Tabla 1: Condiciones operativas previas para un entrenamiento fuera de línea

Posición angular (grados)	Trayectoria (m)
$\theta = 8$	$x = 5$
$\theta = 7$	$y = 6$
$\theta = 12$	$z = 7$
$\phi = -6$	$z = 10$
$\theta = -5$	$x = 12$
$\phi = 1$	$y = 5$

En el análisis para la definición de la estructura de la RNBS se consideraron tres tipos de funciones base con diferente orden. Como resultado se propone el empleo de funciones base monovariante de tercer orden, debido a que presenta una respuesta rápida y estable en su entrenamiento. Estos valores se seleccionaron para alcanzar una respuesta suave en la dinámica del sistema. En los otros dos casos de funciones base multivariable y de cuarto orden se obtuvo un desempeño semejante al presentado en este trabajo, con la desventaja de que se requiere un mayor número de cálculos. Por lo tanto, se buscó un compromiso entre desempeño y costo computacional.

Para el esquema neuronal se han empleado cuatro funciones base dos para la entrada del error y otras dos para su derivada como se muestra en la Figura 3. Los puntos de definición de los extremos de las funciones base son [-8 -5 -2 1], [-0.6 0.4 1.4 2.4] para e y [-5 -3 -1 1], [-2.5 -0.5 1.5 3.5] para \dot{e} . Las doce RNBS utilizan esas mismas funciones base, lo que cambia es el origen de la señal de error que se normaliza en 1, se ha dejado un margen en la definición de los puntos de control para las funciones base considerando que se pudiera presentar una señal de error de mayor magnitud a 1 (positiva o negativa). Esta definición se desarrolló en la etapa de entrenamiento fuera de línea probando experimentalmente algunos

valores de control, sin embargo, todos los casos estuvieron dentro de un rango de -10 a 10 buscando que los cambios en la salida de las funciones base respecto a la señal de entrada no fueran tan abruptos, y en consecuencia la salida de la red neuronal.

4.2. Regla de aprendizaje

El aprendizaje en las redes de neuronas artificiales generalmente se alcanza mediante la minimización del error de la red, la cual es una medida de su desempeño, se define como la diferencia entre el vector de salida real de la red y el valor deseado. En esta propuesta el error se calcula a partir de las señales a controlar.

El aprendizaje en línea de funciones continuas generalmente se realiza vía métodos basados en el gradiente, teniendo la medida de la derivada del error. Es una de las técnicas más poderosas y comúnmente empleadas para entrenar redes de neuronas artificiales con alto número de capas en lo general y para tareas no estacionarias en lo particular (Saad, 1998).

En esta aplicación, se busca una actualización rápida de los parámetros. Las técnicas adaptativas convencionales son adecuadas para representar objetos con parámetros que cambian lentamente, también pueden tratar con problemas complejos de los denominados duros donde se incluyen múltiples modos de operación. Por otra parte, las reglas de entrenamiento instantáneo proveen una alternativa: tal que los pesos se actualizan continuamente alcanzando su convergencia a los valores óptimos si se condicionan adecuadamente. Además, algunas veces las redes convencionales no convergen, o su entrenamiento lleva una gran cantidad de tiempo (Saad, 1998; Osman, Abdelazim, Malik, 2005).

En este artículo, la red neuronal B-spline se entrena en línea empleando la siguiente regla de aprendizaje instantáneo para la corrección del error (Saad, 1998),

$$w_i(t) = w_i(t-1) + \frac{\eta e_i(t)}{\|a(t)\|_2} a_i(t) \quad (16)$$

donde η es la relación de aprendizaje y $e_i(t)$ es la salida instantánea del error, Ej. $e_\theta = \theta_d - \theta$.

Respecto a la relación de aprendizaje, este toma como valor inicial una magnitud en el intervalo de [0, 2] debido a propósitos de estabilidad (Brown *et al.*, 1994). Este valor se ajusta mediante un proceso iterativo en intervalos de 0.005 midiendo las características de respuesta dinámica de la red neuronal. Si η es cercano a cero, el aprendizaje se hace lento. Por el contrario, si este valor es grande, se pueden presentar y crecer las oscilaciones. En este trabajo el mejor desempeño dinámico se alcanzó con un valor de 0.01, que se obtuvo mediante pruebas realizadas fuera de línea, posteriormente, en el aprendizaje en línea permanece constante para todo el estudio.

Se propone que durante el procedimiento de actualización, se incluya una banda muerta que permita mejorar la convergencia de la regla de aprendizaje. Los factores de peso no se actualizan si la magnitud del error tiene un valor menor a 0.1%,

$$w_i(t) = \begin{cases} w_i(t-1) + \frac{\eta e_i(t)}{\|a(t)\|_2} a_i(t), & \text{si } |e_i| > 0.001 \\ w_i(t-1), & \text{cualquier otro caso} \end{cases} \quad (17)$$

Se ha elegido esta regla de aprendizaje como una alternativa a los métodos que emplean, por ejemplo, algoritmos de Newton para actualizar los pesos (Saad, 1998; Osman *et al.*, 2005) que requieren la evaluación de la matriz Jacobiana y Hessiana. En lo que respecta

a la actualización de los pesos de (16) se debe aplicar para cada conjunto de valores entrada-salida en cada valor instantáneo de la señal; la actualización se presenta si el error es diferente de cero, (16). Si se define y condiciona adecuadamente la regla de aprendizaje y sus parámetros, es posible que los pesos converjan a su valor óptimo (Brown *et al.*, 1994). En este trabajo se ha conseguido ajustando experimentalmente η y la zona muerta de aprendizaje (17), funcionando adecuadamente para el modelo expuesto y probado en las trayectorias presentadas.

De esta manera, la propuesta consiste fundamentalmente en establecer la estructura de la RNBS (definición y número de funciones base, número y características de las variables de entrada) y su relación de aprendizaje. De este modo, el proceso de entrenamiento se lleva a cabo de manera continua en línea, mientras que los valores de los pesos se actualizan empleando dos variables de retroalimentación.

5. Resultados y análisis

Algunas simulaciones numéricas fueron realizadas para verificar el desempeño eficiente y robusto del esquema de control neuronal propuesto. Se presentan los siguientes casos de estudio: (i) con el controlador adaptativo propuesto (ganancias dinámicas/variables); (ii) con un controlador no lineal basado en modos deslizantes de segundo orden. La simulación del Quadrotor con el controlador propuesto se realizó en Matlab® como una estrategia que permite analizar la efectividad del esquema. Además, para observar el comportamiento de la red neuronal, se considera que los valores iniciales de la posición y ángulos del Quadrotor son igual a cero al inicio de la simulación. Los parámetros empleados para el Quadrotor se muestran en la Tabla 2. Así, se desarrollan dos casos de estudio para ejemplificar la aplicación.

5.1. Primer caso de estudio

Los valores de referencia para x , y , z , junto con el instante de tiempo donde se presenta el cambio durante la simulación se enlistan en la Tabla 3. Estos resultados presentan el desempeño del esquema neuronal adaptativo con aprendizaje en línea.

Tabla 2: Valores de los parámetros del Quadrotor

Constante	Valor	Unidades
g	9.81	m/s^2
m	0.973	kg
b	1.15×10^{-5}	Ns^2
d	1.15×10^{-5}	Nms^2
l	0.225	m
I_r	2.86×10^{-5}	Nms^2
I_{xx}	8.66×10^{-3}	Nms^2
I_{yy}	8.66×10^{-3}	Nms^2
I_{zz}	2.29×10^{-7}	Nms^2

El seguimiento eficiente de la trayectoria de referencia para Quadrotor se describe en la Figura 4. Se puede observar que el controlador neuronal adaptativo consigue una respuesta transitoria con oscilaciones prácticamente nulas. Es claro que los picos para cada cambio en el valores de referencia indica un sobreimpulso presente en la respuesta transitoria, en este caso no se observa esta característica. Para el caso del esquema de control propuesto se tiene una respuesta suave en cada uno de los cambios de referencia, tanto para en el incremento del valor de referencia como para el regreso a una posición cercana a la inicial.

Tabla 3: Valores de referencia para la simulación

Variable	Valor (m)	Tiempo (seg)
z	7	0.5
y	25	12
x	15	30
y	2	42
x	2	60
z	6	75
x	10	90

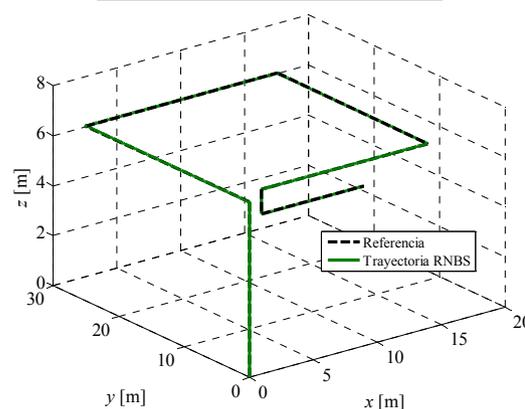


Figura 4: Seguimiento de trayectoria de posición del Quadrotor con controlador adaptativo, caso 1.

La Figura 5 muestra la respuesta temporal de posición en cada una las direcciones x , y , z . El gráfico describe el desempeño satisfactorio del seguimiento de la trayectoria planificada. Se puede observar que la estabilización converge a una condición operativa final que se establece mediante las señales de referencia, Tabla 3. En el periodo transitorio se exhibe la adaptabilidad del controlador propuesto, manteniendo un desempeño aceptable para los cambios en el valor de referencia.

Así, las Figuras 4-7 muestran las respuestas de seguimiento en lazo cerrado de los desplazamientos lineales y angulares del Quadrotor como resultado del uso del esquema de control propuesto. Las señales de control aplicadas al Quadrotor que permiten alcanzar la estabilización y el seguimiento de los movimientos de *pitch*, *roll* y *yaw*, y por lo tanto, la trayectoria deseada se muestran en la Figura 6. Se puede observar que las variables de entrada de control se encuentran acotadas, y como es de esperarse se presentan respuestas rápidas ante cambios de la señal de referencia. Esas señales de control permiten que el sobreimpulso en las variables de interés disminuya en los casos donde se aplica el controlador neuronal, en tal sentido, esta magnitud es igual o cercana a cero.

La Figura 6 exhibe que las cuatro señales de control convergen a un valor en estado estable después de entre tres y cinco segundos, ($u_1=9.83$ N, $u_2=0$ N-m, $u_3=0$ N-m, $u_4=0$ N-m), estos resultados coinciden con lo esperado para el funcionamiento del Quadrotor. La magnitud de las señales de control solo cambia cuando se presenta una perturbación o un cambio en el valor de referencia de entrada. Los valores de las señales de control calculados con el controlador neuronal permiten alcanzar un buen desempeño dinámico del seguimiento de las trayectorias especificadas para el Quadrotor.

Las señales de referencia para la posición angular se calculan mediante (11), la evolución de estas variables se presenta en la Figura 7, considerando el desempeño de la estrategia de control adaptativa propuesta. Para alcanzar estos resultados se definen únicamente como entrada los valores de referencia para las trayectorias en x , y , z , Tabla 3.

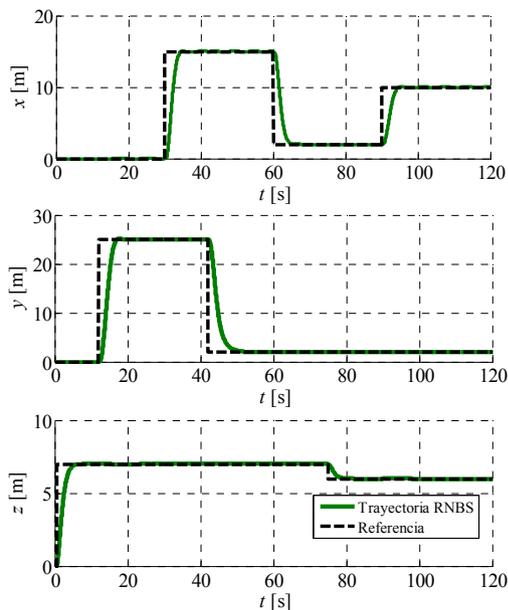


Figura 5: Seguimiento de las trayectorias de referencia de posición planificadas para cada una de las direcciones (x , y , z) usando el controlador neuronal adaptativo, caso 1.

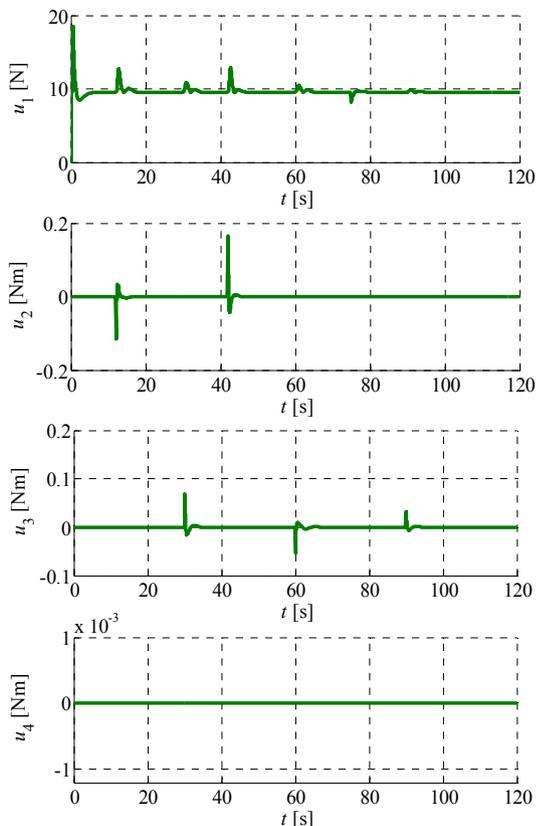


Figura 6: Señales de control asociadas al modelo adaptativo, caso 1.

En las Figuras 8 y 9, se ejemplifica la evolución de las ganancias de cuatro de los esquemas propuestos, donde los valores se actualizan mediante una red neuronal B-spline con entrenamiento continuo en línea. Para el primero, Figura 8, que corresponde al control del movimiento traslacional en x y en z , en las primeras muestras se tiene un valor en estado estacionario como condición

inicial. Es importante mencionar que se realizó un entrenamiento fuera de línea para calcular las condiciones iniciales, sin embargo, el valor óptimo cuando varía la condición operativa puede requerir grandes cambios en los valores de los pesos. Esto se puede conseguir debido a la naturaleza de la red neuronal y su algoritmo de aprendizaje en línea, que actualiza su desempeño en cada instante de tiempo.

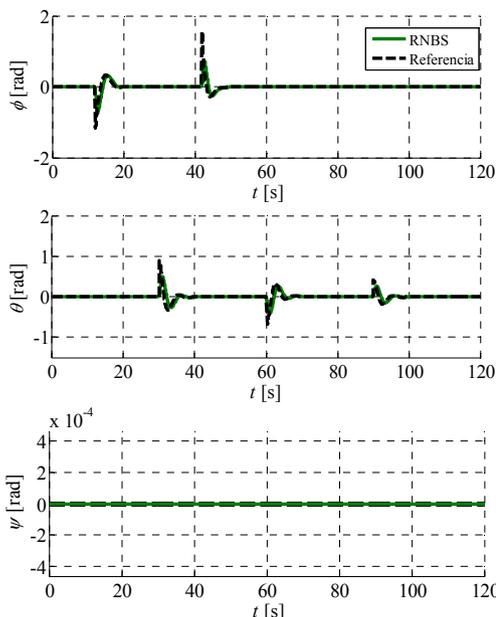


Figura 7: Evolución de las posiciones angulares ϕ , θ , ψ , caso 1.

Posteriormente, la red neuronal continúa aprendiendo acerca del comportamiento de las variables de interés y modos de operación del Quadrotor, los pesos se actualizan hasta alcanzar un valor en estado estacionario. La diferencia más clara se presenta en el periodo transitorio, el controlador propuesto ayuda a que los sobreimpulsos sean de menor magnitud y además el tiempo de establecimiento sea en un tiempo más corto. La regla de aprendizaje en línea responde más rápido cuando las variables de interés tienen un valor lejano del esperado, Figuras 8 y 9. Se obtiene un comportamiento semejante para las doce estructuras de redes neuronales B-spline. En la Figura 9 se muestra el comportamiento de otras dos ganancias, estas asociadas al control de ϕ y θ .

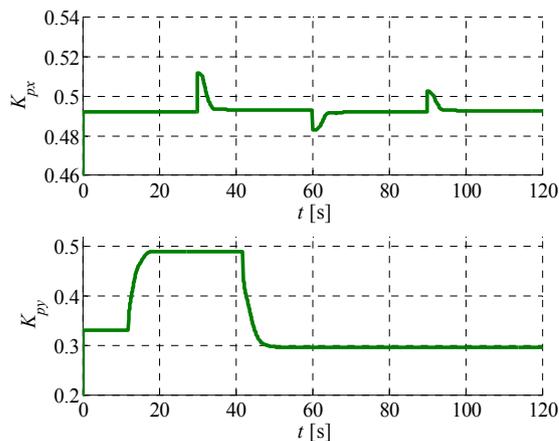


Figura 8: Evolución de dos ganancias asociadas a la ecuación para las trayectorias en x , y , caso 1.

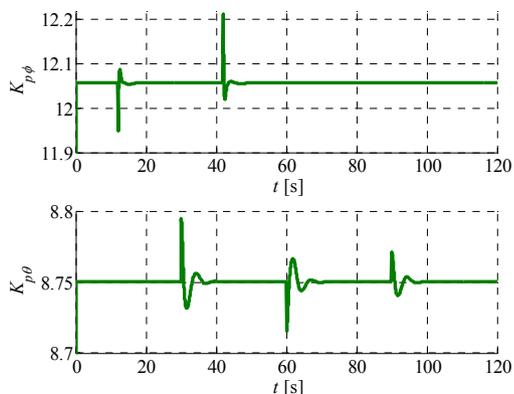


Figura 9: Evolución de dos ganancias asociadas a la ecuación para los ángulos ϕ, θ , caso 1.

5.2. Segundo caso de estudio

Para el segundo caso se realiza un seguimiento de trayectoria donde los valores de referencia en los ejes x, y son funciones del tiempo dadas por

$$x_d = r \cos \omega t$$

$$y_d = r \sin \omega t$$

donde $r=1$ m y $\omega=0.5$ rad/seg.

Además, el desempeño del seguimiento de trayectoria del esquema de control propuesto se compara con los controladores robustos no lineales basados en modos deslizantes de segundo orden presentados en (En-Hui Z et al., 2014).

La Figura 10a exhibe la evolución del sistema en lazo cerrado cuando se cambia la naturaleza de la señal de referencia ahora el seguimiento de trayectoria es función del tiempo. El algoritmo propuesto mantiene un alto desempeño siguiendo la referencia sin que se presente un cambio brusco en las señales controladas, se observa un error inicial en la trayectoria, posteriormente, alcanza el valor deseado. La respuesta se compara con los resultados obtenidos mediante el controlador por modos deslizantes de segundo orden, Figura 10b.

La evolución del desplazamiento en x, y, z se presenta en la Figura 11 para el controlador propuesto y en la Figura 12 con la estrategia no lineal por modos deslizantes. Se puede observar que el cambio en las salidas de la red neuronal ayuda a mantener un desempeño coherente respecto al caso 1, no se degrada su desempeño ante el cambio en la función de referencia. Se observa un cambio drástico cuando se incluye la modificación en la señal de referencia, posteriormente, alcanza y sigue el valor deseado. Por el contrario, con la estrategia con modos deslizantes se observa un error en estado estacionario. La línea punteada en estas gráficas representa el valor de referencia deseado.

Las señales del control neuronal, u_1, u_2, u_3 y u_4 se exhiben en la Figura 13, su evolución produce el comportamiento observado en las Figuras 10a y 11. La Figura 14 presenta las señales de control usando modos deslizantes de segundo orden. Aquí, es evidente la presencia de oscilaciones (*chattering*) de alta frecuencia del control discontinuo. Además, se puede observar que el seguimiento adecuado de la planificación de movimiento especificada para el Quadrotor demanda un esfuerzo de control mayor usando modos deslizantes.

Nótese que, en el esquema de control propuesto, doce ganancias se están actualizando conforme se presentan cambios en el sistema. Por lo tanto, si se tiene bien condicionado el algoritmo, es posible alcanzar un desempeño aceptable para diferentes tipos de tareas de seguimiento de trayectorias de referencia.

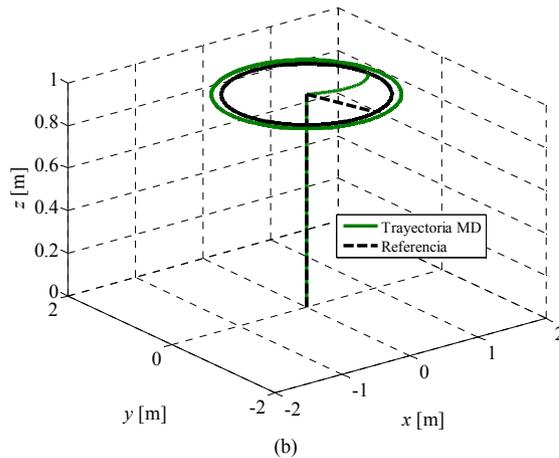
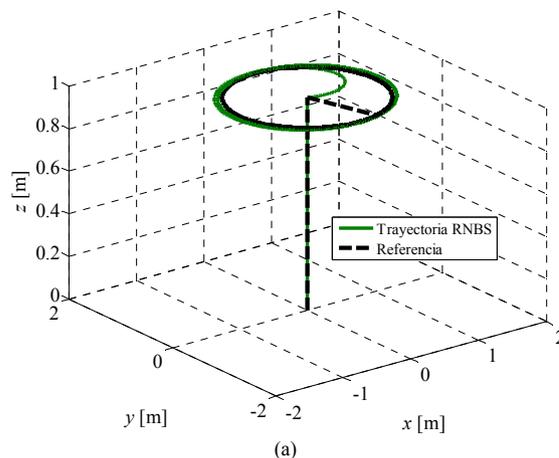


Figura 10: Seguimiento de trayectoria de posición del Quadrotor: a) controlador adaptativo; b) controlador no lineal, caso 2.

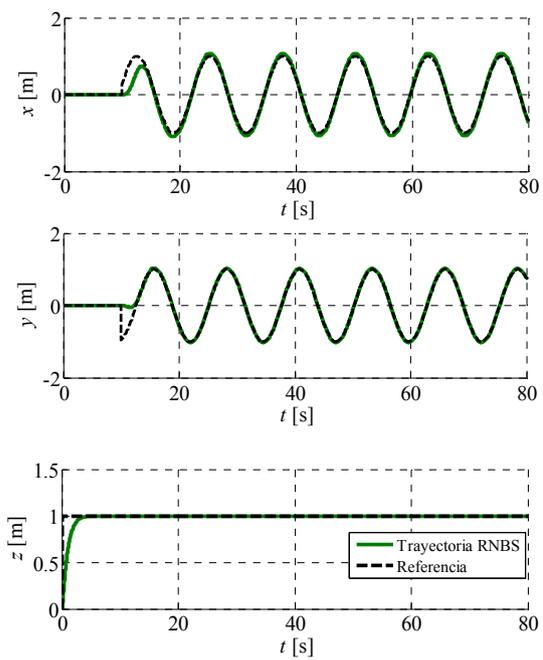


Figura 11: Seguimiento de trayectorias de referencia (x, y, z) con controlador neuronal adaptativo, con referencia definida en el caso 2.

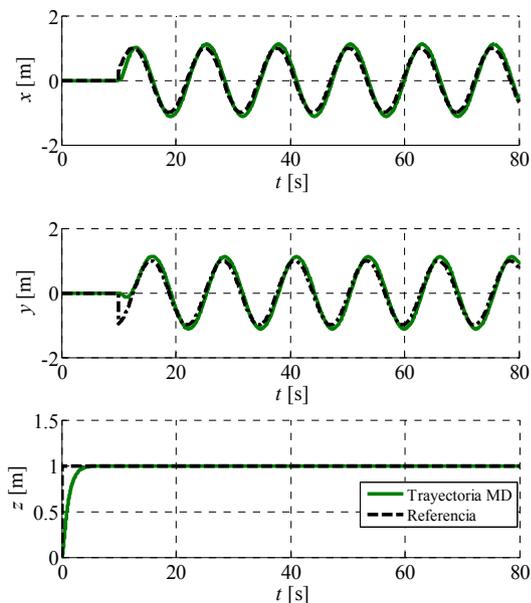


Figura 12: Seguimiento de trayectorias (x, y, z) con controlador no lineal, con referencia definida en el caso 2.

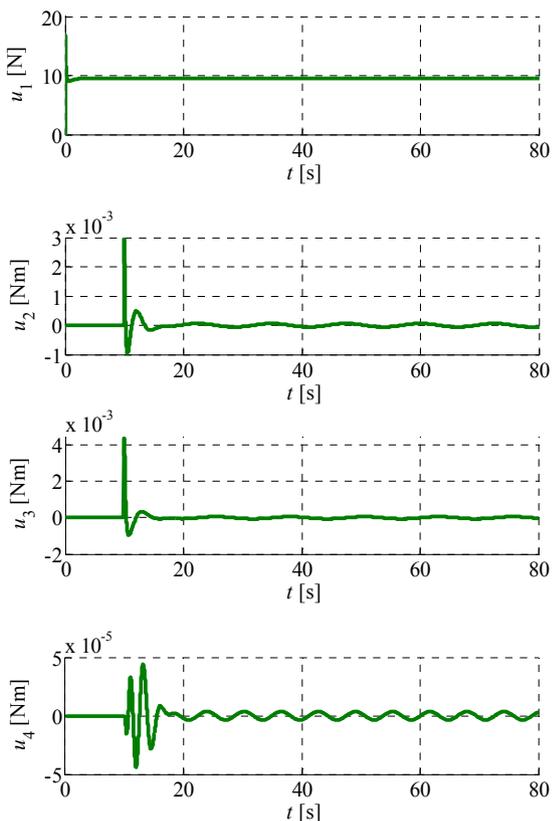


Figura 13: Entradas de control obtenidas con el controlador adaptativo.

Así, los resultados ilustran el impacto del diseño propuesto y la respuesta en el tiempo al aplicar la técnica de control adaptativa en el desempeño del Quadrotor, el estudio consideró altos requerimientos de operación. La red neuronal B-spline utilizada como controlador exhibe mejor desempeño, adaptándose por sí sola a nuevas circunstancias presentes en el sistema. En el estudio realizado y las condiciones operativas expuestas se observa un

desempeño satisfactorio y estable de la dinámica en lazo cerrado del Quadrotor. Los resultados en simulación verifican el seguimiento de las trayectorias planificadas sobre el modelo matemático no lineal completo. La comparación con una técnica no lineal basada en modos deslizantes ejemplifica la utilidad mayor de la propuesta neuronal adaptativa, sin la presencia de oscilaciones de alta frecuencia en las entradas de control.

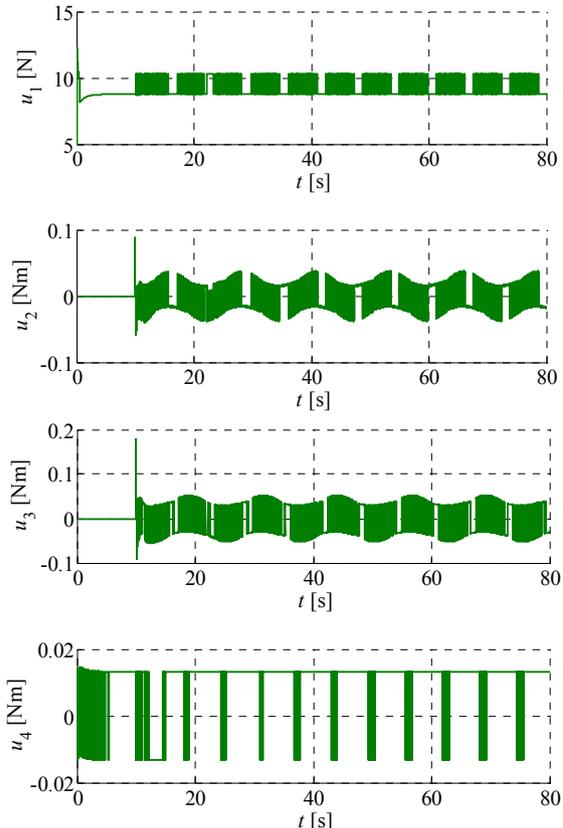


Figura 14: Entradas de control con el controlador no lineal basado en modos deslizantes, caso 2.

6. Conclusión

En este artículo se ha propuesto un esquema de control neuronal para tareas de regulación y seguimiento de trayectorias de referencia planificadas para un Quadrotor. El enfoque de control presentado representa una muy buena alternativa debido a su simplicidad de aprendizaje en línea y a su naturaleza adaptativa ante diversas condiciones de operación. Diferente a algunas contribuciones de control reportadas en la literatura, las doce ganancias del control se adaptan de manera eficiente ante cambios en las señales de referencia, usando redes neuronales, sin requerir un modelo matemático detallado de la dinámica no lineal del sistema. Los resultados de simulación verifican el buen desempeño del control de seguimiento propuesto. Se incluyeron algunas comparaciones con un esquema de control no lineal basado en modos deslizantes de segundo orden, para mostrar que el uso apropiado de redes neuronales permite obtener un desempeño de seguimiento satisfactorio, sin la presencia de oscilaciones de alta frecuencia. Futuros trabajos están orientados a combinar nuestro enfoque de control del tipo adaptable con otras metodologías de diseño de control.

English Summary

On Line Adaptive Neurocontroller for Regulating Angular Position and Trajectory of Quadrotor System

Abstract

Automatic control systems every day become more important in everyday life; therefore, it must find new and better ways to incorporate mathematical models and adaptive control algorithms to cope with a number of technical and physical challenges for exploitation. In this paper, the algorithm of the dynamic model of a Quadrotor applied to an angular position and trajectory control as a study case is detailed. Due to nonlinear nature of this type of systems, an adaptive on line neurocontroller algorithm based on B-spline neural networks is proposed, the learning procedure is divided in two stages: a) an initial off line training and; b) an on line continuous learning. This form of learning allows the Quadrotor extend its satisfactory performance at different operating conditions and trajectory tracking. The simulation results demonstrate the applicability of the developed model and the impact of dynamic control on the system performance, diminishing the exact model requirement and the possibility to incorporate the system non linearities.

Keywords:

Neural Network Control, Model-Free Control, Automatic Learning, Underactuated Systems.

Agradecimientos

Este trabajo ha sido apoyado por CONACYT bajo el proyecto 266333.

Referencias

- Argentim L. M., Rezende W. C., Santos P. E., Aguiar R. A., 2013. PID, LQR and LQR-PID on a Quadcopter Platform. Proc. of the IEEE International Conference on Informatics, Electronics & Vision (ICIEV), 1-6. DOI: 10.1109/ICIEV.2013.6572698
- Bauer P., Ritzinger G., Soumelidis A., Bokor J., 2008. LQ Servo control design with Kalman filter for a quadrotor UAV. Periodica Polytechnic Transportation Engineering 36(1-2), 9-14. DOI: 10.3311/pp.tr.2008-1-2.02
- Benallegue A., Mokhtari A., Fridman L., 2008. High-order sliding-mode observer for a quadrotor UAV. International Journal of Robust and Nonlinear Control 18(4-5), 427-440. DOI: 10.1002/rnc.1225
- Bouabdallah S., Noth A., Siegwart R., 2004. PID vs LQ Control Techniques Applies to an Indoor micro quadrotor. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems 3, 2451-2456. DOI: 10.1109/IROS.2004.1389776
- Bouabdallah S., Siegwart R., 2005. Backstepping and Sliding-mode Techniques. Applied to an Indoor Micro Quadrotor. Proc. of the IEEE International Conference on Robotics and Automation, 2247-2252. DOI: 10.1109/ROBOT.2005.1570447
- Bouabdallah S., Siegwart R., 2007. Full Control of a Quadrotor. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 153-158. DOI: 10.1109/IROS.2007.4399042
- Boudjedir H., Bouhali O., Rizoug N., 2014. Adaptive neural network control based on neural observer for a quadrotor unmanned aerial vehicle. Advanced Robotics 28(17), 1151-1164. DOI: 10.1080/01691864.2014.913498
- Bresciani T., 2008. Modeling, Identification and Control of a Quadrotor Helicopter. Master Thesis. Department of Automatic Control, Lund University, Suecia.
- Brown M., Harris C., 1994. Neurofuzzy Adaptive Modelling and Control. Prentice Hall International, New York.
- Castillo P., García P., Lozano R., Albertos P., 2007. Modelado y estabilización de un Helicóptero con cuatro rotores. Revista Iberoamericana de Automática e Informática Industrial 4(1), 41-57. DOI: 10.1016/S1697-7912(07)70191-7
- Chee K.Y., Zhong Z.W., 2013. Control, navigation and collision avoidance for an unmanned aerial vehicle. Sensors and Actuators A: Physical 190, 66-76. DOI: 10.1016/j.sna.2012.11.017
- Dierks T., Jagannathan S., 2010. Output Feedback Control of a Quadrotor UAV Using Neural Networks. IEEE Transactions on Neural Networks 21(1), 50-66. DOI: 10.1109/TNN.2009.2034145
- Dikmen I. C., Arisoy A., Temeltas H., 2009. Attitude Control of a Quadrotor. Proc. of the 14th IEEE International Conference on Recent Advances in Space Technologies, 722-727. DOI: 10.1109/RAST.2009.5158286
- Dydek Z.T., Annaswamy A.M., Lavretsky E., 2013. Adaptive configuration control of multiple UAVs. Control Engineering Practice 21(8), 1043-1052. DOI:10.1016/j.conengprac.2013.03.010
- Emran B., Yesildirek A., 2014. Robust Nonlinear Composite Adaptive Control of Quadrotor. International Journal of Digital Information and Wireless Communications 4(2), 213-225. DOI: 10.17781/P001100
- En-Hui Z., Jing-Jing X., Ji-Liang L., 2014. Second order sliding mode control for a quadrotor UAV. ISA Transactions 53(4), 1350-1356. DOI: 10.1016/j.isatra.2014.03.010
- Erginer B., Altug E., 2007. Modeling and PD Control of a Quadrotor VTOL Vehicle. Proc. of the IEEE Intelligent Vehicles Symposium, 894-899. DOI: 10.1109/IVS.2007.4290230
- Estelles S., Tomas-Rodríguez M., 2015. Quadrotor multibody modelling by vehiclesim: adaptive technique for oscillations in a PVA control system. Journal of Vibration and Control. DOI: 10.1177/1077546315619776.
- Fatan M., Sefidgari B. L., Barenji A. V., 2013. An Adaptive Neuro PID for Controlling the Altitude of Quadcopter Robot. Proc. of the IEEE 18th International Conference on Methods and Models in Automation and Robotics (MMAR), 662-665. DOI: 10.1109/MMAR.2013.6669989
- Hoffmann G. M., Waslander S. L., Tomlin C. J., 2008. Quadrotor Helicopter trajectory tracking control. In AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, USA.
- Jing-Jing X., En-Hui Z., 2014. Position and attitude tracking control for a quadrotor UAV. ISA Transactions 53(3), 725-731. DOI: 10.1016/j.isatra.2014.01.004
- Johnson N. L., Leang K. K., 2013. Enhanced proportional-derivative control of a micro Quadcopter. Proc. of the ASME Dynamics Systems and Control Conference, 1-5. DOI:10.1115/DSCC2013-3990
- Luque-Vega L., Castillo-Toledo B., Loukianov A. G., 2012. Robust block second order sliding mode control for a quadrotor. Journal of the Franklin Institute 349(2), 719-739. DOI: 10.1016/j.jfranklin.2011.10.017
- Meng Leong B. T., Ming Low S., Po-Leen Ooi M., 2012. Low-Cost Microcontroller-based Hover Control Design of a Quadcopter. Proc. of the International Symposium on Robotics and Intelligent Sensors (IRIS) 41, 458-464. DOI: 10.1016/j.proeng.2012.07.198
- Mian A. A., Daobo W., 2008. Modeling and Backstepping-based Nonlinear Control Strategy for a 6 DOF Quadrotor Helicopter. Chinese Journal of Aeronautics 21(3), 261-268. DOI: 10.1016/S1000-9361(08)60034-5
- Mian A. A., Mian I. A., Daobo W., 2008. Backstepping based PID Control Strategy for an Underactuated Aerial Robot. Proc. of the 17th IFAC World Congress 17, 15636-15641. DOI: 10.3182/20080706-5-KR-1001.02644
- Mohamed Raju H., 2010. Dynamics Modeling and Control of a Quad-Rotor Helicopter. Master Thesis. Memorial University of Newfoundland, Faculty of Engineering & Applied Science, Canada.
- Nicol C., Macnab C. J. B., Ramirez-Serrano A., 2008. Robust Neural Network Control of a Quadrotor helicopter. Proc. of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 1233-1238. DOI: 10.1109/CCECE.2008.4564736
- Osman A. H., Abdelazim T., Malik O. P., 2005. Transmission Line Distance Relaying Using on-line Trained Neural Network. IEEE Transactions on Power Delivery 20(2), 1257-1264. DOI: 10.1109/TPWRD.2004.833897
- Pipatpaibul P. -i., Ouyang P. R., 2013. Application of Online Iterative Learning Tracking Control for Quadrotor UAVs. ISRN Robotics Volume 2013. ID 476153.
- Saad D., 1998. On-line learning in neural networks. Cambridge University Press, UK.
- Salih A. L., Moghavvemi M., Mohamed H. A., Gaeid K. S., 2010. Flight PID controller design for a UAV quadrotor. Scientific Research and Essays 5(23), 3660-3667.