

CONTROL POR REDES NEURONALES DE UN CUADRICÓPTERO

CRISTIAN DAVID MARIN ARANGO

SANTIAGO ORTIZ OSORIO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

INGENIERÍA EN MECATRÓNICA

PEREIRA - RISARALDA

2022

CONTROL POR REDES NEURONALES DE UN CUADRICÓPTERO

CRISTIAN DAVID MARIN ARANGO

SANTIAGO ORTIZ OSORIO

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO
MECATRÓNICO**

DIRECTOR

M.Sc. ANGELO ANTONIO MARIN LÓPEZ

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

INGENIERÍA EN MECATRÓNICA

PEREIRA - RISARALDA

2022

DEDICATORIA

Cristian Marin “Este trabajo va dedicado a mis papás por todo el apoyo emocional prestado, por preocuparse por mi bienestar cada día que asistía a mis clases, por sus consejos incondicionales cada vez que pensaba en desfallecer de este proceso.

Se lo dedico a mi pareja por su paciencia y entender que los compromisos académicos en muchas ocasiones no daban espera y era necesario postergar nuestros planes.

Mis jefes y compañeros quienes dieron apoyo profesional aportando tranquilidad a la hora de abrir espacios en los compromisos laborales con el fin de poder cumplir con las obligaciones pactadas.”

Santiago Ortiz “a Dios, creador de todas las cosas existentes en este mundo, que, gracias a su espíritu y voluntad, me presento esta oportunidad para ser vasto en su camino, adhiriéndome al bien de su intención y guiándome en este procedimiento sin desamparo alguno al éxito.

A mi madre, por la luz que cobija mi esperanza frente a mis logros y futuro dándome el apoyo imbatible de su amor, soportando mis errores y educándome en la senda de lo correcto.

A mi familia, a mi abuela por su bondad, misericordia, ternura y sabiduría incondicional emanada desde las entrañas del amor y la visión de lo que ella algún día quiso ver en mí. A mis primos por su apoyo, lealtad y alegrías, a mi hermana por brindarme la felicidad de cada día y sembrar en mi la meta de ser un hombre mejor. Por último, a mis tíos por el orgullo de verme lo que soy.”

AGRADECIMIENTOS

A Dios nuevamente por ser tan importante en nuestras vidas y cada uno de nuestros pasos y metas que tenemos por cumplir. A nuestros maestros, por su gran entusiasmo, motivación y buena enseñanza que se incentiva para la inclinación de esta gran culminación de nuestros estudios profesionales. Impulsando un desarrollo íntegro en nuestra parte profesional, por su valioso y bien entregado tiempo.

A nuestro director de tesis el profesor Angelo Antonio Marin López, por todo el tiempo de asesoría y sus conocimientos impartidos tanto en las clases como durante el desarrollo de este proyecto y así la culminación satisfactoria de cada uno de los objetivos planteados.

Tabla de contenido

| | |
|--|----|
| Índice de figuras | 7 |
| Resumen | 8 |
| Introducción | 9 |
| Capítulo 1 | 10 |
| 1. Modelos matemáticos de los ángulos del cuadricoptero | 10 |
| 1.1 Fórmula de <i>Haversine</i> (Verseno)..... | 10 |
| 1.2 Medición del ángulo entre dos puntos | 10 |
| 1.3 Cálculo de una nueva coordenada a partir de un punto inicial, la distancia al punto inicial y un ángulo | 11 |
| 1.4 Ángulos de navegación o ángulos de Euler..... | 12 |
| Capítulo 2 | 13 |
| 2. Sistema de control por redes neuronales | 13 |
| 2.1 controladores | 13 |
| 2.1.1 Control en lazo abierto..... | 13 |
| 2.1.2 Control en lazo cerrado..... | 14 |
| 2.2 Controlador PID | 14 |
| 2.3 Control por redes neuronales | 15 |
| 2.3.1 Control por modelo inverso..... | 15 |
| 2.3.2 Controlador con corrección por modelo inverso | 16 |
| Capítulo 3 | 18 |
| 3. Identificación del sistema MIMO | 18 |
| 3.1 Representación del sistema MIMO..... | 18 |
| 3.2 Esquemas de identificación en línea | 21 |
| 3.3 Mínimos cuadrados | 22 |
| 3.3.1 Forma recursiva..... | 24 |
| 3.4 Mínimos cuadrados con factor de olvido..... | 26 |
| 3.5 Proyección..... | 27 |
| 3.6 Proyección ortogonal..... | 28 |
| Capítulo 4 | 30 |
| 4. Análisis y resultados | 30 |
| 4.1 Aprendizaje de la red neuronal..... | 31 |
| 4.2 Control con corrección por modelo inverso | 34 |

| | |
|--------------------|----|
| Conclusiones..... | 37 |
| Bibliografía | 38 |

Índice de figuras

| | |
|---|----|
| Ilustración 1- Ángulos de navegación según convención ZXY. | 12 |
| Ilustración 2 – Diagrama de bloques de un sistema en lazo abierto. | 13 |
| Ilustración 3 – Diagrama de bloques de un sistema en lazo cerrado. | 14 |
| Ilustración 4 – Diagrama de bloques de un sistema en lazo cerrado con controlador PID. | 15 |
| Ilustración 5 – Control por modelo inverso de un sistema | 16 |
| Ilustración 6 – Controlador con corrección por modelo inverso | 17 |
| Ilustración 7 – Diagrama de bloques de aprendizaje por una red neuronal directa. | 18 |
| Ilustración 8 – Cuadricóptero JJR/C. | 30 |
| Ilustración 9 – Control remoto del cuadricóptero | 31 |
| Ilustración 10 – Datos obtenidos de la planta real | 32 |
| Ilustración 11 – Aprendizaje de la red usando mínimos cuadrados con factor de olvido | 33 |
| Ilustración 12 – Aprendizaje de la red usando proyección ortogonal. | 34 |
| Ilustración 13 – Salida controlada del movimiento Pitch..... | 35 |
| Ilustración 14 – Salida controlada del movimiento Yaw | 35 |
| Ilustración 15 – Salida controlada del movimiento Roll..... | 36 |

RESUMEN

Este proyecto de grado presenta el diseño de un controlador por redes neuronales para un sistema multivariable como lo es el cuadricóptero. Se utilizó un modelo identificado en tiempo real del sistema.

Se procedió a estimar la dinámica del modelo a través de un algoritmo de regresión lineal fuera de línea para sistemas MIMO, para luego acoplarlo con el diseño del controlador por la técnica de redes neuronales. Se realizó la simulación al sistema multivariable, con resultados satisfactorios que llevaron a cumplir los objetivos de este proyecto.

El software implementado para la identificación y posterior control de la planta fue Matlab.

INTRODUCCIÓN

El objetivo de este trabajo es la aplicación de los conocimientos adquiridos, donde nos centramos en el diseño de controladores inteligentes por redes neuronales. Para ello elegimos como sistema un vehículo aéreo no tripulado llamado *cuadricóptero*. La razón por la cual este problema es interesante desde el punto de vista de control es por que ilustra muchas de las dificultades asociadas con problemas de control del mundo real, como por ejemplo no linealidades y sistemas multivariables. [1]

En la actualidad, la Ingeniería de control ha sido impulsada por la investigación de la robótica debido a la necesidad de automatizar todo lo relacionado con procesos industriales que involucren el uso de maquinaria y por ende pongan en riesgo la salud de las personas, por este motivo el desarrollo en simples campos como el transporte, la producción y las comunicaciones obtienen un alto grado de seguridad y calidad. Por esto es por lo que la ciencia, precisamente el desarrollo de los sistemas mecatrónicos que tienen elementos que representan no linealidades, son objeto de investigación y un desafío para las técnicas de control. El cuadricóptero permite la aplicación y prueba de algoritmos de control inteligente que pueden ser usados en sistemas mecatrónicos más complejos (posibles aplicaciones en las áreas de robótica, navegación autónoma, sistemas de control de posición, etc.) [1]

Capítulo 1

1. MODELOS MATEMÁTICOS DE LOS ÁNGULOS DEL CUADRICÓPTERO

En este capítulo se presentan unas ecuaciones para observar la dinámica del sistema y por lo tanto el comportamiento de los ángulos en el cuadricóptero.

1.1 Fórmula de Haversine (Verseno): Representa la ecuación para determinar la distancia ortodrómica entre dos puntos de la tierra: es una ecuación ampliamente utilizada en la navegación, se utiliza a partir de las coordenadas en latitud y longitud de los dos puntos, la ecuación consiste en: [2]

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (1.1)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (1.2)$$

$$d = R \cdot c \quad (1.3)$$

Donde

φ_1 : Latitud inicial

φ_2 : Latitud final

λ_1 : Longitud inicial

λ_2 : Longitud final

$\Delta\varphi = \varphi_2 - \varphi_1$: Diferencia entre latitud final y latitud inicial

$\Delta\lambda = \lambda_2 - \lambda_1$: Diferencia entre longitud final y longitud inicial

$R = 6371$ Km: Radio geodésico de la tierra

1.2 Medición del ángulo entre dos puntos: Para realizar la medición se necesitan las coordenadas en latitud y longitud de ambos puntos, y se aplica la

siguiente fórmula: [2]

$$\theta = \text{atan} \left(\frac{\sin(\Delta\lambda) * \cos(\varphi_2)}{\cos(\varphi_1) * \sin(\varphi_2) - \sin(\varphi_1) * \cos(\varphi_2) * \cos(\Delta\lambda)} \right)$$

(1.4)

Donde:

θ : Ángulo entre los puntos

φ_1 : Latitud inicial

φ_2 : Latitud final

λ_1 : Longitud inicial

λ_2 : Longitud final

$\Delta\varphi = \varphi_2 - \varphi_1$: Diferencia entre latitud final y latitud inicial

$\Delta\lambda = \lambda_2 - \lambda_1$: Diferencia entre longitud final y longitud inicial

La anterior fórmula permite obtener una aproximación del ángulo inicial entre los dos puntos coordenados cuando son distancias cortas, el ángulo inicial será el ángulo definitivo entre los dos puntos; cuando son distancias largas, el ángulo va cambiando a medida que el dispositivo se acerca al objetivo.

1.3 Cálculo de una nueva coordenada a partir de un punto inicial, la distancia al punto inicial y un ángulo:

Para determinar una nueva coordenada, se utilizan las siguientes ecuaciones, las cuales entregan una latitud y una longitud a una distancia y dirección del punto inicial. [2]

$$\varphi_2 = \text{asin} \left(\sin(\varphi_1) * \cos\left(\frac{D}{R}\right) + \cos(\varphi_1) * \sin\left(\frac{D}{R}\right) * \cos(\theta) \right)$$

(1.5)

$$\lambda_2 = \lambda_1 + \text{atan} \left(\frac{\sin(\theta) * \sin\left(\frac{D}{R}\right) * \cos(\varphi_1)}{\cos\left(\frac{D}{R}\right) - \sin(\varphi_1) * \sin(\varphi_2)} \right)$$

(1.6)

θ : Ángulo entre los puntos

φ_1 : Latitud inicial

φ_2 : Latitud nueva

λ_1 : Longitud inicial

λ_2 : Longitud nueva

D: distancia a la cual se desea hallar la nueva coordenada

R=6371 Km: Radio geodésico de la tierra

1.4 ÁNGULOS DE NAVEGACIÓN O ÁNGULOS DE EULER

Los ángulos de navegación o matemáticamente ángulos de Euler, son ángulos que a través de coordenadas angulares entregan la posición en la que se encuentra una aeronave en un sistema de tres ejes fijos sobre esta, a estos se les conoce como cabeceo *pitch* representado por θ , alabeo *roll* representado por ϕ y guiñada *yaw* representada por ω . A continuación en la siguiente figura se muestra los ángulos de navegación para una aeronave. [2]

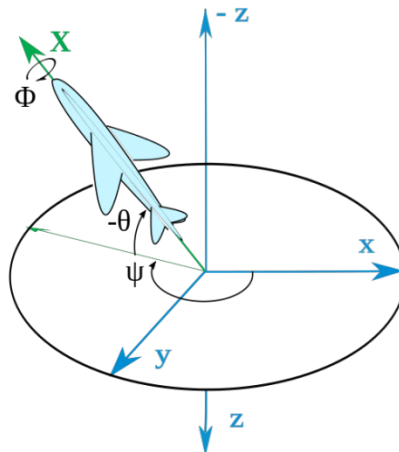


Ilustración 1- Ángulos de navegación según convención ZXY. [2]

Capítulo 2

2. SISTEMA DE CONTROL POR REDES NEURONALES

En este capítulo se presenta la teoría necesaria para el diseño del controlador inteligente, teniendo en cuenta que la red neuronal en algunos casos tarda en aprender se debe diseñar un controlador de estructura convencional el cual será corregido por el control inverso de red neuronal.

2.1 CONTROLADORES

El controlador representa el elemento básico de un sistema de control, ya que determina su comportamiento, condicionando la acción de los actuadores en función de la señal de desviación obtenida, todo relacionado en el mundo discreto a partir de un computador.

Los sistemas de control presentan un comportamiento eficiente cuando garantizan la estabilidad de la señal de salida con error en estado estacionario cero o menor al 2% ante perturbaciones en los modelos. [3][10]

2.1.1 CONTROL EN LAZO ABIERTO

En este tipo de controladores, la señal de salida no influye sobre la señal de entrada. La exactitud de estos sistemas depende de su programación previa y los ajustes que se le realicen al proceso. Se debe prever las relaciones que deben darse entre los diferentes componentes del sistema, a fin de tratar de conseguir que la salida alcance el valor deseado con la exactitud prevista. El diagrama de bloque de un sistema en lazo abierto se muestra a continuación en la ilustración 2.

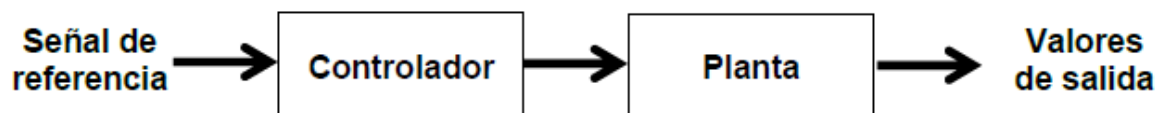


Ilustración 2 – Diagrama de bloques de un sistema en lazo abierto. [3]

Los sistemas de control en lazo abierto son comúnmente utilizados para dispositivos secuenciales y/o cíclicos, donde no se regulan variables sino que se realiza un conjunto de operaciones.

Un sistema en lazo abierto está dado por la función de transferencia:

$$\frac{y_m(z)}{E(z)} = G(z) * H(z) \quad (2.1)$$

2.1.2 CONTROL EN LAZO CERRADO

También llamados control con realimentación, tienen la característica de tener un transductor o dispositivo que censa la señal de salida del proceso constantemente y poder realizar una comparación con la señal de referencia obteniendo una desviación, para que el controlador pueda ejecutar la corrección y así tener un control más estable del proceso en ejecución. El diagrama de bloque de un sistema en lazo cerrado se muestra a continuación en la ilustración 3. [3]

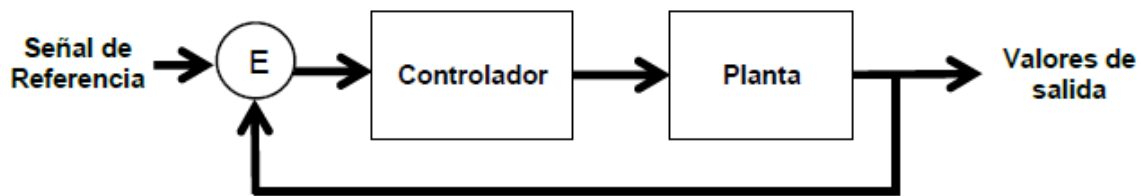


Ilustración 3 – Diagrama de bloques de un sistema en lazo cerrado. [3]

La función de transferencia de un sistema en lazo cerrado es:

$$\frac{Y(z)}{R(z)} = \frac{G(z)*H(z)}{1+G(z)*H(z)} \quad (2.2)$$

2.2 CONTROLADOR PID

Es un sistema de control polinomial por realimentación que calcula la desviación o error entre un valor de la señal de salida del proceso y el valor que se quiere obtener, y así aplicar una acción correctora que ajuste nuevamente la salida del proceso. [3]

Los controladores PID poseen tres tipos de acciones: proporcional (P), integral (I) y derivativa (D). La combinación de estos, reúne las ventajas de las tres acciones de control al mismo tiempo, tal como lo muestra la ilustración 4.

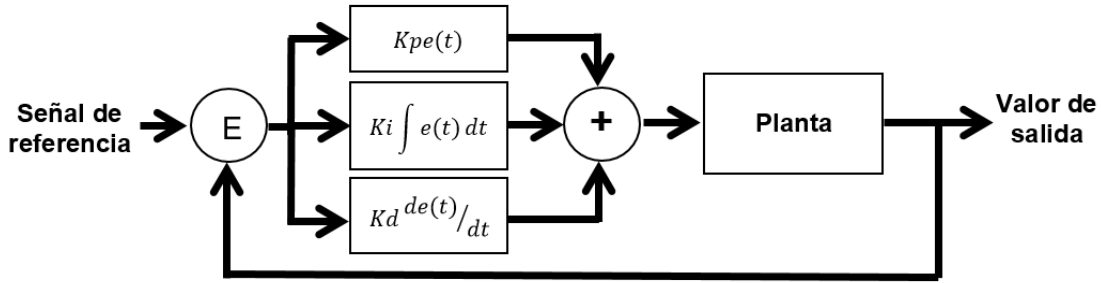


Ilustración 4 – Diagrama de bloques de un sistema en lazo cerrado con controlador PID. [8]

La función de transferencia de un controlador PID es:

$$G_c(s) = k_p + k_d s + \frac{k_i}{s} \quad (2.3)$$

2.3 CONTROL CON REDES NEURONALES

Teniendo en cuenta que el cuadricóptero es un dispositivo de naturaleza multivariable, el diseño de control por redes neuronales tendrá dos etapas una de aprendizaje la cual se basa en estimar los parámetros de la planta y esta teoría se presenta en el siguiente capítulo, el sistema de control basado en esta dinámica es el que podemos observar a continuación. [8]

2.3.1 Control por modelo inverso

Es el primer método que encontramos para controlar un modelo multivariable no lineal usando redes neuronales, y este radica en el entrenamiento de una red para actuar como la inversa del sistema y utilizar esto como un controlador. [9]

Asumiendo que el sistema a controlar se encuentra descrito por:

$$y(k) = f_1[y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-m)] \quad (2.4)$$

Se recopila un conjunto de datos que describe cómo se comporta el sistema en todo el rango de operación:

$$\{[u(k), y(k)], k = 1, \dots, P\} \quad (2.5)$$

Utilizando el modelo de identificación por red neuronal directa descrito en el capítulo siguiente, podemos deducir un modelo de red neuronal del sistema usando este conjunto de datos.

El modelo inverso del sistema se desprende a partir del conjunto de datos:

$$\{[y(k), u(k)], k = 1, \dots, P\} \quad (2.6)$$

De esta forma la salida de control $u(k)$ del modelo inverso está dada por:

$$u(k) = f_2(y(k+1), y(k), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)) \quad (2.7)$$

Así el sistema en lazo cerrado se comporta de una manera deseada, representada por:

$$\frac{Y(z)}{R(z)} = M(z) = z^{-1} \quad (2.8)$$

Donde la salida $y(k + 1)$ es sustituida por una referencia $r(k)$. Si la red representa el modelo inverso exacto, la entrada de control producida por la red impulsará la salida del sistema en el instante $k + 1$ a $r(k)$, como se observa en la ilustración 5.

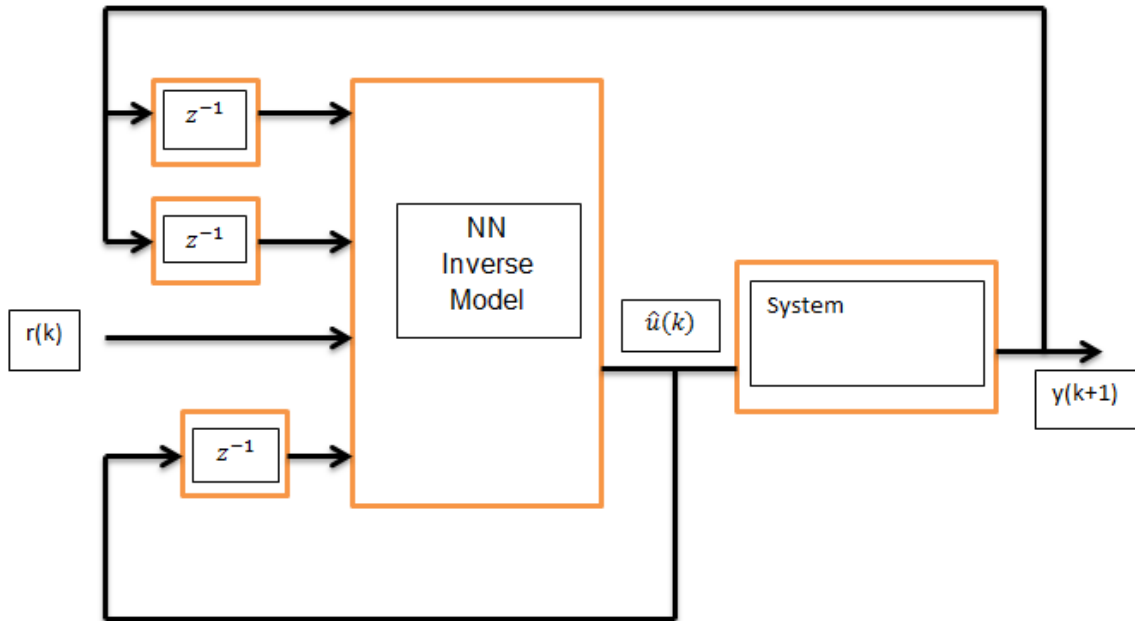


Ilustración 5 – Control por modelo inverso de un sistema. [8]

Entonces la identificación del sistema para lograr un control por modelo inverso radica en la elección de regresores de entrada y salida de la red como se muestra en la ecuación (2.7), por lo tanto la red está en capacidad de reducir al mínimo el criterio de error medio cuadrático de la forma: [9]

$$J = \frac{1}{2P} \sum_{k=1}^P [u(k) - \hat{u}(k|\theta)]^2 \quad (2.9)$$

2.3.2 Controlador con corrección por modelo inverso

Debido a que el control por modelo inverso se vuelve inestable para sistemas no lineales cuando los ceros del sistema se encuentran por fuera del círculo unitario, además que no es un controlador robusto para perturbaciones. Se utiliza la unión de dos controladores mediante la cual por el modelo inverso se mejora el seguimiento de referencia y un controlador de estructura clásica se utiliza para estabilizar el sistema y lograr disminuir las perturbaciones, la estructura del controlador se puede observar en la ilustración 6. [9]

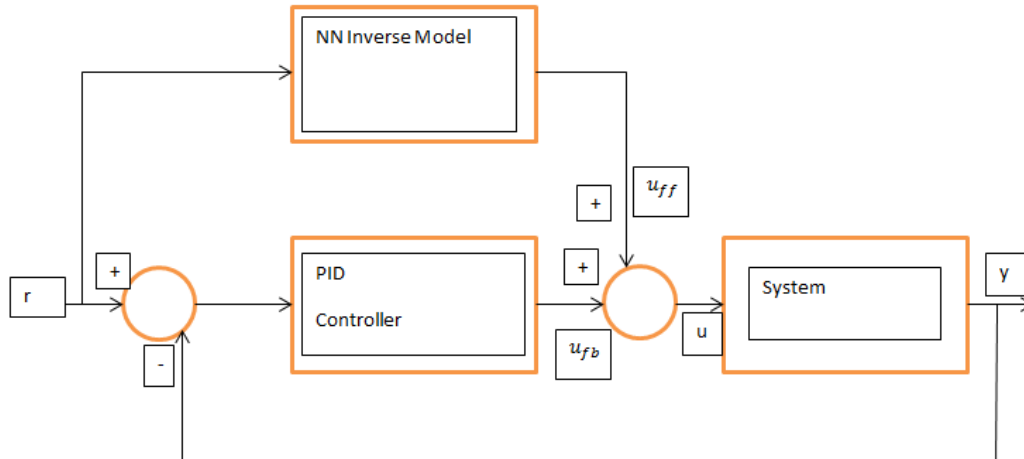


Ilustración 6 – Controlador con corrección por modelo inverso. [8]

De la anterior figura se observa la suma de dos señales de control las cuales conformaran la salida de control $u(t)$ que tendrá efecto sobre el sistema. u_{ff} representa la señal de control entregada por el modelo inverso y su representación se puede observar en la ecuación (2.7), u_{fb} representa la señal de control de un controlador de estructura clásica que para la gráfica está representado por un PID (Proporcional-Integral-Derivativo). Debido a esto el controlador posee una salida $u(t)$ de la siguiente manera: [9]

$$u(t) = u_{ff} + u_{fb} \quad (2.10)$$

Capítulo 3

3. IDENTIFICACIÓN DEL SISTEMA MIMO

En este capítulo se contextualiza sobre los modelos de regresión lineal usados para estimar la dinámica multivariable de la planta, usados en la red neuronal directa de tal modo que el proceso de aprendizaje de la red este ligado al comportamiento real del sistema.

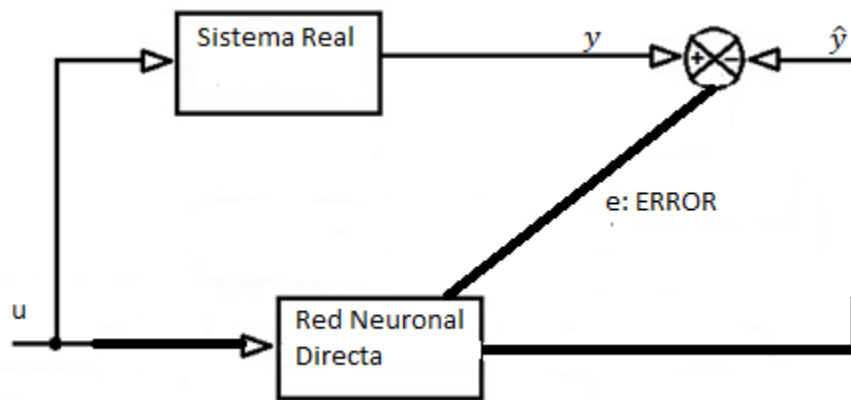


Ilustración 7 – Diagrama de bloques de aprendizaje por una red neuronal directa. Fuente: Autores.

3.1 REPRESENTACIÓN DEL SISTEMA MIMO

La representación de un sistema discreto multivariable con p salidas y l entradas con operador de retardo r puede ser de la forma [4][5]

$$A(r^{-1})y[k] = B(r^{-1})u[k] \quad (3.1)$$

Donde A está dado por

$$A(r^{-1}) = A_0 + A_1(r^{-1}) + A_2(r^{-2}) + \dots + A_n(r^{-n}) \quad (3.2)$$

Y B está dado por

$$B(r^{-1}) = B_1(r^{-1}) + B_2(r^{-2}) + \dots + B_m(r^{-m}) \quad (3.3)$$

Con $n \geq m$ y donde A_i es de dimensión $p \times p$ y B_i de dimensión $p \times l$, y donde las entradas u corresponden a un vector de dimensión $l \times 1$ y las salidas y corresponden a un vector de dimensión $p \times 1$ de la forma

$$y[k] = \begin{bmatrix} y_1[k] \\ y_2[k] \\ \vdots \\ y_p[k] \end{bmatrix}, \quad u[k] = \begin{bmatrix} u_1[k] \\ u_2[k] \\ \vdots \\ u_q[k] \end{bmatrix}$$

Si $A_0 = I$ con I siendo la matriz identidad, se obtiene

$$[I + A_1(r^{-1}) + \dots + A_n(r^{-n})] = [B_1(r^{-1}) + \dots + B_m(r^{-m})]u[k] \quad (3.4)$$

$$y[k] = B_1u[k-1] + \dots + B_mu[k-m] - A_1y[k-1] - \dots - A_ny[k-n] \quad (3.5)$$

Donde A_i y B_i son de la forma

$$A_i = \begin{bmatrix} a_{11}^i & a_{12}^i & \dots & a_{1p}^i \\ a_{21}^i & a_{22}^i & \dots & a_{2p}^i \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1}^i & a_{p2}^i & \dots & a_{pp}^i \end{bmatrix}, \quad B_i = \begin{bmatrix} b_{11}^i & b_{12}^i & \dots & b_{1l}^i \\ b_{21}^i & b_{22}^i & \dots & b_{2l}^i \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1}^i & b_{p2}^i & \dots & b_{pl}^i \end{bmatrix} \quad (3.6)$$

De las ecuaciones (3.6) y (3.4) se puede expresar la salida y_i en términos de las entradas y las salidas anteriores como

$$\begin{aligned}
y_i = & a_{i1}^1 y_1[k-1] + a_{i2}^1 y_2[k-1] + \dots + a_{ip}^1 y_p[k-1] \\
& + a_{i1}^2 y_1[k-2] + a_{i2}^2 y_2[k-2] + \dots + a_{ip}^2 y_p[k-2] \\
& + \dots \\
& + a_{i1}^n y_1[k-n] + a_{i2}^n y_2[k-n] + \dots + a_{ip}^n y_p[k-n] \\
& + b_{i1}^1 u_1[k-1] + b_{i2}^1 u_2[k-1] + \dots + b_{il}^1 u_l[k-1] \\
& + b_{i1}^2 u_1[k-2] + b_{i2}^2 u_2[k-2] + \dots + b_{il}^2 u_l[k-2] \\
& + \dots \\
& + b_{i1}^m u_1[k-m] + b_{i2}^m u_2[k-m] + \dots + b_{il}^m u_l[k-m]
\end{aligned} \tag{3.7}$$

De la ecuación (3.7) se observa que el modelo DARMA de la ecuación (3.1), es posible de expresar como:

$$y(k) = \theta^T \phi[K-1]; \quad K \geq 0 \tag{3.8}$$

En donde la matriz θ^T denota la transpuesta de, θ siendo esta una matriz conformada por los parámetros de las matrices $A(q^{-1})$ y $B(q^{-1})$ por lo tanto el tamaño de θ es de $(l * m + p * n) \times p$, esto es:

$$\theta^T = \begin{bmatrix} b_{11}^i & \dots & b_{11}^m & \dots & b_{1l}^1 & \dots & b_{1l}^m & -a_{11}^1 & \dots & -a_{11}^n & \dots & -a_{1p}^1 & \dots & -a_{1p}^n \\ b_{21}^1 & \dots & b_{21}^m & \dots & b_{2l}^1 & \dots & b_{2l}^m & -a_{21}^1 & \dots & -a_{21}^n & \dots & -a_{2p}^1 & \dots & -a_{2p}^n \\ \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots & \vdots & \dots \\ b_{p1}^i & \dots & b_{p1}^m & \dots & b_{pl}^1 & \dots & b_{pl}^m & -a_{p1}^1 & \dots & -a_{p1}^n & \dots & -a_{pp}^1 & \dots & -a_{pp}^n \end{bmatrix} \tag{3.9}$$

$\phi[K-1]$ Es un vector de dimensión $(l * m + p * n) \times 1$ y contiene los valores de las entradas y las salidas anteriores:

$$\phi[K - 1] = \begin{bmatrix} y_1[k - 1] \\ \vdots \\ y_p[k - 1] \\ \vdots \\ y_1[k - n] \\ \vdots \\ y_p[k - n] \\ \vdots \\ u_1[k - 1] \\ \vdots \\ u_l[k - 1] \\ \vdots \\ u_1[k - m] \\ \vdots \\ u_l[k - m] \end{bmatrix} \quad (3.10)$$

3.2 ESQUEMAS DE IDENTIFICACIÓN EN LÍNEA

La finalidad de la estimación en línea es encontrar la dinámica del sistema basándose en los valores anteriores de las entradas y salidas, la matriz de parámetros estimados $\hat{\theta}[k]$ también es calculada a partir de la matriz anterior de parámetros estimados $\hat{\theta}[k - 1]$, como se muestra en la siguiente ecuación: [5]

$$\hat{\theta}[k] = \hat{\theta}[k - 1] + \mathbf{M}[k - 1]\phi[k - 1]\mathbf{e}[k] \quad (3.11)$$

Donde $\mathbf{M}[k - 1]$ denota la ganancia del algoritmo (posiblemente es una matriz) y $\mathbf{e}[k]$ es un vector de los errores de estimación de las salidas, y es de la forma

$$\mathbf{e}[k] = \mathbf{y}^T[k] - \hat{\mathbf{y}}^T[k] \quad (3.12)$$

El vector $\hat{\mathbf{y}}[k]$ se calcula a partir de la matriz de parámetros obtenidos en la identificación anterior y el vector de entradas y salidas pasadas

$$\hat{\mathbf{y}}[k] = \hat{\theta}[k - 1]^T \phi[k - 1] \quad (3.13)$$

La ecuación (3.11) muestra un algoritmo que permite calcular los parámetros del sistema en cada nuevo instante de muestreo, es decir, es posible implementarlo en línea.

La ganancia de la ecuación (3.11) puede ser motivada por diferentes funciones objetivo, en las siguientes secciones mostraremos varios algoritmos que se desprenden de esta ecuación.

3.3 MÍNIMOS CUADRADOS

El método de mínimos cuadrados puede ser aplicado a una gran variedad de problemas. Este es particularmente simple si el modelo matemático puede ser escrito en la siguiente forma: [5]

$$\mathbf{y}[i] = \varphi_1[i]\theta_1^0 + \varphi_2[i]\theta_2^0 + \dots + \varphi_n[i]\theta_n^0 = \boldsymbol{\varphi}^T[i]\boldsymbol{\theta}^0 \quad (3.14)$$

Donde \mathbf{y} es la variable observada $\theta_1^0, \theta_2^0, \dots, \theta_n^0$ son los parámetros del modelo a determinar, y $\varphi_1, \varphi_2, \dots, \varphi_n$ son funciones conocidas que pueden depender de otras variables conocidas.

El modelo es indexado por la variable i , esta denota tiempo o número de la muestra. Las variables φ_i son llamadas variables de regresión, y el modelo descrito en la ecuación (3.14) es también llamado modelo de regresión.

Si extendemos este modelo a la representación de sistemas descrito en la sección anterior, tenemos:

$$y[i] = -A_1[i]y[i-1] - \dots - A_n[i]y[i-n] + B_1[i]u[i-1] + \dots + B_m[i]u[i-m] = \boldsymbol{\theta}^T \boldsymbol{\phi}[i-1] \quad (3.15)$$

Donde

$$\boldsymbol{\theta}[i]^T = [A_1 \ A_2 \ \dots \ A_n \ B_1 \ B_2 \ \dots \ B_m]$$

$$\boldsymbol{\phi}[i-1] = \begin{bmatrix} -y[i-1] \\ -y[i-2] \\ \vdots \\ -y[i-n] \\ u[i-1] \\ u[i-2] \\ \vdots \\ u[i-m] \end{bmatrix}$$

Los pares de observaciones y variables de $\{(\mathbf{y}[i], \boldsymbol{\phi}[i - 1]), i = 1, 2, \dots, k\}$ pueden ser obtenidos de forma experimental. El problema es entonces determinar los parámetros de tal manera que las salidas calculadas a partir del modelo de la ecuación (3.15) sean tan iguales como sea posible con respecto a las variables medidas $y[i]$ en el sentido de los mínimos cuadrados. Es decir, los parámetros $\boldsymbol{\theta}$ deben ser elegidos de tal manera que minimice la función de costo. [5]

$$V(\boldsymbol{\theta}, i) = \frac{1}{2} \sum_{i=n+1}^k (\mathbf{y}[i] - \boldsymbol{\theta}^T \boldsymbol{\phi}[i - 1])^2 \quad (3.16)$$

Ya que las medidas de la variable \mathbf{y} son lineales en los parámetros $\boldsymbol{\theta}$ y el criterio de los mínimos cuadrados es cuadrático, el problema admite una solución analítica. Introduciendo la notación y con $n = m$

$$\mathbf{Y}[k] = \begin{bmatrix} y[n+1]^T \\ y[n+2]^T \\ \vdots \\ y[k]^T \end{bmatrix}, \quad \mathbf{E}[k] = \begin{bmatrix} e[n+1]^T \\ e[n+2]^T \\ \vdots \\ e[k]^T \end{bmatrix}$$

$$\boldsymbol{\Phi}[k] = \begin{bmatrix} -y(n)^T & -y(n-1)^T & \dots & -y(1)^T & u(n)^T & u(n-1)^T & \dots & -u(1)^T \\ -y[n+1]^T & -y(n)^T & \dots & -y(2)^T & u[n+1]^T & u(n)^T & \dots & -u(2)^T \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ -y[k-1]^T & -y[k-2]^T & \dots & -y[k-n]^T & u[k-1]^T & u[k-2]^T & \dots & -u(k-n)^T \end{bmatrix}$$

$$\boldsymbol{\Phi}[k] = \begin{bmatrix} \boldsymbol{\phi}(n)^T \\ \boldsymbol{\phi}[n+1]^T \\ \vdots \\ \boldsymbol{\phi}[k-1]^T \end{bmatrix}$$

$$P[k] = (\boldsymbol{\Phi}[k]^T \boldsymbol{\Phi}[k])^{-1} = (\sum_{i=n+1}^k \boldsymbol{\phi}[i-1] \boldsymbol{\phi}[i-1]^T)^{-1} \quad (3.17)$$

Donde los errores $e[i]$ son definidos por

$$e[i] = \mathbf{y}[i] - \hat{\mathbf{y}}[i] = \mathbf{y}[i] - \boldsymbol{\theta}^T \boldsymbol{\phi}[i - 1]$$

Con estas notaciones la función de costo (3.16) puede ser reescrita como

$$V(\boldsymbol{\theta}, i) = \frac{1}{2} \sum_{i=n+1}^k e[i]^2 = \frac{1}{2} \mathbf{E}^T \mathbf{E} = \frac{1}{2} \|\mathbf{E}\|^2 \quad (3.18)$$

Donde \mathbf{E} puede ser escrito como

$$\mathbf{E} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \Phi\boldsymbol{\theta}$$

La solución al problema de mínimos cuadrados está dada por el siguiente teorema.

Teorema 1.3.1 La función de la ecuación (3.16) es mínima para parámetros $\hat{\boldsymbol{\theta}}$ tal que

$$\Phi^T \Phi \hat{\boldsymbol{\theta}} = \Phi^T \mathbf{Y} \quad (3.19)$$

Si la matriz $\Phi^T \Phi$ es no singular, el mínimo es único y está dado por

$$\hat{\boldsymbol{\theta}} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{Y} \quad (3.20)$$

3.3.1 Forma Recursiva

Cuando se implementa un control que presenta la necesidad de recalcularse en cada iteración, las muestras son obtenidas de forma secuencial y en tiempo real. En este sentido, con el fin de ahorrar tiempo de computación y de realizar implementaciones en línea es necesario tener todos los algoritmos en ecuaciones de forma recursiva. La ecuación de estimación de mínimos cuadrados puede ser cambiada de tal forma que el resultado obtenido en $k - 1$ pueda ser utilizado para obtener la estimación en el tiempo k . Si $\hat{\boldsymbol{\theta}}[k - 1]$ denota la estimación por mínimos cuadrados basada en $k - 1$ mediciones. [5]

Asumiendo que la matriz $\Phi^T \Phi$ es no singular para todo k . Se deduce de la definición de $\mathbf{P}[k]$ en la ecuación (3.17) que [6][7]

$$\mathbf{P}^{-1}[K] = \Phi^T[k] \Phi[k] = \sum_{i=n+1}^k \phi[i-1] \phi[i-1]^T$$

$$\begin{aligned}
&= \sum_{i=n+1}^{k-1} \phi[i-1]\phi[i-1]^T + \phi[k-1]\phi[k-1]^T \\
&= \mathbf{P}^{-1}[K-1] + \phi[k-1]\phi[k-1]^T \quad (3.21)
\end{aligned}$$

Los parámetros estimados $\hat{\theta}[k]$ es dado por la ecuación (3.17) y puede ser escrita como

$$\begin{aligned}
\hat{\theta}[k] &= (\sum_{i=n+1}^k \phi[i-1]\phi[i-1]^T)^{-1} (\sum_{i=n+1}^k \phi[i-1]y[i]^T) = \mathbf{P}[k] (\sum_{i=n+1}^k \phi[i-1]y[i]^T) \\
&= \mathbf{P}[k] (\sum_{i=n+1}^{k-1} \phi[i-1]y[i]^T + \phi[k-1]y[k]^T) \quad (3.22)
\end{aligned}$$

De las ecuaciones (3.21) y (3.22), tenemos

$$\begin{aligned}
\sum_{i=n+1}^{k-1} \phi[i-1]y[i]^T &= \mathbf{P}^{-1}[K-1] + \hat{\theta}[k-1] = \mathbf{P}^{-1}[K]\hat{\theta}[k-1] - \phi[k-1]\phi[k-1]^T \hat{\theta}[k-1] \\
&= \mathbf{P}^{-1}[K]\hat{\theta}[k-1] - \phi[k-1]\phi[k-1]^T \hat{\theta}[k-1] \quad (3.23)
\end{aligned}$$

La estimación en el tiempo k es entonces

$$\begin{aligned}
\hat{\theta}[k] &= \hat{\theta}[k-1] - \mathbf{P}[k]\phi[k-1]\phi[k-1]^T \hat{\theta}[k-1] + \mathbf{P}[k]\phi[k-1]y[k]^T = \\
&= \hat{\theta}[k-1] - \mathbf{P}[k]\phi[k-1] [y[k]^T - \phi[k-1]^T \hat{\theta}[k-1]] = \hat{\theta}[k-1] + \mathbf{K}[k]e[k]^T \quad (3.24)
\end{aligned}$$

Donde

$$\mathbf{K}[k] = \mathbf{P}[k]\phi[k-1] \quad y \quad e[k]^T = y[k]^T - \phi[k-1]^T \hat{\theta}[k-1]$$

Para continuar, es necesario encontrar una ecuación recursiva para $\mathbf{P}[k]$ o más bien de su inversa (3.21); para ello se utiliza el siguiente lema.

Teorema 1.3.1.1 Sean \mathbf{A}, \mathbf{C} y $\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B}$ matrices cuadradas no singulares. Entonces $\mathbf{A} + \mathbf{BCD}$ es invertible, y

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}[\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B}]^{-1}\mathbf{D}\mathbf{A}^{-1}$$

Aplicando el lema 1.3.1.1 a $\mathbf{P}[k]$ y usando la ecuación (3.21), tenemos

$$\mathbf{P}[k] = (\mathbf{P}[k-1]^T + \phi[k-1]\phi[k-1]^T)^{-1} = \mathbf{P}[k-1] - \mathbf{P}[k-1]\phi[k-1][\mathbf{I} + \phi[k-1]^T\mathbf{P}[k-1]\phi[k-1]]^{-1}\phi[k-1]^T\mathbf{P}[k-1] \quad (3.25)$$

Lo que implica que

$$\mathbf{K}[k] = \mathbf{P}[k]\phi[k-1] = \mathbf{P}[k-1]\phi[k-1][\mathbf{I} + \phi[k-1]^T\mathbf{P}[k-1]\phi[k-1]]^{-1} \quad (3.26)$$

Y, por consiguiente

$$\dot{\boldsymbol{\theta}}[k] = \dot{\boldsymbol{\theta}}[k-1] + \mathbf{P}[k-1]\phi[k-1][\mathbf{I} + \phi[k-1]^T\mathbf{P}[k-1]\phi[k-1]]^{-1}(\mathbf{y}[k]^T - \phi[k-1]^T\dot{\boldsymbol{\theta}}[k-1]) \quad (3.27)$$

$$\mathbf{P}[k] = \mathbf{P}[k-1] - \mathbf{P}[k-1]\phi[k-1][\mathbf{I} + \phi[k-1]^T\mathbf{P}[k-1]\phi[k-1]]^{-1}\phi[k-1]^T\mathbf{P}[k-1] \quad (3.28)$$

Las ecuaciones anteriores resultan de asumir que la matriz $\Phi[k]$ es de rango completo, esto es $\Phi[k]^T\Phi[k]$ es no singular para todo $k \geq k_0$.

Ahora, revisemos el término $\phi[k-1]^T\mathbf{P}[k-1]\phi[k-1]$, las dimensiones de $\phi[k-1]$ es un vector de $(p * n + l * m) \times 1$, $\mathbf{P}[k-1]$ es una matriz de $(p * n + l * m) \times (p * n + l * m)$, por consiguiente, el término $\phi[k-1]^T\mathbf{P}[k-1]\phi[k-1]$ es un escalar; lo que implica que las ecuaciones (3.27) y (3.28) pueden ser escritas como

$$\dot{\boldsymbol{\theta}}[k] = \dot{\boldsymbol{\theta}}[k-1] + \frac{\mathbf{P}[k-1]\phi[k-1]}{1 + \phi[k-1]^T\mathbf{P}[k-1]\phi[k-1]}(\mathbf{y}[k]^T - \phi[k-1]^T\dot{\boldsymbol{\theta}}[k-1]) \quad (3.29)$$

$$\mathbf{P}[k] = \mathbf{P}[k-1] - \frac{\mathbf{P}[k-1]\phi[k-1]\phi[k-1]^T\mathbf{P}[k-1]}{1 + \phi[k-1]^T\mathbf{P}[k-1]\phi[k-1]} \quad (3.30)$$

3.4 MÍNIMOS CUADRADOS CON FACTOR DE OLVIDO

La ecuación final para el algoritmo es [5]

$$\dot{\theta}[k] = \dot{\theta}[k-1] + \frac{P[k-1]\phi[k-1]}{\alpha[k-1] + \phi[k-1]^T P[k-1] \phi[k-1]} (y[k]^T - \phi[k-1]^T \dot{\theta}[k-1]) \quad (3.31)$$

Donde

$$P[k] = \frac{1}{\alpha[k-1]} \left(P[k-1] - \frac{P[k-1]\phi[k-1]\phi[k-1]^T P[k-1]}{\alpha[k-1] + \phi[k-1]^T P[k-1] \phi[k-1]} \right) \quad (3.32)$$

$$\alpha[k] = \alpha_0 \alpha[k-1] + 1 - \alpha_0 \quad (3.33)$$

Normalmente se escoge $\alpha_0 = 0.99$ y $\alpha[0] = 0.95$.

3.5 PROYECCIÓN

El algoritmo de proyección resulta de encontrar $\dot{\theta}[k]$ a partir de los datos conocidos $\dot{\theta}[k-1]$ y $y[k]$ que minimizan la función de costo [5]

$$J = \frac{1}{2} \|\dot{\theta}[k] - \dot{\theta}[k-1]\|^2 \quad (3.34)$$

El problema está sujeto a

$$y[k]^T = \phi[k-1]^T \dot{\theta}[k] \quad (3.35)$$

Para encontrar la solución a este problema de optimización se introducen los multiplicadores de Lagrange, de la ecuación (3.34) y (3.35) se tiene

$$J_c = \frac{1}{2} \|\dot{\theta}[k] - \dot{\theta}[k-1]\|^2 + \lambda [y[k]^T - \phi[k-1]^T \dot{\theta}[k]] \quad (3.36)$$

El mínimo se encuentra cuando las derivadas parciales de ecuación (3.36) son igualadas a cero, esto es

$$\frac{\partial J_c}{\partial \dot{\theta}} = 0, \quad \frac{\partial J_c}{\partial \lambda} = 0 \quad (3.37)$$

De donde se obtiene:

$$\dot{\theta}[k] - \dot{\theta}[k-1] - \lambda \phi[k-1] = 0 \quad (3.38)$$

$$\mathbf{y}[k]^T - \phi[k-1]^T \dot{\theta}[k] = 0 \quad (3.39)$$

Reemplazando (3.38) en (3.39)

$$\mathbf{y}[k]^T - \phi[k-1]^T [\dot{\theta}[k-1] + \lambda \phi[k-1]] = 0 \quad (3.40)$$

Resolviendo (3.40) para λ , se tiene

$$\lambda = \frac{\mathbf{y}[k]^T - \phi[k-1]^T \dot{\theta}[k-1]}{\phi[k-1]^T \phi[k-1]} \quad (3.41)$$

Nuevamente, sustituyendo (3.41) en la ecuación (3.38) podemos encontrar la solución del problema de optimización y con ello una ecuación recurrente

$$\dot{\theta}[k] = \dot{\theta}[k-1] + \frac{\phi[k-1]}{\phi[k-1]^T \phi[k-1]} [\mathbf{y}[k]^T - \phi[k-1]^T \dot{\theta}[k-1]] \quad (3.42)$$

Para evitar la singularidad, se adiciona una pequeña cantidad c al denominador del algoritmo.

3.6 PROYECCIÓN ORTOGONAL

La ecuación final para el algoritmo de proyección ortogonal está dada por: [5]

$$\dot{\theta}[k] = \dot{\theta}[k-1] + \frac{P[k-1]\phi[k-1]}{\phi[k-1]^T P[k-1]\phi[k-1]} [\mathbf{y}[k]^T - \phi[k-1]^T \dot{\theta}[k-1]] \quad (3.43)$$

Donde \mathbf{P} es una matriz con dimensiones $(p * n + l * m) \times (p * n + l * m)$, que se actualiza de la forma

$$\mathbf{P}[k] = \mathbf{P}[k - 1] + \frac{\mathbf{P}[k-1]\phi[k-1]\phi[k-1]^T\mathbf{P}[k-1]}{\phi[k-1]^T\mathbf{P}[k-1]\phi[k-1]} \left[\mathbf{y}[k]^T - \phi[k - 1]^T \hat{\boldsymbol{\theta}}[k - 1] \right] \quad (3.44)$$

Con un estimado inicial $\hat{\boldsymbol{\theta}}$ dado y con $\mathbf{P}(0) = \mathbf{I}$.

Al igual que con el algoritmo de proyección, para evitar una posible división por cero, se le adiciona una pequeña cantidad c al denominador del algoritmo. En algunas ocasiones, si el denominador es cero se opta por conservar los valores anteriores, es decir, $\mathbf{P}[k] = \mathbf{P}[k - 1]$ y $\hat{\boldsymbol{\theta}}[k] = \hat{\boldsymbol{\theta}}[k - 1]$

Capítulo 4

4. Análisis y resultados

En este capítulo presentamos los resultados obtenidos bajo un tiempo de muestreo de 60 milisegundos, y a continuación presentamos el cuadricóptero JJR/C para capturar los datos en tiempo real. [11]



Ilustración 8 – Cuadricóptero JJR/C. Fuente: Autores.

El dispositivo presentado en la anterior ilustración nos entregó la posición de los movimientos Pitch, Yaw y Roll a través de los sensores que posee en su tarjeta de programación, y teniendo esta información se necesitan los valores enviados a dicho sistema para poder realizar la estimación multivariable de la planta.

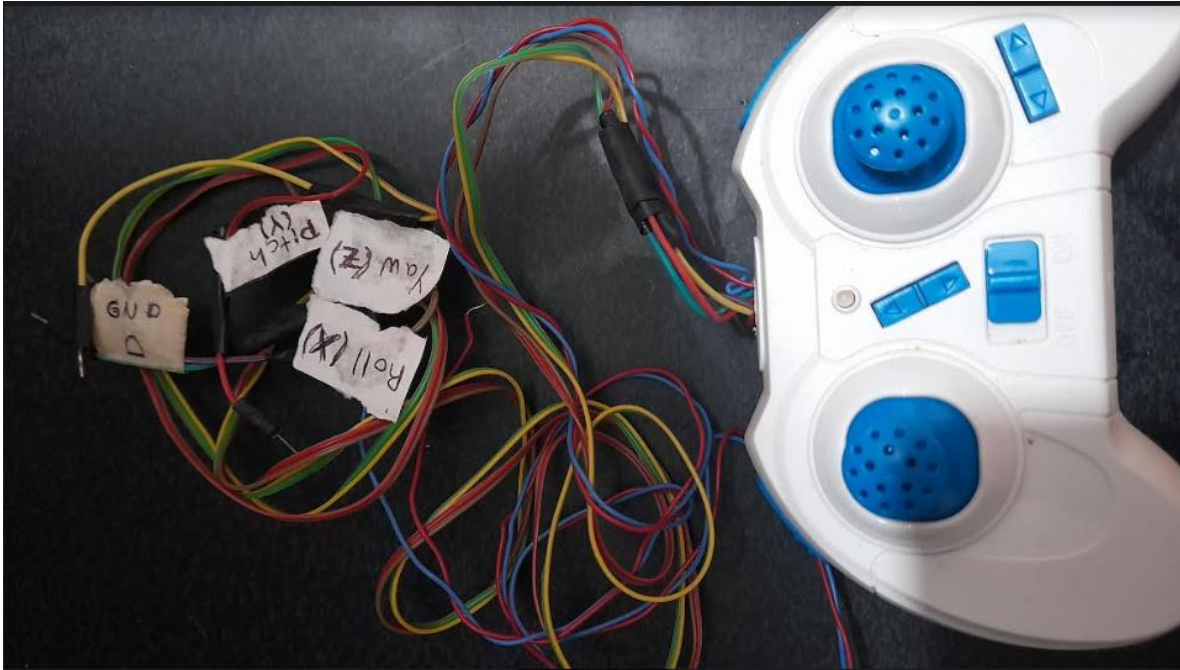


Ilustración 9 – Control remoto del cuadricóptero. Fuente: Autores.

En la ilustración 9 observamos el control remoto del cuadricóptero de donde obtenemos los datos de las entradas del dispositivo, y así poder realizar la teoría de estimación para el aprendizaje de la red neuronal.

4.1 Aprendizaje de la red neuronal

Para diseñar la red neuronal directa mediante la cual se realiza el aprendizaje del sistema de control primero se necesita conocer una aproximación de la dinámica de la planta, por lo que para esto realizamos una estimación fuera de línea.

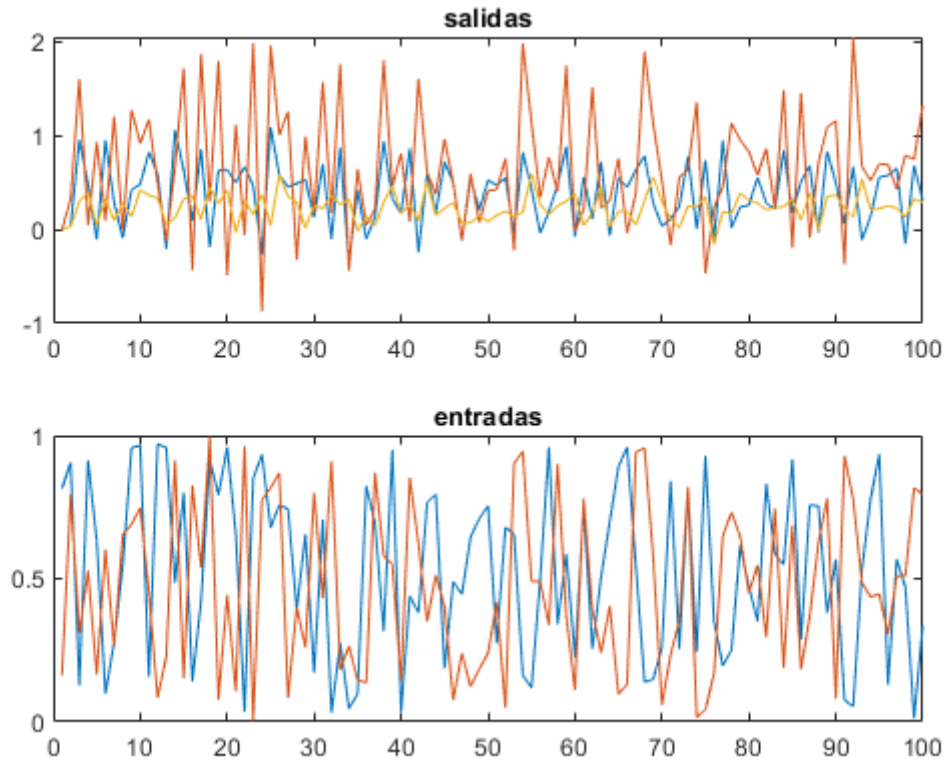


Ilustración 10 – Datos obtenidos de la planta real. Fuente: Autores.

Durante un funcionamiento de 5 minutos donde el dispositivo real sobrevoló por un espacio controlado de perturbaciones externas fuertes como viento, lluvia, etc, se capturaron los datos de entradas y salidas para obtener la dinámica inicial de la planta con la cual se procede a realizar la etapa inicial de aprendizaje de la red.

A partir de este conocimiento inicial realizamos un aprendizaje de la red neuronal a partir de un método de identificación recursiva, donde los valores serán recalculados durante cada iteración y por lo tanto en este caso si aparece alguna perturbación externa los parámetros serán recalculados teniendo en cuenta todo lo que puede alterar la dinámica de la planta

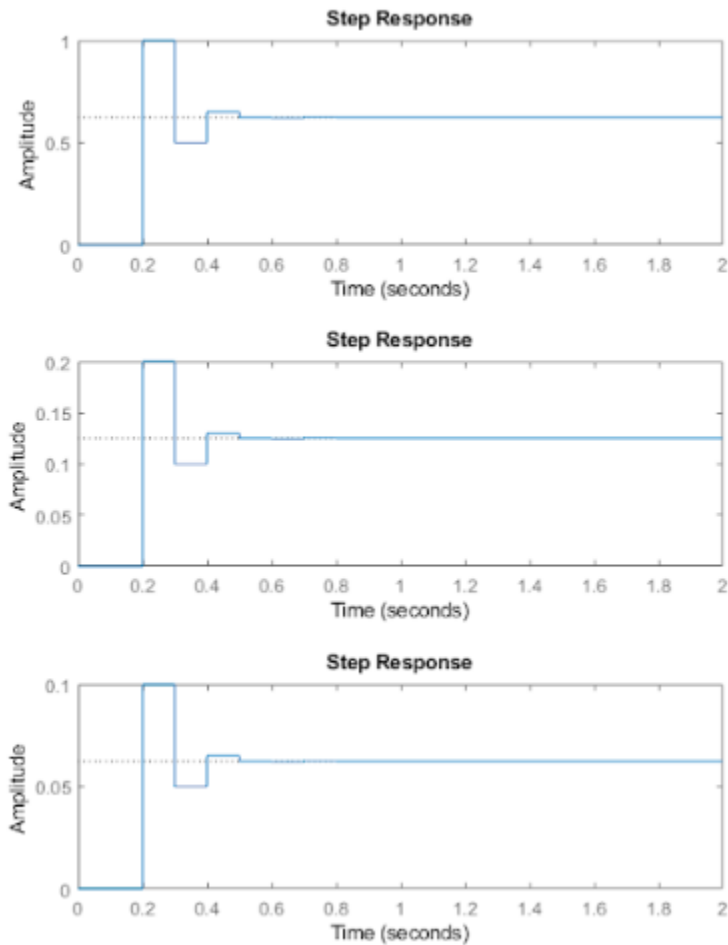


Ilustración 11 – Aprendizaje de la red usando Mínimos cuadrados con factor de olvido. Fuente: Autores.

En la anterior ilustración podemos observar tres graficas en las cuales tenemos las respuestas del cuadricóptero, podemos definir la grafica superior como la salida Pitch, La grafica del medio la salida Yaw y por último la grafica inferior la salida Roll. Podemos definir que el aprendizaje de la red nos esta entregando un sistema multivariable pero con dinámica muy estable.

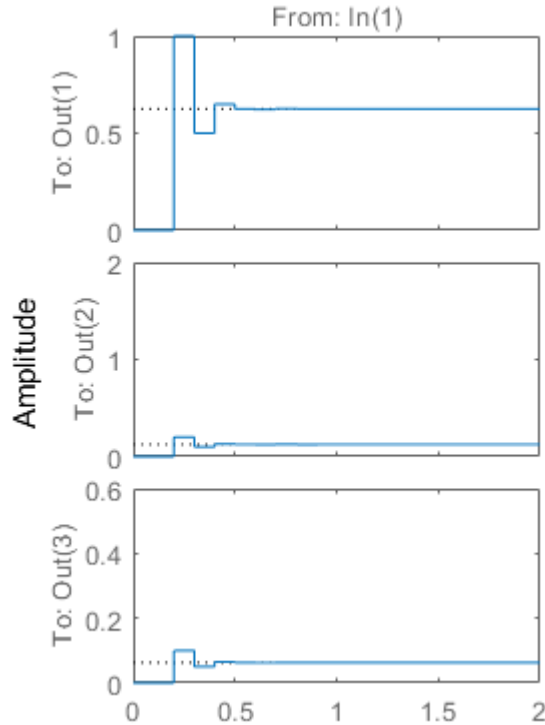


Ilustración 12 – Aprendizaje de la red usando proyección ortogonal. Fuente: Autores.

Al igual que en la ilustración 11, poseemos nuestras tres salidas en el mismo orden, y usando este algoritmo recursivo podemos determinar casi la misma dinámica de la planta por lo que se puede determinar que los algoritmos recursivos en la etapa de aprendizaje de la red funcionan de forma eficiente.

Con lo anteriormente mencionado encontramos que el sistema multivariable, corresponde a un sistema de 3 entradas y 3 salidas las cuales son Pitch (θ), Roll (ϕ) y Yaw (ω). Los valores estimados del sistema MIMO conforman una matriz de transferencia la cual posee toda la dinámica de planta, como se observa a continuación:

$$\begin{bmatrix} \omega \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \frac{-2.746z^2 + 2.476z + 0.296}{z^3 - 2.054z^2 + 1.43z - 0.4275} & \frac{0.1836z^2 - 1.99z + 0.985}{z^3 - 2.054z^2 + 1.43z - 0.4275} & \frac{0.2901z^2 - 0.1932z - 0.0742}{z^3 - 2.054z^2 + 1.43z - 0.4275} \\ \frac{0.1935z^2 + 0.5094z - 0.6593}{z^3 - 2.073z^2 + 1.52z - 0.4673} & \frac{3.991z^2 - 2.412z - 1.421}{z^3 - 2.073z^2 + 1.52z - 0.4673} & \frac{-0.1042z^2 + 0.0139z - 0.05021}{z^3 - 2.073z^2 + 1.52z - 0.4673} \\ \frac{-0.063z^2 + 0.2352z - 0.5089}{z^3 - 2.72z^2 + 1.497z - 0.3253} & \frac{-0.55z^2 + 0.2261z - 0.2901}{z^3 - 2.72z^2 + 1.497z - 0.3253} & \frac{0.6268z^2 + 1.5z - 0.874}{z^3 - 2.72z^2 + 1.497z - 0.3253} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

4.2 Control con corrección por modelo inverso

Teniendo la información de la planta presentada en la sección anterior se procede a realizar la señal de control por modelo inverso como se presenta en el capítulo 2, y sabiendo que el modelo de aprendizaje es lento por lo tanto se determinó un

PID a partir del Auto tune de Matlab sobre la diagonal principal de la matriz de transferencia.

Y así se entrega el control por redes neuronales sobre nuestro sistema en un ambiente simulado como Simulink.

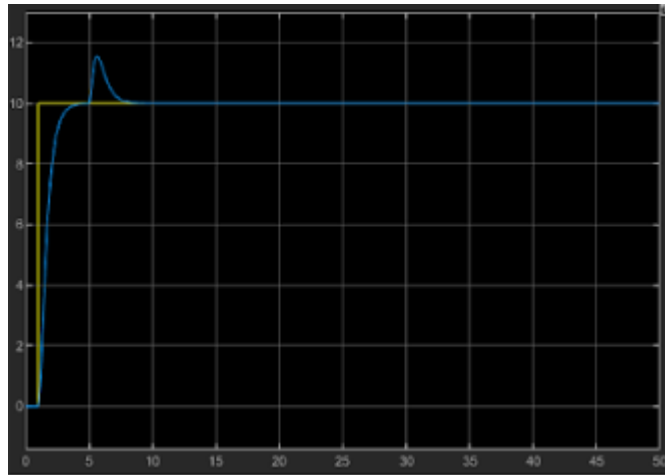


Ilustración 13 – Salida controlada del movimiento Pitch. Fuente: Autores.

La ilustración 13, nos presenta la respuesta controlada en la cual se aplica un escalón como entrada y se observa que logra llegar a la referencia en un tiempo de establecimiento rápido y que aunque presenta un máximo sobrepaso logra corregirlo y entregar un error en estado estacionario de cero. El máximo sobre paso que podemos observar del sistema es un acople interno robusto que posee la dinámica de la planta como se observara a continuación.

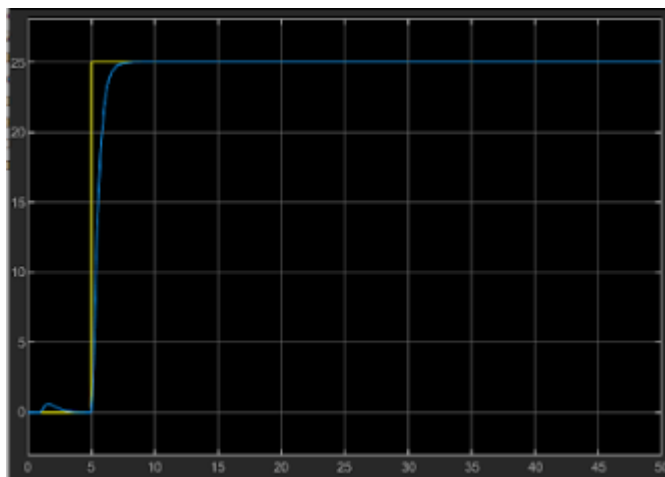


Ilustración 14 – Salida controlada del movimiento Yaw. Fuente: Autores.

En la ilustración 14 también aplicamos un escalón con diferente amplitud sobre la salida Yaw controlada, en este caso empezamos a funcionar en el segundo 5 y se determina que el tiempo de respuesta es mucho más rápido que en el caso de la salida Pitch esto se debe a que el acople que interno perturba de una forma mas agresiva a la salida presentada en la ilustración 13. Se puede determinar que la eficiencia del controlador sobre esta salida es muy alta dado que no presentamos máximos sobrepasos y que responde al cambio de referencia de una forma casi instantánea.

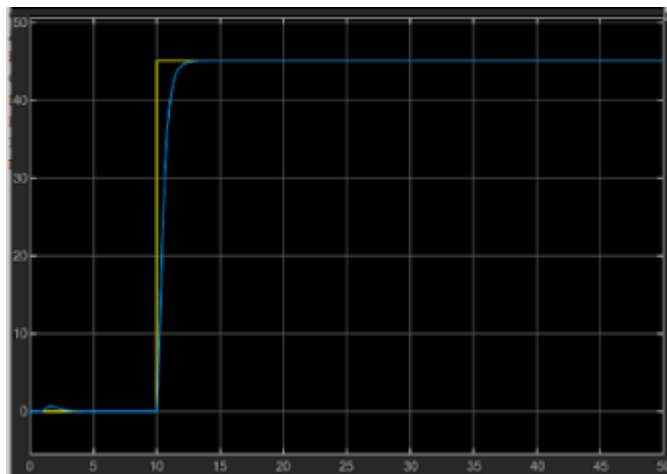


Ilustración 15 – Salida controlada del movimiento Roll. Fuente: Autores.

Para terminar la ilustración 15 nos presenta la ultima salida controlada, donde se observa que la salida Roll presenta un comportamiento similar al presentado en la ilustración 14. Tenemos un tiempo de inicio de 10 segundo y la respuesta es casi instantanea por lo que se acepta el tiempo de respuesta y no posee maximo sobre paso de forma que los acoples internos presentados en la dinamica no son tan fuertes con dicha salida.

Se tiene que el controlador por redes neuronales funciona de forma eficiente sobre el sistema multivariable, y se observo que logra llevar las tres salidas de la planta a los cambios de referencia deseados sim importar que tan fuerte pueda ser la dinámica encontrada del sistema.

CONCLUSIONES

- Se debe tener en cuenta que para realizar la estimación fuera de línea entre más valores guarde de las salidas y las entradas, el modelo determinado será mucho más cercano a la dinámica real. De esta forma se deben tener tiempos de muestreo bajos y realizar un programa en el cual podamos guardar en vectores todos los valores entregados durante todo el tiempo de funcionamiento.
- El sistema de regresión lineal usado para este proyecto necesita siempre los datos actuales y anteriores de la planta real, pero se debe tener en cuenta que cada que ponemos en funcionamiento el cuadricóptero encontramos nuevos valores los cuales tienen que ser usados para un nuevo modelo y no combinarlos con datos encontrados en un funcionamiento anterior.
- Los diferentes algoritmos de identificación recursiva utilizados para el aprendizaje de la red presentaron una respuesta satisfactoria, y podemos decir que sin importar cual sea el algoritmo utilizado encontramos una dinámica de la planta muy cercana a la real.
- Un sistema tipo cuadricóptero, aunque presenta una complejidad por los acoples internos se pudo observar que usando un controlador por redes neuronales se puede controlar y llevar a referencias deseadas, sin importa cuál es la amplitud de la referencia o que tan fuertes son los acoples el sistema multivariable respondió en tiempos de establecimiento rápidos y con errores de estado estacionario de cero.

BIBLIOGRAFIA

- [1] Control de Sistemas Multivariables. {En línea}. {1 de agosto del 2022}. Disponible en: (http://fjartnmusic.com/Personal/UIA_files/mimo.pdf)
- [2] MOVABLE TYPE LTDA, Calculate distance, bearing and more between Latitude/Longitude points [Online]; Fecha de revisión (Septiembre 2022) disponible en: (<http://movable-type.co.uk/scripts/latlong.html>)
- [3] Giraldo Buitrago, Didier. Giraldo Suarez, Eduardo. Teoría de control análogo. Primera edición. Pereira, Colombia: Universidad Tecnológica de Pereira, 2010. 299 p.
- [4] Aström, Karl J and Wittenmark, Björn. Adaptive Control. Second edition. Mineloa, New York: Dover Publications inc, 2008. 590 p.
- [5] Graham C Goodwin and Kwai Sang Sin. Adaptive filtering prediction and control. Courier Dover Publications, 2013.
- [6] Control de Sistemas Multivariables. {En línea}. {1 de agosto del 2022}. Disponible en: (http://fjartnmusic.com/Personal/UIA_files/mimo.pdf)
- [7] Alberto Aguado. Algoritmo de control predictivo-adaptable en el espacio pseudo estados. RIAII, 3(1):52-60, 2010.
- [8] Marin López, Angelo. Control multivariable no lineal de estructura variable aplicado a un vehículo aéreo no tripulado. Pereira, Colombia. 2017. Presentado en la Universidad Tecnológica de Pereira para optar al título de magister en ingeniería eléctrica.
- [9] Gopal, M. Digital Control and State Variable Methods. Second edition. India: Tata McGraw-Hill Publishing Company Limited, 2003. 984 p.
- [10] Giraldo Buitrago, Didier. Giraldo Suarez, Eduardo. Teoría de control digital. Primera edición. Pereira, Colombia: produmedios, 2012. 262 p.
- [11] Drone JJCH20. {En línea}. {4 de julio del 2022}. Disponible en: (<https://www.dronesbaratosya.com/jjrc-h20-un-hexacoptero-para-principiantes-barato/>).