

Control basado en eventos mediante umbral relativo aplicado al control de altitud de cuadricópteros Crazyflie 2.1

Francisco José Mañas-Álvarez, María Guinaldo, Raquel Dormido,
Rafael Socas, Sebastián Dormido

Departamento de Informática y Automática

E.T.S.I. Informática, Universidad Nacional de Educación a Distancia, España,
(fjmanas, mguinaldo, raquel, sdormido)@dia.uned.es, rsocas@telefonica.net

Resumen

En este trabajo se presenta el diseño de un controlador basado en eventos para el control de altitud del Crazyflie 2.1. La novedad estriba en la implementación del generador de eventos mediante un umbral relativo. La adaptación de este umbral se realiza en función de la distancia a la referencia y la presencia de ruido en el sistema. Los resultados obtenidos muestran una reducción del 93% de actualizaciones de la señal de control requeridas frente a controladores basados en eventos con umbral fijo o controladores de tipo periódico.

Palabras clave: Controlador PID, muestreo adaptativo, nano-dron.

1 Introducción

Los nano-drones se han convertido en elemento fundamental para la investigación de cuadricópteros [21, 15]. Entre las ventajas que aportan respecto a otras plataformas destacan su bajo coste en comparación con otros vehículos aéreos, su bajo consumo de energía o su rápida maniobrabilidad. Los desarrollos conseguidos en su capacidad de maniobra plantean múltiples desafíos que abarcan desde el modelado dinámico, la estimación de estados o el seguimiento de trayectorias y el control.

El Crazyflie 2.1 es una plataforma comercial desarrollada por la compañía Bitcraze [2]. Se trata de un nano-dron de código y hardware abierto muy utilizado en distintas aplicaciones. Por ejemplo, en problemas de grandes enjambres [19, 18], en planificadores de rutas [13], en educación [7], en rechazo a perturbaciones [4], etc. Las características que presenta este nano-dron, como las no linealidades o el ser un sistema subactuado, lo convierten en una plataforma muy interesante para la investigación en diseño de sistemas de control para seguimiento de trayectorias. Además, el hecho de que sean vehículos no tripulados sugieren investigar en conseguir un mayor grado de autonomía, que pasa por alargar la vida de las baterías.

En este sentido, las técnicas de control basado

en eventos [11] como alternativa a los métodos de muestreo periódico son efectivas al disminuir el número de muestras y de actualizaciones de la señal de control. En el control basado en eventos, dichas tareas tienen lugar cuando se viola una cierta condición sobre el estado del sistema. Por tanto, hay una adaptación a las necesidades del proceso, que el caso del Crazyflie 2.1 implicarían un mayor tiempo de vuelo, al reducirse el número de actualizaciones y por tanto el uso de energía de las baterías. Sin embargo, en las estrategias de control basadas en eventos, el ruido en los sensores puede reducir la efectividad de esta técnica, ya que este tipo de perturbaciones aumentan el número de eventos detectados [23]. Configurar adecuadamente el controlador para minimizar el impacto del ruido es importante en estos casos.

El principal objetivo de este trabajo es diseñar algoritmos de control de posición y trayectoria para el Crazyflie. Utilizando Simulink se han implementado distintas soluciones haciendo uso de una estructura de control PID paralelo: tanto controladores de tipo periódico, en tiempo continuo y discreto, como controladores basados eventos. En estos últimos controladores se utiliza una estrategia tradicional del tipo “send-on-delta” y otra de tipo “adaptativo” que tiene en cuenta el error relativo respecto a la referencia y el nivel de ruido permitido. Se pretende probar las distintas soluciones analizando las ventajas que aporta cada una.

El trabajo está organizado de la siguiente manera. En la sección 2 se describe brevemente el modelo dinámico del Crazyflie, comentando su arquitectura de control a bajo nivel. A continuación, en la sección 3 se analizan los principios básicos del control basado en eventos y se comenta el diseño propuesto para el controlador basado en eventos. La sección 4 presenta los resultados de simulación obtenidos. Finalmente, en la sección 5, se comentan las conclusiones del trabajo y algunas posibles líneas de actuación futuras.

2 Crazyflie 2.1

El presente trabajo se desarrolla sobre la plataforma experimental de código abierto

Crazyflie 2.1, figura 1. Se trata de un nano-cuadróptero por su reducido tamaño ($92 \cdot 92 \cdot 29 [mm]$) y peso ligero ($27 [g]$). Su diseño, especialmente pensado para vuelo en interiores, junto con la flexibilidad que ofrece en su programación, ha propiciado su auge en el sector educativo y de investigación. En la tabla 1 se muestran los parámetros físicos del dron considerados en el presente trabajo. En la figura 1 se muestran los ejes establecidos para el modelado del dron. Durante el desarrollo se hará referencia al marco global, OXYZ, mediante el índice “o”. Para el caso de las coordenadas locales, O’X’Y’Z’, el índice empleado es “b”.



Figura 1: Cuadróptero Crazyflie 2.1.

Tabla 1: Parámetros físicos del Crazyflie 2.1.

Parámetro	Valor
Masa del dron (m)	$27[g]$
Longitud del brazo (d)	$39.73 \cdot 10^{-3}[m]$
Radio del rotor (r)	$23.135 \cdot 10^{-3}[m]$
Inercia eje x (I_{xx})	$1.36 \cdot 10^{-5}[Kg \cdot m^2]$
Inercia eje y (I_{yy})	$1.43 \cdot 10^{-5}[Kg \cdot m^2]$
Inercia eje z (I_{zz})	$2.17 \cdot 10^{-5}[Kg \cdot m^2]$
Constante de empuje (k_T)	0.2025
Constante de torque (k_D)	0.11

2.1 Modelo

En esta sección se realiza el modelado en el espacio de estados basado en leyes físicas para un cuadróptero genérico empleando la formulación matemática de Newton-Euler. El vector de estados está compuesto por cuatro vectores asociados a los 6 grados de libertad del dron y sus respectivas velocidades, tabla 2.

Como hipótesis iniciales, se considera que el dron es un cuerpo rígido, simétrico y cuyos cuatro rotores son idénticos. De igual forma, se considerará su configuración en “X” para los cálculos. Para determinar las ecuaciones relacionadas con

Vector	Estado	Descripción
p	x	Posición X
	y	Posición Y
	z	Posición Z
Φ	ϕ	Ángulo Roll
	θ	Ángulo Pitch
	ψ	Ángulo Yaw
V	u	Velocidad Lineal X
	v	Velocidad Lineal Y
	w	Velocidad Lineal Z
ω	p	Velocidad Angular X
	q	Velocidad Angular Y
	r	Velocidad Angular Z

Tabla 2: Vector de estados.

los parámetros lineales del dron, se aplica la segunda ley de Newton, calculando la derivada de la velocidad según la ecuación de Coriolis, ecuación (1), donde el subíndice CG hace referencia al centro de gravedad del dron y b/o al movimiento relativo del centro de gravedad del dron respecto al origen de coordenadas.

$$\Sigma F = m \cdot \dot{V}_{CG}^o = m \cdot \left(\dot{V}_{CG}^b + \omega_{b/o} \times V_{CG} \right) \quad (1)$$

Considerando que en estacionario, tanto el ángulo Roll como el ángulo Pitch son nulos, el equilibrio de fuerzas permite determinar la derivada de la velocidad lineal del dron, tal y como se muestra en (2). La transformación correspondiente entre el marco global y la referencia local del dron viene dada por una primera rotación del ángulo Yaw, (ψ), seguida de la rotación del ángulo Pitch, (θ), y una tercera rotación de Roll, (ϕ) (ver [8]).

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{F_z}{m} \end{bmatrix} - R_o^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2)$$

Las derivadas de las variables referentes a la posición, se calculan proyectando las velocidades lineales del dron sobre el marco global mediante la matriz de transformación R_b^o .

$$\dot{p}^o = R_b^o \cdot V^b \quad (3)$$

La fuerza de empuje generada por cada hélice, expresada en Newtons, se determina mediante la ecuación (4), donde C_T es el coeficiente de empuje del motor y ω_i la velocidad de rotación de cada motor. La ecuación (5) se corresponde con el empuje total generado por los cuatro motores.

$$F_i = C_T \omega_i^2 \quad (4)$$

$$\Sigma F_i^b = \begin{bmatrix} 0 \\ 0 \\ C_T (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} \quad (5)$$

El coeficiente de empuje de un motor depende de una constante adimensional k_T , la densidad del aire, ρ y del radio del rotor, r :

$$C_T = k_T \rho \frac{(2r)^4}{3600} \quad (6)$$

Para determinar los componentes angulares del vector de estados, se emplea el equilibrio de momentos. Aplicando el teorema del momento angular e introduciendo la ecuación de Coriolis, se obtiene la ecuación (7) aplicada al marco global, donde h representa el momento angular alrededor del centro de gravedad.

$$\Sigma M^o = {}^o \dot{h} \rightarrow \Sigma M^o = {}^b \dot{h} + \omega_{b/o} \times h \quad (7)$$

Si se considera el marco local del dron, el equilibrio de momentos se puede expresar mediante la ecuación (8), donde J representa la matriz de inercia. Bajo la hipótesis de simetría, J se puede expresar según la ecuación (9).

$$\Sigma M^b = J^b \dot{\omega}_{b/o} + \omega_{b/o} \times J \omega_{b/o} \quad (8)$$

$$J = \text{diag}(I_{xx}, I_{yy}, I_{zz}) \quad (9)$$

Despejando el vector $\dot{\omega}_{b/o}$ en (8) se obtiene:

$$\dot{\omega} = (J)^{-1} (M_{x,y,z} - \omega \times J \omega) \quad (10)$$

La última ecuación de estado se determina de la relación entre el vector $\omega_{b/o}$ y la derivada de Φ :

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (11)$$

El momento total generado por la acción de los rotores:

$$M = \Sigma P_i \times F_i + \Sigma \tau_i \quad (12)$$

donde P_i representa la distancia de cada motor al centro de gravedad, vectorialmente, y τ_i representa el momento inducido por cada motor. El momento total inducido por los rotores es:

$$\Sigma \tau_i^b = \begin{bmatrix} 0 \\ 0 \\ C_D (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \end{bmatrix} \quad (13)$$

Conociendo la fuerza que genera cada motor calculada según la ecuación (5), el momento que produce el empuje de cada motor es:

$$\begin{aligned} P_1 \times F_1 &= \frac{d}{\sqrt{2}} \begin{bmatrix} -C_T \omega_1^2 \\ -C_T \omega_1^2 \\ 0 \end{bmatrix} & P_2 \times F_2 &= \frac{d}{\sqrt{2}} \begin{bmatrix} -C_T \omega_2^2 \\ C_T \omega_2^2 \\ 0 \end{bmatrix} \\ P_3 \times F_3 &= \frac{d}{\sqrt{2}} \begin{bmatrix} C_T \omega_3^2 \\ C_T \omega_3^2 \\ 0 \end{bmatrix} & P_4 \times F_4 &= \frac{d}{\sqrt{2}} \begin{bmatrix} C_T \omega_4^2 \\ -C_T \omega_4^2 \\ 0 \end{bmatrix} \end{aligned} \quad (14)$$

El cálculo de los momentos, con referencia local, es el siguiente:

$$M^b = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} dC_T/\sqrt{2} (-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \\ dC_T/\sqrt{2} (-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2) \\ C_D (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \end{bmatrix} \quad (15)$$

El coeficiente de torque, C_D , se determina según la ecuación (16), [9], donde se muestra la dependencia con el coeficiente adimensional k_D , la densidad del aire, ρ y del radio del rotor, r .

$$C_D = k_D \rho \frac{(2r)^5}{3600} = 7.9379 \cdot 10^{-12} \left[\frac{Nm}{rpm^2} \right] \quad (16)$$

En la ecuación (15), los términos que incluyen aceleraciones angulares y momentos giroscópicos se han omitido ya que su contribución se puede considerar despreciable [20, 3].

2.2 Arquitectura de control

En la figura 2 se muestra el diagrama general de la arquitectura de control de un cuadricóptero. Esta es la más extendida en la literatura. En color azul y con el subíndice “dron” se representan las señales de realimentación medidas del dron. El caso de las señales con el subíndice “ref” se tratan de las señales de referencia para los controladores. La referencia global del dron está representada mediante el término “Pose3D”. Finalmente, “Control Mixer” recibe la señal de control de referencia de giro de los motores, Ω , mientras que $\Delta\phi, \Delta\theta, \Delta\psi$ representa la variación de velocidad de giro que debe realizar el dron para alcanzar la referencia. El “Control Mixer” es el encargado de aplicar la parte proporcional de cada término a cada motor.

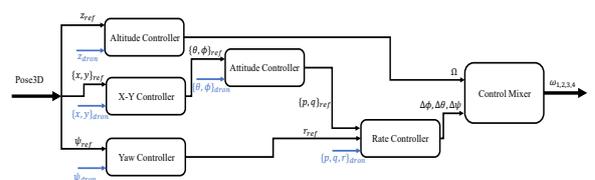


Figura 2: Arquitectura de control implementada.

Todos los controladores que incorpora la arquitectura anterior están basados en controladores del tipo PID, optándose por tomar como referencia las frecuencias recomendadas por el fabricante. En el caso del “Attitude Controller” y “Rate Controller”, la frecuencia de funcionamiento es de 500[Hz], mientras que los controladores de posición y del ángulo *Yaw* se ejecuta a 100[Hz]. La discretización de los controladores se ha realizado mediante la aproximación trapezoidal, considerando sus respectivos periodos de muestreo.

De estos en la tabla 3 solo se muestran los parámetros correspondientes a una configuración PID paralela para el controlador de altitud del dron, en el que se centra este trabajo. La señal de control que recibe el dron es un pulso PWM transmitido en una señal de 16 bits, por lo que el rango de la señal de control está comprendido entre [0, 65536]. Los parámetros empleados son una combinación extraída de [14, 22]. Para atenuar el ruido de alta frecuencia se ha implementado un filtro paso bajo en el término derivativo del PID.

Tabla 3: Parámetros PID.

Parámetro	Valor
K_p	15000 [bits/m]
K_i	3500 [bits/m]
K_d	9000 [bits/m]

3 Control basado en eventos

3.1 Fundamentos

El control basado en eventos supone un método de muestreo alternativo al periódico que se ha demostrado efectivo en disminuir el número de muestras, actualizaciones de la señal de control, etc. a la vez que se garantizan niveles de desempeño similares a un controlador convencional [11]. Su auge en los últimos veinte años ha venido motivado por las ventajas que aporta en sistemas de control en red o, más recientemente, en los llamados sistemas ciberfísicos, en los cuales existe un fuerte acoplamiento entre el control, la computación y la comunicación, y el diseño de estrategias ha de abordarse de manera integral [1]. El uso eficiente de estos recursos tiene un especial impacto en aquellos sistemas que están alimentados por baterías, como son los nano-drones, ya que permite prolongar la autonomía del sistema.

Normalmente, en un muestreo por eventos, se define una función de disparo dependiente de una señal de error y una cota, de tal manera que se produce un evento (muestreo) cuando dicha señal alcanza ese valor máximo definido.

Sea $x(t)$ el estado del sistema a tiempo t . Se define el error como $e(t) = x(t_k) - x(t)$, donde t_k representa el último instante de muestreo. Éstos se definen de manera recursiva como

$$t_{k+1} = \inf\{t : t > t_k, f(e(t), x(t)) > 0\}, \quad (17)$$

donde $f(e(t), x(t))$ es la función de disparo. Ésta puede depender: a) del error y el estado (estrategias de *umbral relativo*), y su definición se deriva a través de análisis de estabilidad mediante funciones de Lyapunov [24, 26, 5] que permite garantizar estabilidad asintótica del sistema; b) sólo del

error -y otros parámetros constantes- como en el caso de las estrategias send-on-delta [12, 16] (o de *umbral fijo*), que solo garantizan que el estado permanecerá confinado en una región en torno al equilibrio; c) una combinación de ambas estrategias y de otros parámetros como el tiempo, que permiten buscar una solución de compromiso entre ambas situaciones [10, 17]. Esta solución también se ha presentado en la literatura cuando el estado completo del sistema no está disponible, ya que el uso de la estrategia de umbral relativo puro lleva a lo que se conoce como efecto Zeno, es decir, la ocurrencia de infinitos eventos en un intervalo de tiempo finito [6].

3.2 Control basado en eventos para Crazyflie 2.1

Con el objetivo de disminuir la frecuencia promedio del controlador, se ha propuesto un diseño de control basado en eventos en el cual se define una función de disparo para las variables de estado x, y, z . Si denotamos de manera genérica una de estas variables como p_i , el error asociado al muestreo basado en eventos de la variable p_i como e_i , y el tiempo de muestreo de dicha variable como t_k^i , la función de disparo propuesta es la siguiente:

$$f_i(e_i(t), p_i(t)) = |e_i(t)| - (c_0^i + a_i |p_i(t) - p_{ref}^i| + c_n^i(t)), \quad (18)$$

donde c_0^i y a_i son parámetros constantes, $c_n^i(t)$ es un parámetro que se ajusta al nivel de ruido de la señal y p_{ref}^i es la señal de referencia para la variable p_i . Nótese que en este caso la cota consta de tres términos: uno constante dado por c_0^i que nos determinará el error máximo que tengamos en torno a la referencia, uno relativo a la distancia a la referencia ponderado por a_i , y un tercer término dado por $c_n^i(t)$ que permite acomodar el muestreo a la presencia de ruido, que puede generar eventos innecesarios. El cálculo de este término se basa en la estimación del valor de ruido al que está sometida la señal [23]. Para ello se determina $c_n^i(t)$ como el valor máximo de desviación de la medida respecto al cálculo del valor medio de la señal calculado mediante una ventana móvil de longitud predefinida:

$$c_n^i(t) = \max(|y_l[n] - \sum_{i=1}^l y_l[i]|), \quad (19)$$

donde l es la longitud de la ventana analizada e $y_l[n]$ el vector de medidas estudiado.

De esta manera, cuando el sistema está alejado de la referencia, se toleran valores de e_i superiores a cuando el sistema se acerca al punto deseado, lo cual permite mantener un nivel adecuado de desempeño del controlador pero reduciendo el número de actualizaciones de la señal de control.

4 Resultados

El caso de estudio se centra en los resultados obtenidos para el control de altitud del dron. Para la comparativa se toman tres casos de estudio en los cuales los parámetros del PID permanecerán constantes (ver tabla 3). En el primero, el controlador PID será discretizado mediante la aproximación trapezoidal con un periodo constante de 10 [ms]. En el segundo caso, se plantea un disparador por eventos con una cota constante de 1 [cm]. Este parámetro, en la práctica experimental está condicionado por la precisión del sistema de localización empleado. Finalmente, el tercer caso contempla la condición de disparo descrita en (18), con $c_0^i = 1$ [cm] y $a = 1.5$ [cm]. El parámetro $c_n^i(t)$, fundamental para evitar falsos disparos a causa del ruido, se ha estimado a partir del error máximo respecto a una media de ventana móvil con una longitud de 20 muestras. Al tratarse de una simulación, el ruido se ha añadido de forma artificial mediante una señal de ruido blanco con ancho de banda limitado. La señal se ha simulado con una densidad espectral de potencia de $1 \cdot 10^{-8}$ y una frecuencia de funcionamiento de 1[kHz]. El controlador PID implementado incluye el filtro paso bajo en la derivada.

$$\frac{U(s)}{E(s)} = K_p + K_i \frac{1}{s} + K_d \frac{s}{1 + \frac{K_d}{K_p N_d \cdot s}} \quad (20)$$

Vasyutynskyy y Kabitzsch [25] presentan, una implementación de este controlador basada en eventos para cada instante de muestreo k:

$$u_k = P_k + I_k + D_k \quad (21)$$

$$P_k = K_p \cdot e_k \quad (22)$$

$$I_k = I_{k-1} + K_i \cdot T_k \cdot e_{k-1} \quad (23)$$

$$D_k = \frac{K_d}{K_d + N_d \cdot T_k \cdot K_d} D_{k-1} + \frac{K_d \cdot N_d}{\frac{K_d}{K_p} + N_d \cdot T_k} \cdot (e_k - e_{k-1}) \quad (24)$$

donde P_k , I_k y D_k son las acciones proporcional, integral y derivativa, respectivamente, T_k representa el tiempo transcurrido desde el último evento y N_d es el parámetro característico del filtro de la derivada cuyo valor adoptado es 20.

La trayectoria empleada en este estudio se muestra en la figura 3. Se propone elevar el dron a una altura de un metro y dibujar una trayectoria de un cuadrado de 1 [m] de lado a dicha altura. En dos de los lados del cuadrado se reduce la consigna del eje Z en un 20% para observar su comportamiento mejor sacándolo de estacionario en dos ocasiones. En la figura 4 se muestran los datos obtenidos para el movimiento en el eje Z (gráfica superior) y la

correspondiente señal de control generada por el controlador transformando la señal PWM a la velocidad angular de cada rotor.

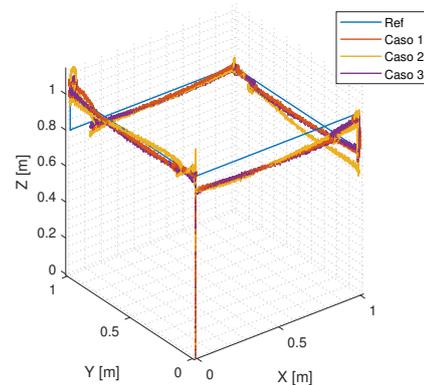


Figura 3: Trayectoria realizada por el Crazyflie.

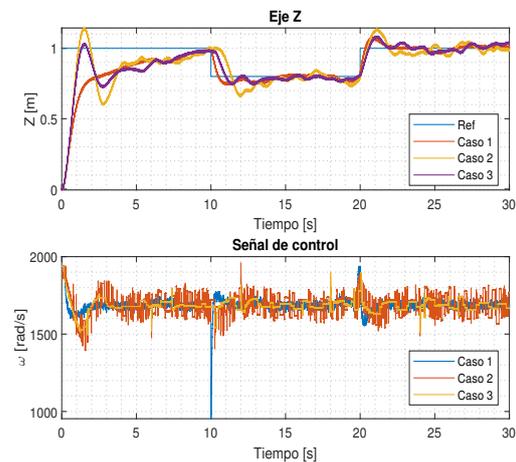


Figura 4: Control altitud Crazyflie 2.1.

A simple vista, se observa la discrepancia entre el control del PID discretizado y las dos implementaciones del control basado en eventos. La mayor discrepancia de comportamiento se encuentra en el despegue, donde en el caso de los basados en eventos, se aprecia una clara sobreoscilación sobre el controlador discreto. Los tres controladores parecen converger a partir del instante $t = 5[s]$, aunque los controladores basados en eventos mantienen cierto rizado en estacionario. Atendiendo al comportamiento de la señal de control, se observa que para el caso 2, el valor adoptado para el filtro en la derivada no es suficientemente bajo. El valor del parámetro del filtro $N_d = 20$, se ha escogido tras un barrido del mismo. Para el caso 2 concretamente, debería tomar el valor 7 para conseguir la reducción del ruido en la señal de control. No obstante, dicho valor provoca que el sistema os-

cile al llegar a estacionario por lo que empeora significativamente su desempeño. Con la finalidad de analizar el desempeño de los controladores diseñados se utilizan distintos índices: el número de muestras (N), la Integral del Error Absoluto (*Integral of Absolute Error*, IAE), la Integral del Tiempo por el Error Absoluto (*Integral of Time-weighted Absolute Error*, ITAE) y el consumo energético. Para este último índice, sólo se ha considerado la velocidad de giro de los rotores. Una estimación más precisa y que reforzaría los resultados obtenidos debería considerar el consumo intrínseco asociado al procesado de mensajes por parte del dron. En la tabla 4 se muestran los valores de cada índice para cada uno de los casos.

Parámetro	Caso 1	Caso 2	Caso 3
N	3000	1027	207
IAE [m]	2.2568	2.6417	2.1825
ITAE [m · s]	12.485	21.224	14.989
Consumo [W]	13.909	13.931	13.897

Tabla 4: Resultados analíticos.

Se observan claras diferencias entre los distintos controladores. Como era de esperar, el control basado en eventos reduce notablemente la cantidad de mensajes en la arquitectura de control. En este sentido, para el caso 2 se consigue una reducción del 65.77% y para el caso 3 se alcanza una reducción del 93%. Atendiendo al desarrollo propuesto al control basado en eventos mediante el caso 3, las mejoras son más significativas que respecto al controlador discreto (3.29% de mejora en la IAE). En el número de eventos, la mejora es del 79.844%, mejora el desempeño en la IAE e ITAE un 17.383% y 29.374% respectivamente.

En la figura 5 se observa la comparativa de generación de eventos entre el caso con una cota variable (Caso 3) y la cota fija (Caso 2). Dado que el sistema es muy rápido para visualizar con detalle la generación de eventos, en las figuras 6 y 7 se presenta una ampliación. Se observa claramente como el umbral se ajusta al ruido y como resultado se consigue reducir la generación de eventos. En ese caso se muestra durante el periodo transitorio del movimiento del dron en Z . En la figura 7 se muestra durante un periodo estacionario donde se aprecia el valor prácticamente nulo del término a de la ecuación (18) y la adaptación al ruido.

Finalmente, en la figura 8, se muestra la variación del periodo entre las acciones de control para los tres casos estudiados. Como es lógico, en el caso del controlador discreto, el periodo es constante a 0.01[s]. En el caso del umbral fijo, el periodo máximo que se alcanza es 0.337[s]. Los

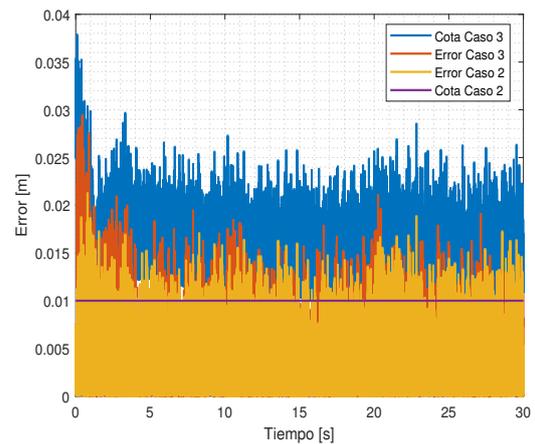


Figura 5: Comparativa de generación de eventos.

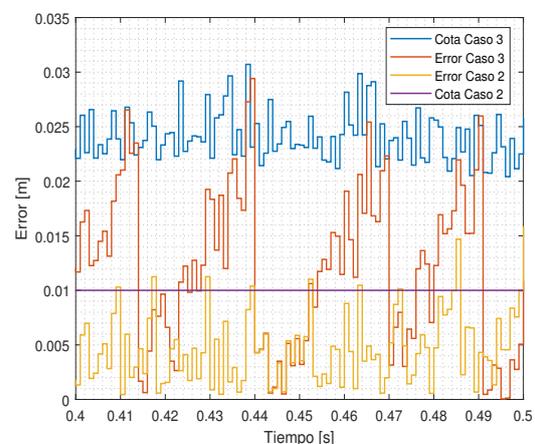


Figura 6: Generación de eventos. Transitorio.

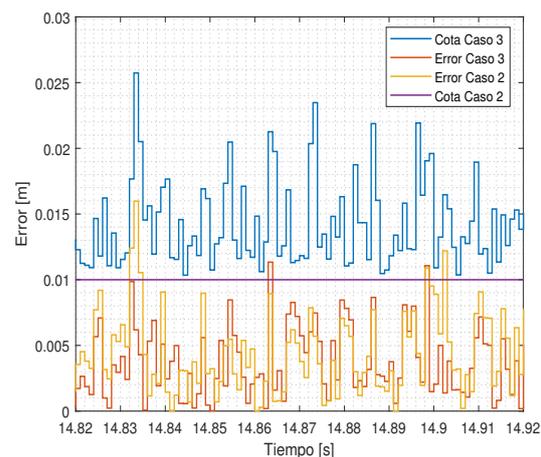


Figura 7: Generación de eventos. Estacionario.

mejores resultados, como era de esperar, se alcanzan para el caso del umbral relativo, logrando hasta un máximo de 1.25[s], manteniéndose por encima siempre de las otras dos implementaciones.

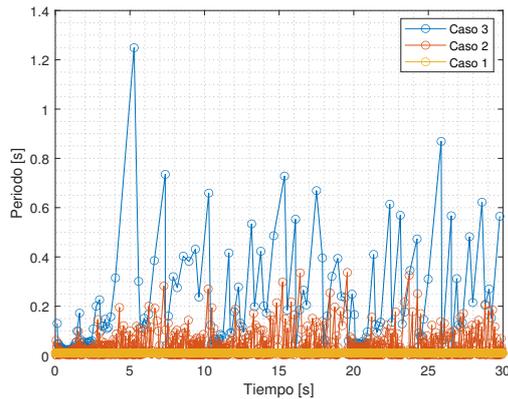


Figura 8: Período entre acciones de control.

5 Conclusiones

En este trabajo se ha probado un cálculo adaptativo del umbral de disparo de un sistema de control basado en eventos aplicado al control de altitud del nano-dron Crazyflie 2.1. Se han comparado los resultados obtenidos para el umbral relativo (Caso 3) con el desempeño de un umbral constante (Caso 2) y con un controlador PID discretizado con periodo constante (Caso 1). Los resultados muestran una mejora en el número de cambios en la señal de control, siendo máximos para el caso 1 (3000) y mínimos, con una reducción del 93% para el caso 3 (207 señales).

Una vez verificadas las bondades de la arquitectura de control propuesta, el siguiente paso es extenderla al resto de controladores, figura 2. En el caso del índice de consumo, se puede estudiar la inclusión de gasto asociado al procesamiento de los mensajes de control para observar el efecto de la mejora en la duración de la batería, parámetro crítico en los robots de tamaño tan reducido. Además, se abre un abanico de opciones de estudio modificando las condiciones de operación, tales como el equipamiento de los robots, inclusión de perturbaciones o evaluar la robustez del sistema frente a pérdida de mensajes en la comunicación.

Agradecimientos

Este trabajo fue financiado en parte por los proyectos CICYT RTI2018-094665-B-I00 y DPI2017-84259-C2-2-R del Ministerio de Ciencia e Innovación

English summary

Event-based control with relative threshold applied to Crazyflie 2.1 quadrotor altitude control

Abstract

In this work the design of an event-based controller for the altitude controller of the Crazyflie 2.1 is presented. The novelty is based on the implementation of the event generator by means of a relative threshold. The adaptation of this threshold is carried out according to the distance to the reference and the presence of noise in the system. This obtained results show a 93% reduction of the number of control signal updates with respect to event-based controllers with a fixed threshold or periodic controllers.

Keywords: PID Controller, adaptive sampling, nano-dron.

Referencias

- [1] E. Aranda-Escolástico, M. Guinaldo, R. Heradio, J. Chacon, H. Vargas, J. Sánchez, and S. Dormido. Event-based control: a bibliometric analysis of twenty years of research. *IEEE Access*, 8:47188–47208, 2020.
- [2] Bitcraze. Crazyflie 2.1 — bitcraze. <https://www.bitcraze.io/products/crazyflie-2-1/>. Accessed: 2021-06-12.
- [3] S. Bouabdallah, A. Noth, and R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2451–2456, 2004.
- [4] N. Bucki and M.W. Mueller. Improved quadcopter disturbance rejection using added angular momentum. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4164–4170, 2018.
- [5] C. De Persis, R. Sailer, and F. Wirth. Parsimonious event-triggered distributed control: A zeno free approach. *Automatica*, 49(7):2116–2124, 2013.
- [6] M.C.F. Donkers and W.P.M.H. Heemels. Output-based event-triggered control with guaranteed \mathcal{L}_∞ -gain and improved and decentralized event-triggering. *IEEE Transactions on Automatic Control*, 57(6):1362–1376, 2012.

- [7] W. Giernacki, M. M. Skwierczyński, W. Witwicki, P. Wroński, and P. Koziński. Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 37–42, 2017.
- [8] Marcus Greiff. *Modelling and control of the crazyflie quadrotor for aggressive and autonomous flight by optical flow driven state estimation*. PhD thesis, Department of Automatic Control, Lund University, 2017.
- [9] E.M. Greitzer, Z.S. Spakovszky, and I.A. Waitz. Thermodynamics and propulsion. *Mechanical Engineering, MIT*, 2006.
- [10] M. Guinaldo, D.V. Dimarogonas, K.H. Johansson, J. Sanchez, and S. Dormido. Distributed event-based control strategies for interconnected linear systems. *IET Control Theory and Applications*, 7(6):877–886, 2013.
- [11] W.P.M.H. Heemels, K.H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *2012 IEEE 51st IEEE conference on decision and control (cdc)*, pages 3270–3285, 2012.
- [12] W.P.M.H. Heemels, J.H. Sandee, and P.P.J. Van Den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4):571–590, 2008.
- [13] M. Huptych and S. Röck. Real-time path planning in dynamic environments for unmanned aerial vehicles using the curve-shortening flow method. *International Journal of Advanced Robotic Systems*, 18(1), 2021.
- [14] C. Luis and J.L. Ny. Design of a trajectory tracking controller for a nanoquadcopter. *arXiv preprint arXiv:1608.05786*, 2016.
- [15] M. Mack, C. Noe, T. Rice, I.S. Ahn, and J. Wang. Distributed localization and control of quadrotor UAVs using ultra-wideband sensors. In *2019 IEEE Aerospace Conference*, pages 1–9, 2019.
- [16] M. Miskowicz. Send-on-delta concept: An event-based data reporting strategy. *Sensors*, 6(1):49–63, 2006.
- [17] N. Mu, X. Liao, and T. Huang. Consensus of second-order multi-agent systems with random sampling via event-triggered control. *Journal of the Franklin Institute*, 353(6):1423–1435, 2016.
- [18] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt. The robotarium: A remotely accessible swarm robotics research testbed. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1699–1706, 2017.
- [19] J.A. Preiss, W. Honig, G.S. Sukhatme, and N. Ayanian. CrazySwarm: A large nano-quadcopter swarm. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3299–3304, 2017.
- [20] F. Sabatino. Quadrotor control: modeling, nonlinear control design, and simulation, 2015.
- [21] Z. Shang and Z. Shen. Indoor testing and simulation platform for close-distance visual inspection of complex structures using micro quadrotor uav, 2019.
- [22] G. Silano, E. Aucone, and L. Iannelli. CrazyS: a software-in-the-loop platform for the Crazyflie 2.0 nano-quadcopter. In *2018 26th Mediterranean Conference on Control and Automation (MED)*, pages 1–6, 2018.
- [23] R. Socas, S. Dormido, and R. Dormido. Event-based controller for noisy environments. In *2014 Second World Conference on Complex Systems (WCCS)*, pages 280–285, 2014.
- [24] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.
- [25] V. Vasyutynskyy and K. Kabitzsch. Time constraints in PID controls with send-on-delta. *IFAC Proceedings Volumes*, 42(3):48–55, 2009.
- [26] X. Wang and M.D. Lemmon. Event-triggering in distributed networked control systems. *IEEE Transactions on Automatic Control*, 56(3):586–601, 2011.



© 2021 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).