

Análisis de la frecuencia de muestreo en sistemas multi-robot

Mañas-Álvarez, F. J.^{a,*}, Guinaldo, M.^a, Dormido, R.^a, Dormido, S.^a

^aDepartamento de Informática y Automática, E.T.S.I. Informática, Universidad Nacional de Educación a Distancia (U.N.E.D.),
C/ Juan del Rosal, 14, 28040, Madrid, España.

To cite this article: Mañas-Álvarez, F. J., Guinaldo, M., Dormido, R., Dormido, S. 2023. Sampling frequency analysis in multi-robot systems. XLIV Jornadas de Automática, 329-334. <https://doi.org/10.17979/spudc.9788497498609.329>

Resumen

Los Sistemas Ciber-Físicos, SCFs, tienen cada vez un papel más presente en la actualidad dada la sinergia que debe existir entre el mundo físico y digital en esta sociedad “conectada”. Una parte importante de estos sistemas la ocupan los Sistemas Multi-Robot, SMRs, que pueden aportar soluciones más eficientes a gran parte de los retos de la automatización actual frente a la solución homónima proporcionada por un único robot. En el presente trabajo se examina el diseño del protocolo de comunicación entre agentes de un SMR para un problema de formación por consenso de robots aéreos. Se determina la frecuencia mínima operativa a la que debe ejecutarse la comunicación periódica entre agentes conectados, 10 Hz. Finalmente se analiza el umbral de disparo que debe tener un protocolo de comunicación basado en eventos para obtener una gestión más eficiente del canal de comunicación sin empeorar el desempeño del sistema de referencia periódico, 2 cm.

Palabras clave: UAVs, Sistemas multi-agente, Control basado en eventos, Control descentralizado y distribuido, Sistemas robóticos en red

Sampling frequency analysis in multi-robot systems

Abstract

Cyber-Physical Systems, CPSs, are playing an increasingly important role today given the synergy that must exist between the physical and digital world in this “connected” society. An important part of these systems is covered by Multi-Robot Systems, SMRs, which can provide more efficient solutions to many of today’s automation challenges compared to the homonymous solution performed by a single robot. This paper examines the design of the inter-agent communication protocol of an SMR for an aerial robot consensus training problem. The minimum operating frequency at which the periodic communication between connected agents should be executed, 10 Hz, is determined. Finally, the trigger threshold that an event-driven communication protocol should have to obtain more efficient management of the communication channel without worsening the performance of the periodic reference system, 2 cm, is analyzed.

Keywords: UAVs, Multi-agent systems, Event-based control, Decentralized and distributed control, Networked robotic systems

1. Introducción

En la era de la tecnología digital, los sistemas ciberfísicos (SCF) pueden considerarse una nueva generación de sistemas digitales con gran impacto en la aceleración del progreso tecnológico. Estos sistemas son el resultado de la sinergia entre sistemas físicos, computacionales y el marco de comunicación y control que los conforman (Lee, 2008). En el paradigma ac-

tual de la sociedad “conectada” juegan un papel clave en la integración de la gran cantidad de sensores, actuadores, sistemas computacionales y la gestión de los grandes volúmenes de datos que generan y requieren en su funcionamiento del desarrollo tecnológico (García-Valls et al., 2018). La naturaleza heterogénea de los SCFs hace que sea imprescindible la correcta colaboración entre distintas ramas del conocimiento para su diseño e implementación. Ejemplos de SCFs incluyen aplicacio-

*Autor para correspondencia: fjmanas@dia.uned.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

nes en muchas áreas como aeroespacial, transporte, fabricación, automoción, etc. (Zanero, 2017).

Uno de los principales ámbitos beneficiados del desarrollo de estos sistemas es la robótica móvil. La demanda creciente de soluciones autónomas y conectadas requiere sistemas cada vez más complejos. En muchos casos, los requerimientos que deben cumplir las soluciones aportadas pueden ser cubiertas por un sólo agente. No obstante, la complejidad que puede requerir el diseño de este agente o el coste computacional asociado a la ejecución de la tarea requerida puede hacer la solución inviable por un único agente. En este sentido, los Sistemas Multi-Robot, SMR (Cortés and Egerstedt, 2017), como los que se emplean en este trabajo, permiten mejorar la eficiencia en el procesamiento de datos al permitir que los robots ejecuten colectivamente tareas más complicadas que las que podría completar un solo robot. El SCF utilizado en este trabajo, *Robotic Park*, es un SMR heterogéneo desarrollado para operar con robots aéreos del tipo nano-cuadricópteros y robots móviles diferenciales. Estos robots disponen de algoritmos desarrollados para la realización de tareas cooperativas como la resolución de un problema de formación o de navegación. El diseño e implementación de soluciones para este tipo de sistemas plantean distintos desafíos desde el punto de vista del control.

Al tratarse de sistemas distribuidos con recursos limitados, la gestión adecuada de tales recursos es uno de los aspectos que más atención requiere. En nuestro caso, los robots emplean canales con restricciones de volumen en la comunicación y los recursos energéticos también es factor limitante, por lo que la gestión eficiente de recursos se convierte en un problema crítico. Hay una serie de compromisos que se tienen que considerar en la fase de diseño del sistema. Por ejemplo, los relacionados con la elección de la frecuencia de muestreo, con los retardos o la cuantificación (Franklin et al., 2005). En concreto, la frecuencia de muestreo de las actualizaciones de la señal de control plantea un compromiso entre el comportamiento de control y la carga del procesador. Por un lado la frecuencia de muestreo está estrechamente relacionada con el ancho de banda. Sintonizando cuidadosamente el controlador un aumento en la frecuencia implica un aumento en el ancho de banda que se puede llegar a obtener. Pero por otro, si la aplicación tiene capacidad de procesamiento limitada, se hace necesario reducir la frecuencia de muestreo, requiriéndose mayor reducción cuanto más complejo es el algoritmo de control. Es por lo tanto importante el análisis de la frecuencia de muestreo del sistema bajo estudio.

Por otro lado, pensando en la reducción de la pérdida de ancho de banda de comunicación, las estrategias de control basadas en eventos suponen una herramienta de gran utilidad frente a otras soluciones activadas por tiempo (Heemels et al., 2012). Las estrategias de muestreo o control basadas en eventos cambian el paradigma clásico de operar de forma periódica a intervenir sólo cuando es necesario. De esta forma, se reduce la transmisión innecesaria de mensajes y el consumo energético asociado.

El principal objetivo de este trabajo es analizar el muestreo de la realimentación en el problema de control de formación por consenso en un SMRs. Para ello se plantea una formación de cúpula que deben cubrir un total de cinco robot aéreos empleando como restricciones en su ley de control la distancia euclídea entre ellos y respecto al centro de la cúpula. Las señales de re-

alimentación analizadas en este caso serán las transmisiones de cada robot al resto del sistema en la que indican su posición. En primer lugar se analizará el desempeño del sistema con una transmisión periódica a la máxima frecuencia que permite el sistema de posicionamiento interno de los robots, 50Hz. Esta frecuencia se reduce progresivamente hasta que los índices de desempeño analizados muestran una degeneración en la respuesta del sistema. Durante el barrido realizado se determina la frecuencia periódica mínima funcional. Tomando de referencia este dato, se propone una estrategia de muestreo basado en eventos con un umbral de disparo constante en la que analizando el error cometido se determina la configuración que permite un desempeño equivalente con una reducción significativa en el muestreo de las señales analizadas.

El trabajo está organizado de la siguiente manera. En la sección 2 se describe la plataforma *Robotic Park* y el robot *Crazyflie 2.1*, comentando su arquitectura de control a bajo nivel. A continuación, en la sección 3 se analizan los principios básicos de los SMRs y del muestreo basado en eventos. En la sección 4 se presentan las características de las experiencias realizadas y los índices que se emplearán para analizar los resultados. La sección 5 presenta los resultados obtenidos. Finalmente, en la sección 6, se comentan las conclusiones del trabajo y algunas posibles líneas de actuación futuras.

2. Plataforma experimental

Las experiencias realizadas se han llevado a cabo en la plataforma experimental *Robotic Park*, Figura 1 (Mañas-Álvarez et al., 2023). Esta plataforma está enfocada para la experimentación con SMRs en interiores con agentes heterogéneos. El espacio de trabajo tiene un volumen de $3,6 \times 3,6 \times 2$ m que posibilita la operación de múltiples robots de tamaño reducido en tareas de formación y navegación. *Robotic Park* cuenta con diferentes sistemas de posicionamiento que utilizan distintas tecnologías (captura de movimiento, UWB, infrarrojos, etc.). Esto permite combinar diferentes agentes que por su tipología no acepten el mismo sistema de posicionamiento. Dada la versatilidad que se persigue en la plataforma, se dispone de robots comerciales que ofrecen soluciones cerradas con el interés enfocado en sus sensores y el control colaborativo, y robots de código abierto que permiten una mayor manipulación de las configuraciones de los robots. Entre los robots disponibles destacan los robots móviles *Khepera IV* (Tharin et al., 2015) y *Turtlebot 3 Burger* (Amstels and Slaets, 2020), y los robots aéreos *DJI Tello* (Giernacki et al., 2022) y los *Crazyflie 2.X* (Giernacki et al., 2017).

La integración de todos los elementos que componen *Robotic Park* (robots, sistemas de posicionamiento, PC, etc) se realiza a través de ROS 2 (*Robot Operating System 2*) (Macenski et al., 2022) en su distribución Humble. Este sistema es el estándar de código abierto más empleado en robótica. Permite implementar en la misma red arquitecturas centralizadas y distribuidas para la comunicación entre componentes. El uso de tipos estándar en las variables que se emplean facilita la integración de nuevos componentes que ya dispongan de drivers para este sistema sin requerir modificaciones adicionales en el entorno o en el resto de agentes.


 Figura 1: Plataforma experimental *Robotic Park*.

2.1. Crazyflie 2.1

El Crazyflie 2.X, Figura 2, es una plataforma de código abierto desarrollada por Bitcraze (Giernacki et al., 2017). Se trata de un micro-cuadróptero por su reducido tamaño (92 mm×92 mm×29 mm), baja masa (27 g), e inercia. Sus características lo hacen especialmente recomendable para la experimentación en interiores. Al tratarse de una plataforma de bajo coste y código abierto, existe una gran comunidad de desarrolladores trabajando con ella. Esto facilita encontrar gran cantidad de trabajos disponibles que aceleran su desarrollo y facilitan su uso en tareas más específicas.

Este robot dispone de una API en Python que permite la comunicación a través de la antena Crazyradio PA. Cuenta con un microcontrolador STM32F405 y entre sus sensores destacan las cubiertas *multiranger*, que permiten obtener la distancia a obstáculos en los ejes principales del robot; el *Flow deck v2*, para la estimación del movimiento en el plano X-Y mediante odometría visual; y el *AI-deck*, que permite un aumento de la computación y comunicación a través del procesador de bajo consumo GAP8.

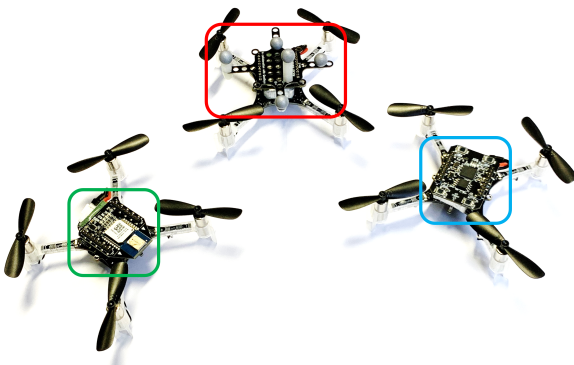


Figura 2: Crazyflies con las cubiertas para los sistemas Loco Positioning (izquierda, verde), Vicon Tracker (centro, rojo) y Lighthouse (derecha, azul).

El modelo de este tipo de robot ha sido muy estudiado en la literatura (Feng et al., 2015). Las soluciones de control propuestas van desde algoritmos no lineales avanzados (Esmail et al., 2022) hasta los clásicos controladores lineales PID (Hasseni et al., 2021). Los controladores lineales tienen un mejor ren-

dimiento cuando el robot trabaja en torno a un punto de funcionamiento. La arquitectura de control a bajo nivel de este tipo de agentes suele tener dos niveles formados cada uno por un sistema en cascada, tal y como se muestra en la Figura 3. El nivel superior tiene una frecuencia típica de 100Hz y es el responsable de controlar la posición y la velocidad del robot. Su señal de entrada es la posición objetivo y sus salidas son el empuje y los ángulos de cabeceo y balanceo. El siguiente nivel es el control de orientación de los ángulos de cabeceo y balanceo tanto en posición como en velocidad. Requiere una frecuencia de funcionamiento mayor, 500Hz, y sus señales de salida son los comandos que deben recibir los rotores de cada robot.

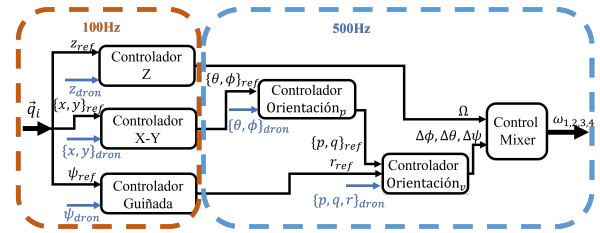


Figura 3: Diagrama de bloques de la arquitectura de control típica a bajo nivel de un Crazyflie 2.X.

3. Sistema Multi-Robot

Como se ha comentado anteriormente, en ocasiones, la mejor solución para determinados problemas se consigue mediante el desarrollo de sistemas que combinen múltiples robots. El SMR se modela normalmente en términos de un grafo $\mathcal{G}(\mathcal{V}, \mathcal{E})$, donde $\mathcal{V} = \{v_1, \dots, v_N\}$ es un conjunto finito de N vértices que representan los nodos o agentes y $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ es un conjunto finito de aristas, que representan los enlaces de comunicación entre ellos. Estos sistemas tienen asociados una serie de desafíos que deben abordarse en su etapa de diseño. Entre ellos, la implementación de un nivel de control inmediatamente superior al control de bajo nivel de cada agente y la comunicación entre agentes, que se resuelve en este trabajo tal y como se explica a continuación.

3.1. Arquitectura de control

La implementación del nivel de control encargado de coordinar los movimientos entre agentes para garantizar que se alcanza la formación deseada puede ser centralizada o descentralizada, siendo la segunda opción la preferible cuando el número de agentes es grande. Además, una implementación distribuida en la que un nodo sólo se comunica con un subconjunto de agentes (vecinos), definido por el grafo, reduce los retardos en la transmisión de señales entre los distintos niveles de control. En estos sistemas, la formación deseada puede definirse a partir de posiciones relativas o distancias entre agentes. En este caso, la formación viene definida por un conjunto de distancias objetivo. Es decir, si dos nodos i y j están conectados en el grafo por una arista, entonces la distancia entre ellos, d_{ij} , debe alcanzar un valor deseado d_{ij}^* . Tanto la conexión entre nodos como las distancias objetivo d_{ij}^* se suponen dadas para garantizar la formación deseada.

Si $p_i \in \mathbb{R}^3$ es la posición del agente i el controlador de formación se define según la siguiente ecuación:

$$u_i^s = - \sum_{j \in N_i} \mu_{ij} ((d_{ij}^*)^2 - \|p_i - p_j\|^2)(p_i - p_j), \quad (1)$$

donde $\mu_{ij} > 0$ es la ganancia del controlador asociado al enlace entre los agentes i y j y $\|p_i - p_j\| = d_{ij}$ representa la distancia entre ellos. Normalmente, se puede considerar un valor igual para todas las ganancias, es decir, $\mu_{ij} = \mu, \forall i, j$.

Obsérvese que cuando todas las distancias entre i y sus vecinos $j \in N_i$ convergen a los valores objetivo d_{ij}^* , la señal de control u_i^s se aproxima a cero.

3.2. Muestreo basado en eventos

Uno de los principales desafíos de los SMRs que emplean agentes como los Crazyflies es la gestión de recursos. Especialmente, cuando los recursos energéticos son escasos, la reducción de consumos juega un papel fundamental para poder extender la duración de las experiencias. En el control de formación, la información clave que deben transmitir los agentes entre sí son sus posiciones. En el caso presentado en este trabajo, los Crazyflies utilizan un sistema de posicionamiento interno y emplean el canal de comunicación con la red de ROS 2 para publicar su posición y suscribirse a la de los agentes vecinos.

Estas transmisiones no pueden realizarse de forma continua al implementarse en un sistema digital, sino en instantes discretos de tiempo. La elección de estos instantes de tiempo suponen un factor crítico en el desempeño de los niveles de control que hacen uso de la información transmitida. Se propone un protocolo de comunicación basado en eventos frente al muestreo periódico clásico. El muestreo basado en eventos es una alternativa al muestreo periódico que ha demostrado ser eficaz para reducir el número de muestras (Heemels et al., 2012). La idea principal es que es el estado/la salida del sistema y no el tiempo lo que determina cuándo muestrear el sistema. De esta forma se pretende obtener una gestión más eficiente de los instantes discretos de tiempo para la realización de transmisiones.

En este caso, la función de disparo se define en base a una función de error que depende de la posición del agente y de un umbral, de forma que se produce un evento (muestreo) cuando dicha función alcanza ese valor límite. Normalmente, la función de error $e(t)$ se define como la norma de la diferencia entre la medida actual del estado $x(t)$ y la última medida transmitida, $x(t_k)$. Existen diferentes propuestas para el umbral en la literatura (constante, adaptativo, etc.) pero, en términos generales, cuanto mayor sea el umbral, menor será la tasa de eventos. Así, el tiempo de disparo de un sistema de control basado en eventos se define mediante la ecuación recursiva dada por

$$t_{k+1} = \inf\{t : t > t_k, f(e(t), x(t)) > 0\}, \quad (2)$$

donde $e(t) = x(t_k) - x(t)$ es el error de la función y $f(e(t), x(t))$ representa la función de disparo. La evaluación de la función de disparo sí se hace de forma periódica en la salida del estimador interno del agente a una frecuencia de 50 Hz.

4. Desarrollo experimental

El objetivo del presente trabajo es establecer el comportamiento de un SMR cuando se emplea un muestreo periódico de

sus agentes, la frecuencia periódica mínima que garantiza un desempeño adecuado y el umbral de una implementación basada en eventos que garantice un comportamiento equivalente al obtenido con muestreo periódico y que reduzca de forma significativa las transmisiones empleadas por cada agente al resolver un problema de formación basado en distancias. La representación tridimensional del grafo que modela el sistema y las distancias deseadas se muestran en la Figura 4. El conjunto finito de vértices se encuentra compuesto por un total de cinco agentes del tipo Crazyfly 2.1 ($CF01, CF02, CF03, CF04, CF05$) que deben mantener una distancia entre sí $d_{ij}^* = 0,70 \text{ m}$. Esto implica una formación de semiesfera de radio $0,50 \text{ m}$. Adicionalmente se considera un sexto agente (*Origin*), virtual, ubicado en el origen de coordenadas, O . Se toma esta medida para restringir los grados de libertad de la formación y evitar desplazamientos en conjunto que la pueda sacar del volumen de trabajo seguro. Este último agente sólo transmite su posición una vez al inicio de la experiencia. Los agentes $CF02 - 05$ deberán mantener una distancia $d_{i0}^* = 1,00 \text{ m}$ mientras que el dron $CF01$ deberá mantener una distancia $d_{10}^* = 1,37 \text{ m}$.

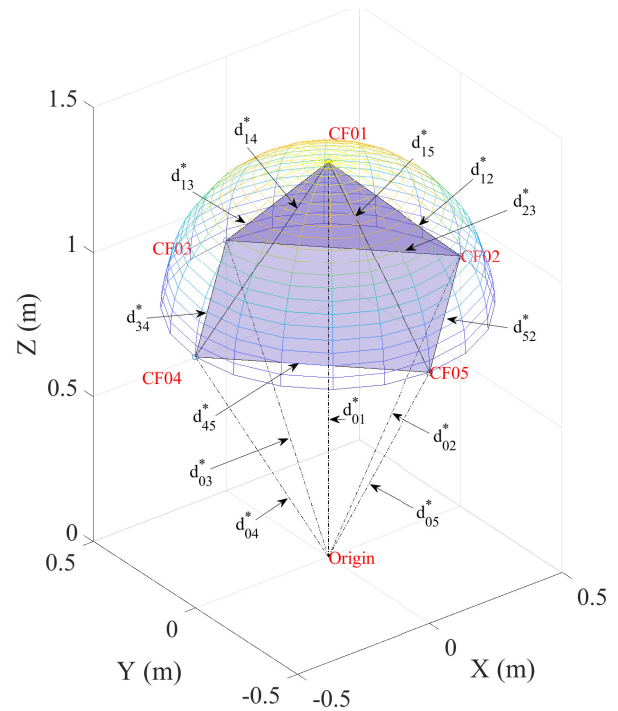


Figura 4: Representación gráfica tridimensional del Grafo $\mathcal{G}(\mathcal{V}, \mathcal{E})$ empleado en las experiencias desarrolladas.

Para el estudio del comportamiento del sistema con muestreo periódico se comenzará en la frecuencia máxima que admite la API de Python para leer el posicionamiento interno del robot, 50 Hz. A continuación se explorará la siguiente secuencia de frecuencias: 40 Hz, 33 Hz, 20 Hz, 10 Hz, 5 Hz, 1 Hz, 0,5 Hz. Para la evaluación de la arquitectura basada en eventos, se toma como punto de partida un umbral de 1 cm. Este umbral se incrementa progresivamente hasta un valor máximo de 8 cm. En el barrido de valores realizado se trabaja con umbrales intermedios de valores 2 cm, 4 cm y 6 cm.

Los índices de rendimiento analizados en las distintas experiencias son los siguientes:

- *Tiempo de establecimiento, t_e* . Tiempo que emplea la formación hasta que permanece en un límite del 5 % de error en la formación deseada respecto del error inicial.
- *Tiempo medio de muestreo, T* . Con este parámetro se analiza el número de transmisiones realizadas por los agentes a lo largo del experimento. Este índice representa el flujo de información a través del canal de comunicación.
- *Integral del Error Absoluto, IEA*. Este índice pondera todos los errores por igual a lo largo del tiempo. Proporciona información global sobre los agentes.
- *Integral del Error Absoluto ponderado en el Tiempo, IEAT*. En los sistemas que utilizan entradas escalonadas, el error inicial es siempre elevado. En consecuencia, para hacer comparaciones entre sistemas, es más relevante que los errores que se mantienen en el tiempo tengan un peso mayor que los errores iniciales.

Estos índices permiten evaluar cuantitativamente los experimentos realizados. En concreto, considerando tres parámetros: IEA, IEAT y tiempo de establecimiento, se puede medir por un lado la cantidad de mensajes transmitidos por cada agente y, por otro lado, estudiar el error en la formación.

5. Resultados

Todos los ensayos realizados han seguido la misma secuencia de acciones y duración para establecer una comparativa entre ellos. Inicialmente todos los robots han recibido la orden despegue y la cota $z = 0,7 \text{ m}$ como punto de equilibrio. Una vez alcanzado el estacionario han recibido la orden de iniciar la formación. Desde dicho instante, se han registrado los siguientes 16 s, espacio temporal con el que se desarrolla todo el transitorio del sistema ($\tau \approx 3,2 \text{ s}$; $5\tau = 16,0 \text{ s}$). Los resultados obtenidos para cada uno de los parámetros indicados en la sección 4 se muestran en la Tabla 1. La representación gráfica de los resultados para los índices IEA e IEAT se muestra en las Figuras 5 y 6 respectivamente. En la Figura 7 se representa gráficamente a modo de ejemplo la respuesta temporal del error del sistema para cuatro de las experiencias más representativas realizadas. Los casos representados emplean una frecuencia de muestreo de 10 Hz y 1 Hz para el caso periódico y umbrales de 2 cm y 8 cm para la implementación basada en eventos.

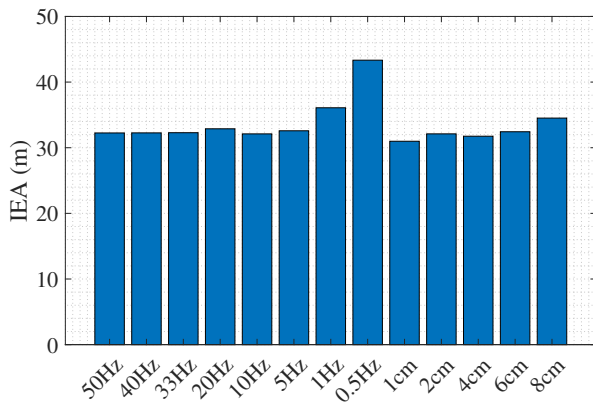


Figura 5: Integral del Error Absoluto, IEA.

Caso	t_e (s)	T (s)	IEA (m)	IEAT ($m \cdot s$)
50 Hz	7,72	0,020	32,3685	70,190
40 Hz	7,64	0,025	32,3794	69,6588
33 Hz	6,54	0,030	32,4426	73,8772
20 Hz	7,58	0,050	33,0321	74,3059
10 Hz	6,59	0,100	32,2444	72,0423
5 Hz	7,89	0,200	32,7242	74,6127
1 Hz	7,26	1,000	36,1560	87,5696
0,5 Hz	8,19	2,000	43,5029	114,3360
1 cm	6,98	0,097	31,1275	65,3618
2 cm	7,32	0,187	32,2453	72,0034
4 cm	8,48	0,408	31,8367	70,9773
6 cm	10,53	0,636	32,6532	75,9826
8 cm	10,27	0,925	34,7546	84,1132

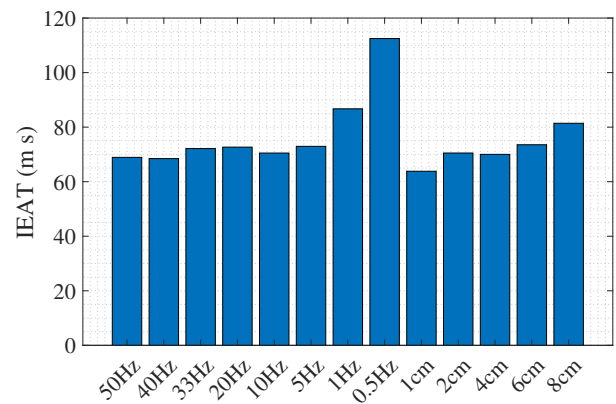


Figura 6: Integral del Error Absoluto ponderado en el Tiempo, IEAT.

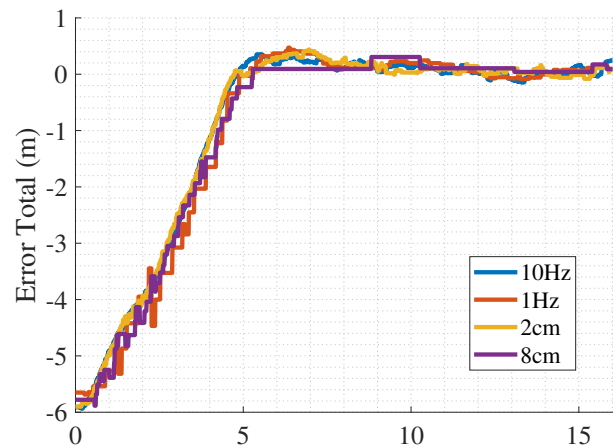


Figura 7: Respuesta temporal del error total en el movimiento de formación.

Los resultados obtenidos para los casos de muestreo periódico reflejan un comportamiento muy similar para frecuencias de muestreo superiores a 5 Hz. A partir de este valor, la respuesta tardía del controlador a causa del alto periodo entre señales de la realimentación provocan un incremento en los errores analizados (IEA: +11,15 % y +33,73 % para $T = 1 \text{ s}$ y $T = 2 \text{ s}$ respectivamente) y en el tiempo de establecimiento de la formación (+13,9 % para $T = 2 \text{ s}$). Por lo tanto, se puede concluir que el tiempo de muestreo adecuado de referencia

para el caso periódico es 0,1 s donde se encuentra el error mínimo y el menor tiempo de establecimiento. Este mismo tiempo de muestreo se tomará de referencia para determinar la mejora que supone la implementación basada en eventos. Para los casos que hacen uso del muestreo basado en eventos, se observa un comportamiento similar en la evolución de los errores respecto del muestreo periódico para umbrales inferiores a 6 cm. En estos casos, la respuesta del sistema para 1 cm de umbral es equivalente a la obtenida para una frecuencia periódica de 10 Hz con una ligera mejoría en los errores obtenidos. Con la finalidad de reducir el uso del canal de comunicación se analiza la respuesta obtenida para umbrales mayores para determinar el umbral que, sin deteriorar en exceso la respuesta del sistema, optimice la transmisión de señales. En este caso se observa que para umbrales de 2 cm y 4 cm los resultados obtenidos en los errores analizados se mantienen dentro de un margen aceptable ($\pm 5\%$). Para estos casos, la mejora obtenida en la transmisión de mensajes es del 46,7 % y 75,5 % respectivamente. Atendiendo al conjunto total de resultados, se considera que el umbral constante óptimo que mantiene un desempeño global similar al muestreo periódico con una reducción significativa del canal de comunicación es de 2 cm.

6. Conclusiones

En el presente trabajo se ha analizado la comunicación entre agentes en un SMR compuesto por cinco agentes que deben mantener una formación basada en distancias relativas mediante un protocolo de control de consenso. Se ha realizado un barrido de frecuencias para una implementación de muestreo periódico. A partir de los resultados obtenidos se ha determinado que la frecuencia periódica de referencia para el nivel de control colaborativo debe ser de 10 Hz. Una vez determinado este caso como referencia, se ha implementado un protocolo de muestreo basado en eventos con un umbral de disparo constante. Se ha llevado a cabo una secuencia de experiencias incrementando el valor de dicho umbral tomando como punto de partida un valor de 1 cm. Los resultados obtenidos han demostrado que un valor adecuado que mantiene el desempeño de la formación similar al caso de referencia de muestreo periódico es 2 cm. Con este umbral se consigue una reducción del 46,7 % en el uso del canal de comunicación entre agentes sin desviar el error de formación ni el tiempo de establecimiento del sistema en más de un 5 %. Estos resultados corroboran la utilidad del uso de implementaciones basadas en eventos para sistemas con limitaciones de recursos frente a las implementaciones clásicas periódicas.

Como futuras líneas de desarrollo, se trabajará en la mejora del desempeño de las leyes de control multi-robot para sistemas heterogéneos con distintas restricciones de movimiento y sensorización. Además, se explorará la extensión de las técnicas basadas en eventos a otras tareas propias de los SMR como

la navegación o fusión de sensores donde las limitaciones de los agentes pueden suponer un cuello de botella en su desarrollo.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo de la Agencia Estatal de Investigación (AEI) a través de los proyectos PID2020-112658RB-I00/AEI/10.13039/501100011033, 2021V/-TAJOV/001 e IEData 2016-6.

Referencias

- Amsters, R., Slaets, P., 2020. Turtlebot 3 as a robotics education platform. In: *Robotics in Education: Current Research and Innovations* 10. Springer, pp. 170–181.
- Cortés, J., Egerstedt, M., 2017. Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration* 10 (6), 495–503.
- Esmail, M. S., Merzban, M. H., Khalaf, A. A., Hamed, H. F., Hussein, A. I., 2022. Attitude and altitude nonlinear control regulation of a quadcopter using quaternion representation. *IEEE Access* 10, 5884–5894.
- Feng, Y., Rabbath, C. A., Rakheja, S., Su, C.-Y., 2015. Adaptive controller design for generic quadrotor aircraft platform subject to slung load. In: *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, pp. 1135–1139.
- Franklin, G. F., Powell, J. D., Emami-Naeini, A., Powell, J. D., 2005. *Feedback control of dynamic systems*. Vol. 4. Prentice hall Upper Saddle River.
- García-Valls, M., Dubey, A., Botti, V., 2018. Introducing the new paradigm of social dispersed computing: Applications, technologies and challenges. *Journal of Systems Architecture* 91, 83–102.
- Giernacki, W., Rao, J., Sladic, S., Bondyra, A., Retinger, M., Espinoza-Fraire, T., 2022. Dji tello quadrotor as a platform for research and education in mobile robotics and control engineering. In: *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, pp. 735–744.
- Giernacki, W., Skwierczyński, M., Witwicki, W., Wroński, P., Kozierski, P., 2017. 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)*. IEEE, pp. 37–42.
- Hasseni, S.-E.-I., Abdou, L., Glida, H.-E., 2021. Parameters tuning of a quadrotor pid controllers by using nature-inspired algorithms. *Evolutionary Intelligence* 14, 61–73.
- Heemels, W. P., Johansson, K. H., Tabuada, P., 2012. An introduction to event-triggered and self-triggered control. In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 3270–3285.
- Lee, E. A., 2008. Cyber physical systems: Design challenges. In: *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*. IEEE, pp. 363–369.
- Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W., 2022. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics* 7 (66), eabm6074. DOI: 10.1126/scirobotics.abm6074
- Mañas-Álvarez, F.-J., Guinaldo, M., Dormido, R., Dormido, S., 2023. Robotic park: Multi-agent platform for teaching control and robotics. *IEEE Access* 11, 34899–34911. DOI: 10.1109/ACCESS.2023.3264508
- Tharin, J., Lamberg, F., Carron, T., 2015. *Khepera iv user manual*. K-Team SA, Switzerland 700.
- Zanero, S., 2017. Cyber-physical systems. *Computer* 50 (4), 14–16.