

Computación frontera: influencia de latencias en la precisión de robots

Urbaniak, D.^a, Rosell, J.^a, Suárez, R.^{a,*}, Suppa, M.^b

^a*Institute of Industrial and Control Engineering (IOC), Universitat Politècnica de Catalunya (UPC), Av. Diagonal 647, 08028 Barcelona, España.*

^b*Roboception GmbH, Kaferstraße 2, 81241 Munich, Alemania.*

To cite this article: Urbaniak, D., Rosell, J., Suárez, R., Suppa, M., 2023. Edge computing: influence of latencies on the precision of robots.

XLIV Jornadas de Automática, 779-784 <https://doi.org/10.17979/spudc.9788497498609.779>

Resumen

La computación en la frontera (*edge computing*) permite acceder a grandes recursos computacionales y a redes de computadores, facilitando que pequeños dispositivos con recursos limitados realicen cálculos complejos descargándolos en computadores potentes de la red, y permite que la latencia en la comunicación se pueda reducir en comparación con la computación en la nube (*cloud computing*). Los robots móviles autónomos podrían beneficiarse enormemente de computación en la frontera, ya que en general su energía y recursos computacionales son limitados, pero su uso les permitiría realizar tareas computacionalmente complejas y que requieran adaptaciones rápidas a un entorno cambiante. En este trabajo se realiza una comparación del impacto de las tecnologías de comunicación más destacadas usando como tarea de ensayo, en simulación con Gazebo y ROS2, la corrección de la orientación de un cubo al ser empujado por un robot manipulador UR5e equipado con una pinza de dedos paralelos Robotiq 2F-85. Los resultados muestran un mejor rendimiento de 5G Ultra Reliable Low-Latency Communication (URLLC) sobre las comunicaciones 5G y 4G privadas, reduciendo la media del error de orientación en un 36 % y 55 %, respectivamente, tras haber realizado 50 acciones de corrección de la orientación con cada tecnología.

Palabras clave: Tecnología robótica, Percepción y medición, Robots manipuladores, Análisis estadístico de datos, Simulación de sistemas estocásticos, Redes de sensores y actuadores, Simulación con hardware-in-the-loop.

Edge computing: influence of latencies on the precision of robots

Abstract

Edge computing allows access to large computational resources and computer networks, making it easier for small devices with limited resources to perform complex computations by downloading them to powerful computers on the edge, and allows latency in communications to be reduced compared to cloud computing. Autonomous mobile robots could greatly benefit from edge computing, as their energy and computational resources are generally limited but its use would allow them to perform computationally complex tasks that require rapid adaptations to a changing environment. In this work, a comparison of the impact of the most outstanding communication technologies is carried out using as a test task, in simulation with Gazebo and ROS2, the correction of the orientation of a cube when being pushed by a robot manipulator UR5e equipped with a parallel-finger gripper Robotiq 2F-85. The results show a better performance of 5G Ultra Reliable Low-Latency Communication (URLLC) over private 5G and 4G communications, reducing the median orientation error by 36 % and 55 %, respectively, after having carried out 50 runs of corrective actions of the orientation with each technology.

Keywords: Robotics technology, Perception and sensing, Robots manipulators, Statistical data analysis, Simulation of stochastic systems, Networks of sensors and actuators, Hardware-in-the-loop simulation.

*Autor para correspondencia: raul.suarez@upc.edu
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

1. Introducción

La tarea de acomodar objetos en un espacio confinado optimizando su ubicación se denomina “empaquetamiento denso” (Wang and Hauser, 2021), y realizarla automáticamente conlleva serios desafíos que requieren sistemas de posicionamiento muy precisos que eviten colisiones con otros objetos o con el propio contenedor. En este contexto, el uso de robots para el posicionamiento final de un objeto empujándolo puede ser muy relevante (Shome et al., 2019). Para ello se debe realizar un monitoreo continuo mediante visión que permita la reacción y el ajuste del movimiento del robot. Por lo tanto, la latencia entre el cambio de pose del objeto y la reacción del robot puede ser crítica para la precisión de la tarea (ver Figura 1). Sin embargo, la detección y el seguimiento de objetos mediante visión exige, por lo general, un alto esfuerzo computacional y los largos tiempos de computación afectan la capacidad del robot para reaccionar rápidamente a los cambios producidos por su propia acción. En este contexto, la computación en la frontera, también denominada computación perimetral (en inglés *edge computing*), se presenta como una solución prometedora, particularmente para los robots móviles que no pueden portar computadores potentes.

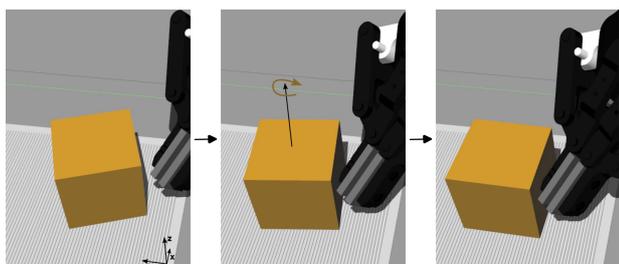


Figura 1: Configuración experimental para medir el impacto de las latencias en la precisión de la rotación de un cubo a través de una acción de empuje lineal. La orientación del objetivo está ilustrada por las líneas en la superficie de soporte. Una vez que la orientación del cubo alcanza la orientación deseada, el robot invierte su movimiento y retrocede. La precisión de la orientación final del cubo varía según la latencia de la comunicación y la velocidad de medición y control del sistema robótico.

En este trabajo se analiza el impacto de la latencia en la precisión de las acciones de empuje de un robot para aplicaciones de empaquetamiento denso con tres contribuciones principales:

- Comparación de varias tecnologías de comunicación (Ethernet, 5G privada, 5G URLLC privada, 4G privada, Wi-Fi 5 ideal, Wi-Fi 5 cargada) basada en las mediciones proporcionadas por Rodríguez et al. (2021b,a).
- Análisis del efecto de la estrategia de gestión de datos ante grandes valores de latencias, especialmente en el caso de redes Wi-Fi 5 cargadas.
- Identificación de la frecuencia necesaria para la medición y control como característica para evaluar la mejor tecnología de comunicación.

Tras esta introducción, en la Sección 2 se analizan trabajos previos relacionados con las latencias en la computación en la frontera. En la Sección 3 se explica el sistema de experimentación implementado, incluyendo los nodos ROS2 y la consideración de la latencia en las comunicaciones entre el sistema de

medición y el bucle de control de actuación. En la Sección 4 se describen y discuten resultados experimentales, remarcando el impacto de la latencia en los experimentos realizados. Finalmente, en la Sección 5 se dan unas conclusiones y se menciona el trabajo futuro de interés.

2. Trabajos previos

Para permitir la computación en la frontera se requiere una comunicación inalámbrica potente, y en este sentido la comunicación ultra-confiable de baja latencia (URLLC, de su nombre en inglés *Ultra Reliable Low-Latency Communication*) de las redes celulares 5G promete un alto potencial, tal como ya se muestra experimentalmente en los trabajos de investigación que se describen a continuación.

Voigtländer et al. (2017) utilizan un prototipo 5G para realizar una tarea de equilibrio de una bola usando computación en la frontera. Utilizan un Robot Móvil Autónomo (AMR, de su nombre en inglés *Autonomous Mobile Manipulator*) compuesto por un manipulador de 5 grados de libertad (GdL) y una base móvil holonómica de 3 GdL para mantener una esfera en el centro de una pantalla táctil resistiva. Los autores proponen una arquitectura de control distribuido en la que la pantalla táctil detecta la posición de la esfera y la envía de forma inalámbrica a un computador en la frontera, donde, mediante un controlador específico de alto nivel (no detallado), se calcula la posición cartesiana 3D deseada de la pantalla táctil necesaria para llevar a cabo la corrección de la posición de la esfera. Finalmente, mediante la cinemática inversa, esta posición cartesiana 3D se traduce al sistema de coordenadas de 8 GdL del robot y se envía de regreso al AMR donde se realiza el control de bajo nivel de los actuadores.

Raunholt et al. (2021) muestran que la planificación de la navegación y el control de aparcamiento de un AMR se pueden realizar de forma fiable utilizando la computación en la frontera y comunicaciones 5G. Con este fin, la planificación del movimiento se descarga en un computador en la frontera. Los autores comparan el impacto de la latencia de la comunicación privada 5G, 5G URLLC y Ethernet en la navegación y aparcamiento de robots móviles. En ambas tareas se muestra una disminución del rendimiento con la comunicación 5G, sin embargo, se mantiene en unos niveles aceptable para el funcionamiento confiable de los AMR.

Balogh et al. (2021) proponen un sistema en que el control de bajo nivel de un AMR se descarga en un servidor en la frontera mediante 5G para realizar navegación, y justifican el uso de la computación en la frontera para cálculos tan pequeños por la facilidad de mantenimiento y resiliencia mejorada frente a fallos de software y hardware de este servidor.

Nikhileswar et al. (2022) investigan las redes sensibles al tiempo (TSN de su nombre en inglés *Time-Sensitive Networking*) sobre 5G para los sistemas de control industrial que realizan continuamente detección, computación y actuación en lazo cerrado, y también usaron el control del equilibrio de una pelota sobre una pantalla táctil movida por un servomotor de 2 GdL con el fin de guiar la pelota a lo largo de un camino predefinido. Los datos del sensor de la pantalla táctil se envían/reciben a/desde un computador en la frontera a través de 5G. Para garantizar la correcta ejecución de la tarea se debe

tener un retardo máximo de 20 ms. El uso de TSN puede dar, en una red con mucho tráfico de datos, prioridad a las comunicaciones sensibles al tiempo frente a otras comunicaciones que no lo son, y, por lo tanto, lograr el requisito de latencia máxima requerido en comparación con una solución sin TSN.

Damigos et al. (2023) comparan el rendimiento de la computación en la frontera (5G), la computación en la nube (4G y 5G) y la computación a bordo, para el control de vehículos aéreos no tripulados. La solución de computación en la frontera 5G logra una latencia promedio entre 20 y 30 ms, que es significativamente más baja que las soluciones de computación en la nube (más de 100 ms en promedio). Esto genera un menor error de control, lo que permite realizar trayectorias más suaves y evitar con éxito obstáculos que las soluciones de computación en la nube no son capaces de evitar. En general, la computación a bordo logra un mejor rendimiento, pero la computación en la frontera 5G muestra un comportamiento bastante similar.

A diferencia de los trabajos anteriormente mencionados, en este trabajo se analiza el impacto de la computación en la frontera usando diferentes tecnologías de comunicación (Ethernet, 5G privada, URLLC 5G privada, 4G privada, Wi-Fi 5 ideal, Wi-Fi 5 cargada). Además, se considera el impacto de la latencia en el proceso de medición y control. Se espera que los resultados sean beneficiosos para desarrollar estrategias de orquestación de recursos.

3. Implementación

Los experimentos se han implementado con ROS2 Humble (Macenski et al., 2022) en un entorno de simulación Gazebo. La estructura de control de los experimentos se ilustra en la Figura 2. Consta de tres nodos: uno de detección de la pose, otro de control de alto nivel y el tercero de control de bajo nivel. La detección de la pose y el control de alto nivel se ejecutan en paralelo en el mismo *script* utilizando el ejecutor de subprocesos múltiples de ROS2, de este modo la detección de la pose se puede retrasar sin afectar el control del robot. El otro nodo se ejecuta individualmente y se comunica a través de servicio/cliente o publicador/suscriptor. A continuación, se explican con más detalle las partes principales.

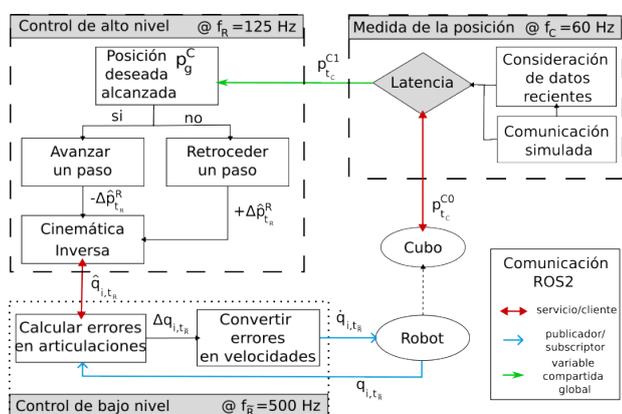


Figura 2: Estructura del lazo de control. Consta de tres nodos ROS2 y el entorno de simulación Gazebo con el robot manipulador y el cubo a manipular. Los nodos para el control de alto nivel y la detección de poses se ejecutan en paralelo utilizando el ejecutor de subprocesos múltiples de ROS2. Los tres nodos y la introducción de la latencia están resaltados en gris.

Medición de la pose del cubo. La pose real del cubo p^{C0} se detecta mediante el complemento *gazebo_ros_state* (definido en Gazebo) que ofrece un servicio *get_entity_state*. Este servicio se llama con una frecuencia $f_C = 60$ Hz (periodo de tiempo t_C). Se ha elegido esta frecuencia de medición para emular la velocidad típica de muestreo de las cámaras. Una vez recibida la respuesta con la pose actual, ésta se retrasa (efecto de la latencia) y se escribe en un parámetro global p^{C1} al que puede acceder el nodo de control de alto nivel.

Latencia. Como tecnologías de comunicación se han considerado Ethernet, 4G privada, 5G privada, 5G URLLC privada, Wi-Fi 5 ideal, Wi-Fi 5 cargada, y las correspondientes latencias se agregan en los experimentos en función de las distribuciones de probabilidad que el valor de latencia tiene en cada una de ellas (Rodríguez et al., 2021b,a), y que se ilustran en la Figura 3. Como valores de referencia se usan los de Ethernet. En comparación con Wi-Fi 5, las diversas tecnologías de telecomunicación tienen un menor rango de dispersión de la latencia, es decir sus posibles variaciones son menores, lo que hace que su comportamiento sea más confiable, especialmente cuando se considera más tráfico en la red; nótese que en el caso de Wi-Fi 5 cargada la latencia puede variar desde un par de milisegundos hasta casi un segundo. Por otro lado, en general, Wi-Fi 5 se comunica más rápido que 4G/5G privada. Los datos de 5G URLLC representan la latencia esperada de esta tecnología, ya que las redes URLLC aún no están ampliamente implementadas. Los datos de latencia de cada tecnología se generan de forma aleatoria al comienzo del experimento tomando muestras según su dispersión estadística, y se genera así un vector de valores a aplicar secuencialmente en cada iteración de la ejecución. A partir de este vector de latencias se puede determinar a priori cuándo una muestra va a llegar más tarde que la(s) muestra(s) siguiente(s) debido a la aparición de unas latencias significativamente mayores. Para ello, el tiempo en que se recibe la muestra correspondiente a cada iteración se contrasta con el tiempo requerido por la muestra siguiente, considerando para la muestra siguiente un retraso adicional debido al propio del periodo de muestreo. Cuando una muestra posterior llega al controlador antes que la anterior, ésta se elimina y se trabaja con la primera, obteniéndose así un conjunto de latencias *más recientes* (denominadas *freshes* en inglés) que indican los tiempos necesarios para recibir una nueva muestra cuando se descartan aquellas que llegan más tarde que una muestra anterior. La Tabla 1 muestra un ejemplo de latencias más recientes obtenidas a partir de una secuencia de latencias original para una red Wi-Fi 5 cargada, con $f_C = 60$ Hz (es decir, una muestra cada $16666 \mu s$). La primera columna da la secuencia de las latencias originales (obtenidas de forma aleatoria) que afectan el envío de cada muestra durante la tarea, en la segunda columna se dan las latencias de las muestras recibidas más recientes. Por ejemplo, la muestra en la línea 1 se recibe a los $37740 \mu s$ y la segunda se recibe un tiempo más tarde equivalente a un periodo de muestreo más la latencia propia de la segunda muestra, es decir, tras $16666 \mu s + 3820 \mu s = 20486 \mu s$, y por lo tanto llegará antes que la primera muestra, considerándose entonces como muestra más reciente y haciendo que la primera pase a ser ignorada. Para la experimentación, del vector de latencias que se considere en cada experimento (sea la secuencia original o las

más recientes) se toma una latencia L en cada llamada al servicio de medición de la pose del cubo configurada en la función `rc1cpp::sleep_for(L)` que retrasa la ejecución del hilo durante el valor temporal dado por L .

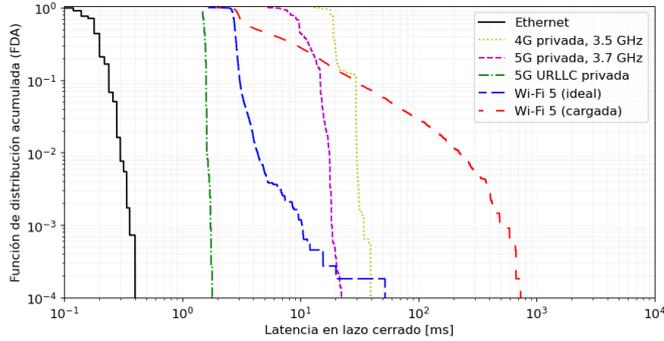


Figura 3: Funciones de distribución de las latencias generadas a partir de diferentes tecnologías de comunicación (Rodríguez et al., 2021b,a).

Tabla 1: Ejemplo de una secuencias de latencias y las correspondientes latencias más recientes en una red Wi-Fi 5 cargada con $f_c = 60$ Hz ($t_c = 16666\mu s$).

	Secuencia de latencias [μs]	Latencias más recientes [μs]
1	37740	20486
2	3820	
3	9460	9460
4	44160	31646
5	14980	
6	62300	38126
7	21460	
8	7000	7000
9	56200	19746
10	3080	
11	2640	2640
12	3000	3000

Control de alto nivel. El control de alto nivel es el encargado de tomar la decisión de avanzar o retroceder el robot manipulador a lo largo de una determinada dirección con un tamaño de paso constante previamente especificado. Se compara la pose del cubo retrasada por la latencia, p^{C1} , con el valor deseado p_g^C con una frecuencia f_R , correspondiente a intervalos de tiempo t_R . El robot manipulador comienza en una configuración inicial dada p_0^R y avanza un paso hacia adelante a $\hat{p}_{tr}^R = p_0^R + \delta\hat{p}$. La nueva pose del robot \hat{p}_{tr}^R se traduce a coordenadas del robot, $\hat{q}_{i,tr}$ para cada articulación i usando la cinemática inversa y se envía al controlador de bajo nivel mediante una “solicitud” de servicio ROS2 (que ofrece el propio controlador de bajo nivel). La pose inicial y la dirección del movimiento del robot se eligen de modo que se asegure que la pinza contacta con el cubo. La pose del cubo p_{tr}^{C1} se compara con p_g^C , y cuando el cubo alcanza o supera p_g^C el robot comienza a dar pasos hacia atrás.

Control de bajo nivel. El control de bajo nivel realiza un control de velocidad para suavizar el comportamiento del robot ante un cambio de dirección del movimiento. Recibe el valor deseado de cada articulación, lo compara con el correspondiente

valor actual, y calcula la velocidad de la articulación \dot{q}_i usando la diferencia Δq_i al cuadrado (en radianes) según

$$\dot{q}_i(\Delta q_i) = \begin{cases} -a\Delta q_i^2 & \text{if } \Delta q_i < 0 \\ a\Delta q_i^2 & \text{if } \Delta q_i \geq 0, \end{cases} \quad (1)$$

donde $a = 10000$ es un factor de escala que equilibra la capacidad de respuesta del control con la suavidad del movimiento, como se ilustra en la Figura 4 para una articulación. Las velocidades están limitadas a $\dot{q}_{max} = \pm\pi$ rad/s y pasadas (publicadas) al controlador de velocidad a $f_R = 500$ Hz. En este trabajo se utilizó el *JointGroupVelocityController* proporcionado en el paquete *ros2_control* (*ros2_control* Maintainers, 2023).

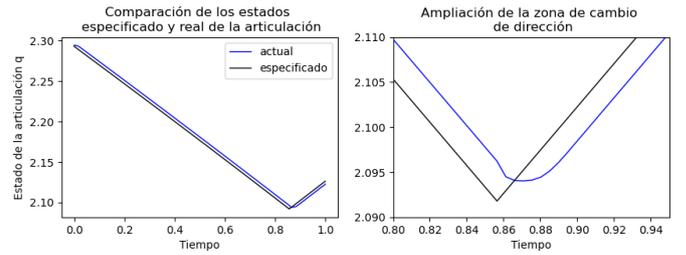


Figura 4: Comportamiento del control de velocidad: un cambio de dirección se suaviza a costa de la precisión del control.

4. Resultados experimentales

Para la validación experimental se usó el control del empuje de un cubo para realizar un empaquetamiento denso, como se ilustra en la Figura 5. El empuje del cubo de 0,05 m de lado se realiza simulando un robot manipulador UR5e equipado con una pinza de dedos paralelos Robotiq 2F-85. El experimento se realiza con cuatro combinaciones de parámetros, usando dos frecuencias de control de alto nivel (125 Hz y 500 Hz) y dos de medición con los sensores (60 Hz y 480 Hz), y el vector de latencias aleatoria se procesa para considerar las latencias que dan las muestras más recientes.

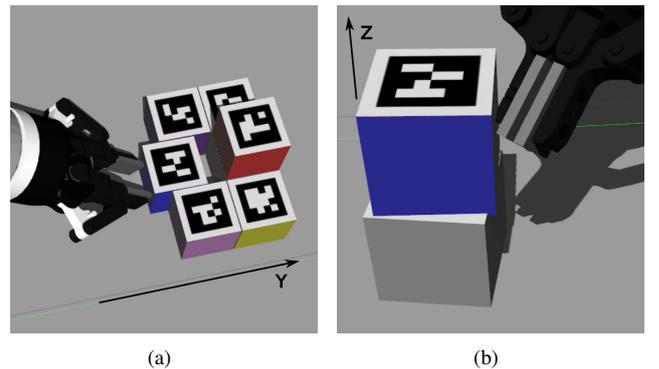


Figura 5: Dos ejemplos de acciones de empuje para el caso de empaquetamiento denso: a) otros objetos impiden una colocación directa, b) ajuste fino de la orientación de un objeto imprecisamente colocado.

4.1. Comparación de las tecnologías de comunicación

Al comenzar cada ejecución se especifica la tecnología de comunicación a ensayar (Ethernet, 4G privada, 5G privada, 5G URLLC privada, Wi-Fi 5 ideal, Wi-Fi 5 cargada) y se dan las latencias correspondientes, y también se fijan la pose de la pinza del robot (predefinida) p_0^R , la pose inicial del cubo p_0 con

una rotación inicial $\alpha_0 = 20^\circ$ alrededor del eje Z (ver Figura 5). Posteriormente, el robot se mueve hacia el cubo con incrementos Δs y lo empuja, para cambiar su orientación, hasta que se consigue la orientación objetivo deseada $\alpha_g = 0^\circ$ alrededor del eje Z. Para ello, el robot realiza un movimiento lineal en el plano XY con el punto de impacto sobre el cubo elegido arbitrariamente (tal que permita cambiar la orientación del cubo). El tamaño del paso de avance del robot se ha fijado en $\Delta s = (0,024 / f_R)$ m en cada periodo de tiempo t_R . Hacer que Δs dependa de f_R asegura la misma velocidad de movimiento del robot en los distintos experimentos.

Esta configuración experimental se corresponde con la ya mostrada en la Figura 1 y está diseñada para ayudar en la alineación de objetos en escenarios de empaquetamiento denso. En los experimentos se agrega ruido aleatorio acotado en un rango $[0, \Delta s]$ a la coordenada Y de la posición inicial y_0^R de la pinza del robot para crear perturbaciones con respecto al impacto inicial y, por lo tanto, a la respuesta del cubo al ser empujado.

Una vez que el control de alto nivel recibe $\alpha_{t_R} > \alpha_g$, se invierte la dirección del movimiento del robot durante $n_{t_R} = f_C/2$ pasos, y con esto se concluye una ejecución. Durante una ejecución, en cada periodo t_R se registra la orientación del cubo en el nodo de control de alto nivel. Una vez finalizada cada ejecución, las variables del experimento se restablecen a los valores p_0^R y p_0^C .

Se realizaron 50 ejecuciones para cada una de las 6 tecnologías de comunicación mencionadas y para cada uno de los cuatro casos considerados. En la Figura 6 se muestra la precisión final de la orientación del cubo obtenida en los diferentes casos. En general, se puede observar que las tecnologías con latencias más bajas (Ethernet, 5G URLLC) logran una mayor precisión que las tecnologías más lentas (4G y Wi-Fi 5 cargada). En el caso de los datos obtenidos con la secuencia de latencias original, Figura 6(a) y Figura 6(b), los gráficos de la red Wi-Fi 5 cargada han sido recortados debido a que con ella se alcanza un elevado valor de error ($-17,9^\circ$) causado por una muestra de latencia de 773 ms. Estos valores de error grandes pero poco frecuentes se pueden mitigar cuando se consideran las muestras de latencias más recientes, Figura 6(c) y Figura 6(d). Los datos de latencia de la peor muestra de Wi-Fi 5 cargada mostrados en la Figura 6(c) se corresponden con los mostrados en la Tabla 1. En este caso en particular, se extrajeron muestras de latencias grandes de forma consecutiva, de modo que la primera información sobre la pose recibida después de alcanzar la orientación deseada del cubo aun tenía un error de $-3,03^\circ$. Cuando se aumentan las frecuencias de control y de medición, Figura 6(b) y Figura 6(d)), se puede observar claramente que la desviación estándar disminuye significativamente para el caso de las tecnologías de comunicación más rápidas (en 5G URLLC pasa de 0,2419 a 0,0467), y que la mediana de su error también disminuye (en 5G URLLC: mínimo en $-0,209^\circ$, mediana con frecuencias bajas: $-0,751^\circ$, mediana con frecuencias altas: $-0,391^\circ$). La comparación de las diferentes tecnologías de telecomunicación y Wi-Fi en los cuatro escenarios presentados es particularmente interesante. La red Wi-Fi 5 cargada claramente se beneficia más del uso de las muestras más recientes y, dependiendo de las frecuencias de control y de medición del sistema, supera a las redes 4G e incluso a las 5G. Una frecuencia alta, por ejemplo 480 Hz (periodo de 2083 μs), también puede

mejorar situaciones en las que hay latencias grandes consecutivas, como, por ejemplo, dada la secuencia de latencias en la Tabla 1, en la que después de la muestra recibida en la línea 3, la siguiente en la secuencia no llegará hasta después de 31646 μs , mientras que la muestra de la línea 8 lo hará después de $7000 \mu s + 2083 \mu s * 5 = 17415 \mu s$. Finalmente, a frecuencias de control y de medición altas, se puede ver una pequeña diferencia de rendimiento entre Ethernet (mediana: $-0,34^\circ$) y 5G URLLC (mediana: $-0,39^\circ$), lo que sugiere que sólo en sistemas que realizan mediciones y acciones de control a altas frecuencias puede sacarse algún beneficio de la comunicación por cable, en caso de que el rendimiento esperado de 5G URLLC realmente llegue a obtenerse en su aplicación real.

5. Conclusiones y trabajo futuro

En este trabajo se ha mostrado experimentalmente que la latencia de la comunicación tiene un claro impacto en la precisión de un robot realizando una tarea de posicionamiento fino cuando se usa computación en la frontera. El rendimiento esperado de 5G URLLC no muestra una disminución significativa en la precisión en comparación con Ethernet, de hecho es equivalente para frecuencias de control de 125 Hz y de medición de 60 Hz. Además, al considerar las muestras de datos por suscripción, en lugar de requerir que todos los paquetes lleguen secuencialmente, el rendimiento de la red Wi-Fi 5 cargada aumenta mucho; conjuntamente con un sistema rápido de control (500 Hz) y de medición (480 Hz), supera a las redes privadas 4G y 5G. Como trabajo futuro, se considera la realización de simulaciones más avanzadas de las redes de comunicación que permitan tener en cuenta los diferentes tamaños de los paquetes de datos hacia y desde los computadores en la frontera. Finalmente, sería interesante considerar otra fuente importante de latencia como es la latencia de cómputo, que se genera al procesar la información de los sensores y en un control de alto nivel de alta complejidad.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Gobierno Español mediante el proyecto del Plan Nacional de I+D+I PID2020-114819GB-I00, y por European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska - Curie grant agreement No 956670.

Referencias

- Balogh, M., Vidács, A., Fehér, G., Maliosz, M., Horváth, M. Á., Reider, N., Rác, S., 2021. Cloud-controlled autonomous mobile robot platform. In: 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). IEEE, pp. 1-6.
- Damigos, G., Lindgren, T., Nikolakopoulos, G., 2023. Towards 5G edge computing for enabling autonomous aerial vehicles. IEEE Access.
- 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.
- Nikhileswar, K., Prabhu, K., Cavalcanti, D., Regev, A., 2022. Time-sensitive networking over 5G for industrial control systems. In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation. IEEE.
- Raunholt, T., Rodríguez, I., Mogensen, P., Larsen, M., 2021. Towards a 5G mobile edge cloud planner for autonomous mobile robots. In: 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall). IEEE, pp. 01-05.

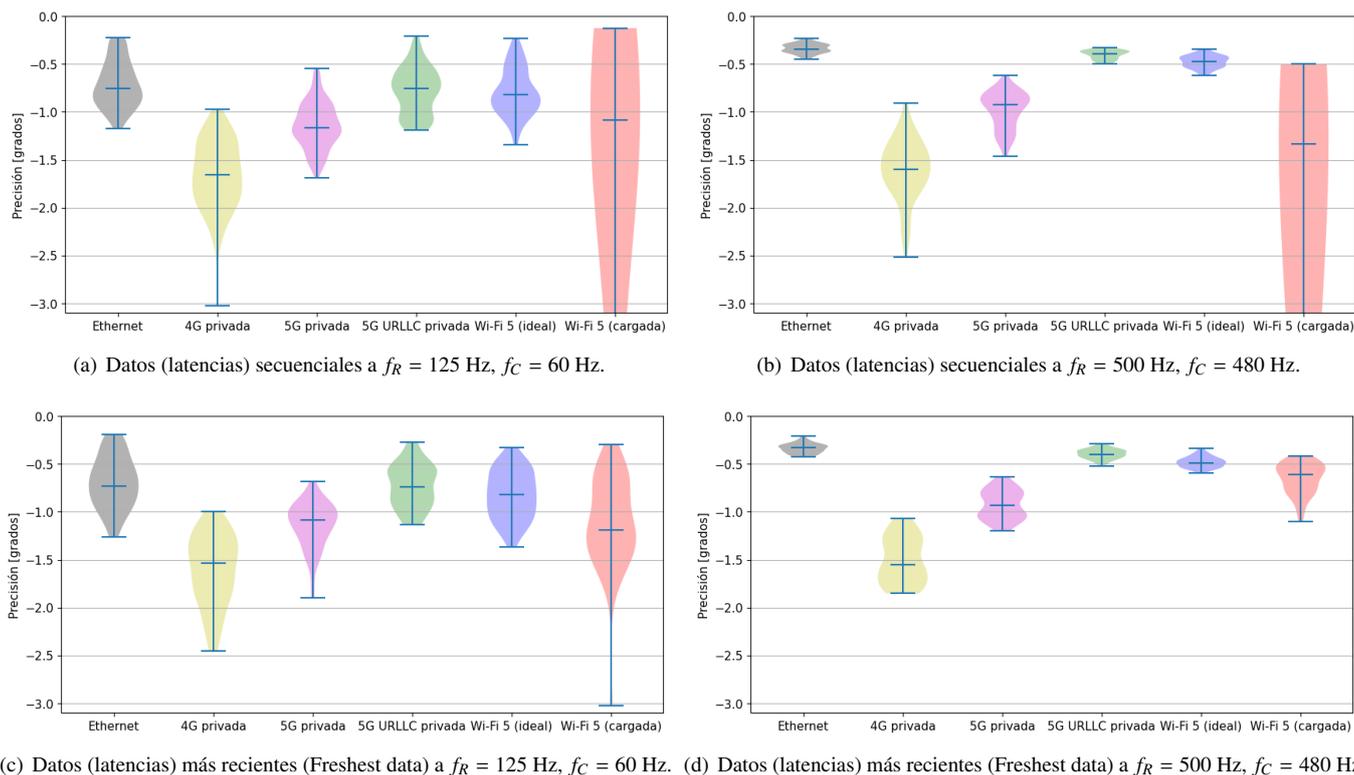


Figura 6: Impacto de las latencias en la precisión del posicionamiento del cubo para cada tecnologías de comunicación (se indica el error final de orientación del cubo). En gráficas (a) y (b), la representación de la red Wi-Fi 5 cargada aparece recortada debido a que alcanza un elevado valor de error de $-17,9^\circ$.

Rodriguez, I., Mogensen, R. S., Fink, A., Raunholt, T., Markussen, S., Christensen, P. H., Berardinelli, G., Mogensen, P., Schou, C., Madsen, O., 2021a. An experimental framework for 5g wireless system integration into industry 4.0 applications. *Energies* 14 (15), 4444.

Rodriguez, I., Mogensen, R. S., Schjørring, A., Razzaghpour, M., Maldonado, R., Berardinelli, G., Adeogun, R., Christensen, P. H., Mogensen, P., Madsen, O., et al., 2021b. 5G swarm production: Advanced industrial manufacturing concepts enabled by wireless automation. *IEEE Communications Magazine* 59 (1), 48–54.

ros2_control Maintainers, 2023. Welcome to the ros2_control documentation! available online: <https://control.ros.org/master/index.html>

Shome, R., Tang, W. N., Song, C., Mitash, C., Kourtev, H., Yu, J., Boularias, A., Bekris, K. E., 2019. Towards robust product packing with a minimalistic end-effector. In: 2019 International Conference on Robotics and Automation (ICRA). IEEE, pp. 9007–9013.

Voigtländer, F., Ramadan, A., Eichinger, J., Lenz, C., Pensky, D., Knoll, A., 2017. 5G for robotics: Ultra-low latency control of distributed robotic systems. In: 2017 International Symposium on Computer Science and Intelligent Controls (ISCSIC). IEEE, pp. 69–72.

Wang, F., Hauser, K., 2021. Dense robotic packing of irregular and novel 3d objects. *IEEE Transactions on Robotics* 38 (2), 1160–1173.