UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Comunicaciones

Desarrollo de un escenario de control remoto para brazos robóticos conectados con 5G

Trabajo Fin de Máster

Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones

AUTOR/A: Wang, Liangyue

Tutor/a: Sempere Paya, Víctor Miguel

Cotutor/a externo: Meseguer Valenzuela, Andrés

CURSO ACADÉMICO: 2023/2024



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Comunicaciones

Desarrollo de una plataforma de control remoto para brazos robóticos conectados con 5G

Tesis de Máster

Máster en Tecnologías, Sistemas y Redes de Telecomunicación

Autor: LIANGYUE WANG

Tutor: Víctor Miguel Sempere Paya

Primer cotutor: Andrés Meseguer Valenzuela

AÑO ACADÉMICO: 2023/2024

Objetivos — Este estudio pretende construir un sistema de control remoto de brazo robótico de alta precisión basado en la tecnología de comunicación 5G. Aprovechando al máximo la baja latencia y la alta capacidad de transmisión de datos de las redes 5G, el diseño de este sistema se dedica a lograr un control preciso del brazo robótico, y luego explorar el potencial de aplicación de la tecnología 5G en el campo de la automatización compleja y la operación remota. El objetivo central de la investigación es validar la estabilidad y fiabilidad de la plataforma de brazo robótico de control remoto basada en comunicación 5G, y comparar su rendimiento respecto a una comunicación Wi-Fi 5. El contenido específico de la investigación incluye.

Metodología — Los pasos seguidos para la elaboración de este trabajo son:

. Estudiar la tecnología de comunicación 5G y su aplicación en el control remoto.

. Estudiar la tecnología de reconocimiento de gestos del controlador Leap Motion y construir un entorno experimental.

. Estudiar y utilizar el software Unity para analizar los movimientos gestuales y desarrollar los algoritmos correspondientes.

. Configurar Raspberry Pi y su módulo de comunicación para lograr la conexión y el control con el brazo robótico.

. Diseñar y construir entornos de red 5G y Wi-Fi para garantizar que el entorno de prueba sea coherente en ambas condiciones de red.

. Desarrollar y probar el programa de control en la Raspberry Pi para garantizar que pudiera recibir y ejecutar comandos de Unity con precisión.

. Realización de pruebas de rendimiento en entornos de red 5G y Wi-Fi para recopilar datos de latencia, estabilidad y precisión, respectivamente.

. Comparar y analizar el rendimiento del sistema en ambos entornos de red para validar las ventajas de la tecnología 5G en el control remoto del brazo robótico.

Desarrollo de prototipos y trabajo de laboratorio — El desarrollo del código para esta investigación se llevó a cabo de acuerdo con las especificaciones de la tecnología de comunicación 5G para garantizar que cualquier usuario pudiera realizar pruebas con sus propios datos. Estas versiones de código abierto se produjeron tras múltiples pruebas y correcciones de las versiones existentes, y también incluyeron la evaluación e implementación de diferentes modelos nuevos. Mediante trabajo de laboratorio, el hardware y el software del sistema se integraron y optimizaron en múltiples rondas para garantizar su rendimiento y fiabilidad en condiciones de redes 5G y Wi-Fi.

Resultados— Realización de una serie de mejoras y optimizaciones en el sistema de brazo robótico de control remoto, en particular la evaluación de su rendimiento en redes 5G y Wi-Fi.Se han adaptado y optimizado los algoritmos de reconocimiento de gestos y análisis sintáctico de Unity de Leap Motion para que funcionen mejor con la Raspberry Pi y el brazo robótico.

Líneas futuras — Mientras que las conexiones Wi-Fi han mostrado mayores velocidades de transmisión y estabilidad en las pruebas, las conexiones 5G son capaces de ofrecer velocidades de descarga aún más rápidas en el entorno adecuado. De cara al futuro, la optimización continua de la infraestructura y la tecnología de las redes 5G para mejorar sus velocidades de transmisión y la estabilidad de las comunicaciones de enlace ascendente será una importante dirección de desarrollo. La conectividad 5G se beneficia de su potencial capacidad de descarga a alta velocidad, especialmente para escenarios de aplicaciones de alta demanda, como la descarga de archivos de gran tamaño y la transmisión de vídeo en alta definición, en los que se espera que la 5G ofrezca una experiencia de mejor calidad y una transferencia de datos más eficiente. Por lo tanto, seguir explorando y utilizando el potencial de la tecnología 5G, especialmente en los ámbitos del control remoto y la transmisión de datos exigentes, abrirá más posibilidades y oportunidades para el futuro desarrollo tecnológico.

Abstract — En este trabajo, se presenta el desarrollo de un sistema para el control remoto de un brazo robótico mediante el uso de tecnología de reconocimiento de gestos, que permite una interacción a distancia intuitiva y natural con un brazo robótico. Además, para garantizar una comunicación fluida y eficiente, el sistema emplea la tecnología 5G para la comunicación, conocida por su alta velocidad y baja latencia, y este enfoque garantiza una interacción fluida entre los comandos gestuales y el brazo robótico. Adicionalmente, se compara el comportamiento del escenario respecto a una red Wi-Fi 5.

Por lo tanto, se presentan los resultados de las pruebas realizadas en distintos escenarios para evaluar el rendimiento del sistema en términos de respuesta en tiempo real y precisión del control. Los resultados de las pruebas demuestran la viabilidad de un sistema preliminar que a futuro puede ser extrapolable en aplicaciones basadas en entornos industriales, proporcionando una nueva forma de interactuar con la robótica de manera eficiente y natural.

Autor: LiangYue Wang, email: lwang@teleco.upv.es

Director 1: Víctor Miguel Sempere Paya …, email: vsempere@dcom.upv.es

Director 2: Andrés Meseguer Valenzuela …, email: ameseguer@iti.es

Fecha de entrega: 17-06-2024

Índice

I. Introducción

Con la llegada de la Industria 4.0, cada vez se desarrollan más sistemas de control de equipos mecánicos para control remoto, lo que no solo reduce la carga de trabajo del personal de planta, sino que también ofrece una garantía para una producción segura. La maquinaria industrial y el equipo en la escena de operación pueden ser excavadoras, cargadoras, vehículos de construcción y otros equipos de operaciones de ingeniería a gran escala, también pueden ser robots, brazos robóticos y otros equipos de operación de maquinaria de precisión. En el caso de un brazo robótico, este elemento está situado como un componente clave de la producción industrial automatizada contemporánea, y se utiliza para una amplia gama de tareas, desde el montaje y el procesamiento en la fabricación hasta la cirugía en la atención sanitaria. Los comandos de control remoto llegan al controlador, que suele estar compuesto por PLC, microcontroladores y otros chips comunicados a través de bus industrial como puede ser CAN (Controller Area Network).

Sin embargo, los métodos tradicionales de control por cable son limitados cuando se enfrentan a nuevos retos. El control por cable está limitado cuando el operario o el controlador remoto se encuentran en una posición distinta a la del robot, lo que puede afectar a la maniobrabilidad en tiempo real del brazo robótico. Esta limitación ha impulsado a los investigadores a buscar soluciones innovadoras para mejorar la flexibilidad y aplicabilidad de los brazos robóticos, además de la adaptación de redes inalámbricas a este tipo de aplicación.

Con la aparición de la tecnología de comunicación móvil de quinta generación (5G), se ha producido un gran avance en la comunicación inalámbrica. Por un lado, la mejora de la tecnología de modulación de radio puede garantizar la fluidez y la transmisión en tiempo real de vídeo de alta definición y, por otro, la baja latencia puede garantizar la emisión en tiempo real de comandos interactivos entre los controladores y los equipos mecánicos in situ, reduciendo el retraso del enlace de comunicación entre las personas y los equipos, y hacer que la operación sea más precisa y cómoda.

En la automatización industrial, permitir el control remoto significa que los operadores pueden supervisar y operar brazos robóticos sin tener que estar físicamente presentes, y la naturaleza de baja latencia y gran velocidad de la conectividad 5G hace que la operación remota sea más fiable y en tiempo real, abriendo más posibilidades para una variedad de aplicaciones industriales. La importancia de esta investigación es combinar la tecnología 5G con el control de brazos robóticos para aumentar la eficiencia y la flexibilidad de la automatización industrial, al tiempo que se ofrecen soluciones más innovadoras en áreas como la sanidad.

La tecnología de control remoto en tiempo real se centra en el establecimiento de un bucle cerrado de información entre el controlador y el controlado. En el proceso de control remoto en tiempo real, el controlador envía las instrucciones de operación al elemento controlado a través de la red de comunicación, de modo que la persona encargada de la operación observa al mismo tiempo el comportamiento del sistema para realizar las modificaciones oportunas. En todo este proceso, la red de comunicación desempeña un papel crucial, por lo que esta debe tener características de transmisión de alta velocidad, baja latencia y alta fiabilidad para garantizar que la información pueda transmitirse a tiempo y llegar con precisión al destino. Además, la red de comunicación debe cumplir los requisitos de seguridad y estabilidad, como encriptación de la información, compresión de datos, tolerancia a fallos y corrección de errores, para garantizar el funcionamiento estable del sistema de control remoto y la seguridad de los datos. Por lo tanto, el diseño y la optimización de la red de comunicación son cruciales para la aplicación eficaz de la tecnología de control remoto en tiempo real.

El control remoto en tiempo real puede clasificarse en tres formas según la velocidad de movimiento de la persona controlada (punto fijo, velocidad lenta y media-alta) y los requisitos de tiempo real (bajo y alto):

- Maniobra de punto fijo: la maniobra de punto fijo de la persona controlada está en una posición fija, como la telemedicina, la automatización industrial, el hogar inteligente, la firma remota, los diferentes tipos de aplicaciones serán diferentes requisitos de tiempo real, de los cuales, con el fin de proteger la seguridad de la vida del paciente, la telemedicina tendrá los requisitos de tiempo real más altos.
- Manipulación lenta: la manipulación lenta de la persona controlada es a velocidad de marcha o velocidad de funcionamiento similar, por ejemplo, vuelos espaciales (se refiere principalmente a la manipulación remota de rovers en la luna y Marte), pilotaje remoto de naves no tripuladas, estos vehículos están funcionando a una velocidad lenta, los requisitos de tiempo real son menores.
- Maniobras a media y alta velocidad: Las maniobras a media y alta velocidad, en las que la persona controlada se encuentra a velocidades de funcionamiento de vehículos urbanos, es decir, entre 30 km/h y 120 km/h, se utilizan principalmente para pilotar a distancia vehículos, aperos de labranza mecánicos y para maniobrar drones. Los primeros requieren desplazarse por carretera, mientras que los dos últimos lo hacen con un mayor grado de libertad. Debido a la alta velocidad de funcionamiento de la persona controlada, el requisito de tiempo real es elevado.

La maniobra remota de brazos robóticos en líneas de producción es una aplicación típica para el control remoto en tiempo real, a menudo basada en la tecnología del Internet de las Cosas (IoT). Además, la conectividad 5G hace que los brazos robóticos sean adecuados también para cirugías médicas, respuesta a emergencias y otras áreas. En el ámbito médico, la conectividad 5G puede utilizarse para cirugías a distancia, lo que permite a los especialistas prestar atención médica a escala mundial. En la respuesta a emergencias, los brazos robóticos pueden manejarse a distancia para realizar tareas peligrosas, reduciendo así los riesgos.

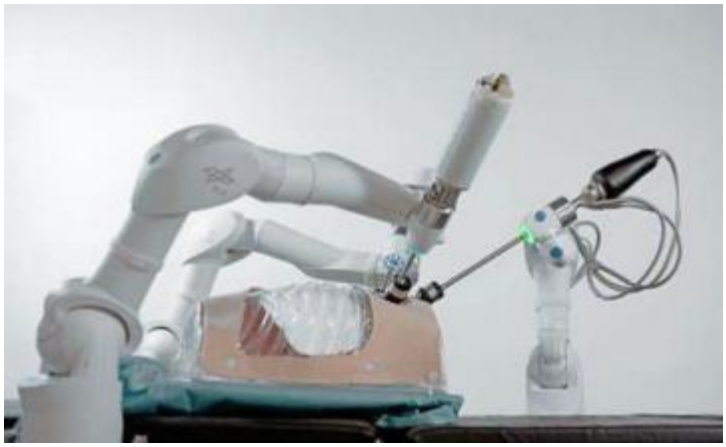


Fig. 1: Operación quirúrgica realizada remotamente mediante brazos robot.

Imagen de la web (puxiang.com)

Para permitir la manipulación remota de brazos robóticos, la red de comunicación debe proporcionar una latencia de comunicación extremadamente baja. La tecnología 5G puede proporcionar una latencia de comunicación incluso menor que la tecnología 4G, lo que la hace más adecuada para la manipulación remota. En una red 4G, la latencia de ida y vuelta de extremo a extremo suele ser de 20-50 milisegundos, mientras que un tiempo de reacción humano normal debería ser superior a 0,2 segundos, por lo que la latencia de la comunicación puede tener un impacto significativo en el tiempo de reacción humano.[1]

En este este, se detalla el proceso de diseño e implementación del sistema de control remoto del brazo robótico. En primer lugar, se desarrolla un sistema de control por gestos fiable utilizando tecnología avanzada de reconocimiento de gestos para lograr una interacción intuitiva y natural entre el usuario y el brazo robótico. En segundo lugar, se profundiza en cómo integrar eficazmente las tecnologías de comunicación 5G en el sistema para garantizar el tiempo real y la estabilidad. Esto implica trabajar en el diseño de protocolos de comunicación, la optimización de la transmisión de datos y la estabilidad de las conexiones de red. Por último, se prueba y evalúa el sistema en diferentes entornos (red 5G y Wi-Fi) para verificar su rendimiento en términos de respuesta en tiempo real y precisión del control. En conclusión, esta investigación de tesis de máster es de relevancia para la integración de la automatización industrial, el control remoto y las tecnologías 5G. Al desarrollar una

plataforma de control remoto que conecta un brazo robótico con 5G, proporcionará una solución más eficiente, segura y flexible para una variedad de aplicaciones, avanzando en la industria y la tecnología modernas. Esta investigación hará avanzar aún más el uso de brazos robóticos y tecnología 5G en diversos campos, allanando el camino para futuros desarrollos tecnológicos.

II. Objetivo

Este estudio pretende construir un sistema de control remoto de brazo robótico de alta precisión basado en la tecnología de comunicación 5G. Aprovechando al máximo la baja latencia y la alta capacidad de transmisión de datos de las redes 5G, el diseño de este sistema se dedica a lograr un control preciso del brazo robótico, y luego explorar el potencial de aplicación de la tecnología 5G en el campo de la automatización compleja y la operación remota. El objetivo central de la investigación es validar la estabilidad y fiabilidad de la plataforma de brazo robótico de control remoto basada en comunicación 5G, y comparar su rendimiento respecto a una comunicación Wi-Fi 5. El contenido específico de la investigación incluye:

- Algoritmo de reconocimiento de gestos: Se investiga y desarrolla un algoritmo avanzado de reconocimiento de gestos, utilizando las capacidades de diseño de interfaz gráfica y desarrollo de programas de Unity. Este algoritmo está diseñado para identificar de manera precisa y en tiempo real gestos complejos, con el objetivo de mejorar la velocidad de respuesta del sistema y la interacción intuitiva del usuario, proporcionando así una experiencia de control remoto más natural y flexible.

- Desarrollo de escenario de experimento de control remoto: Realizar el desarrollo de un experimento basado en control de un brazo mediante sensorización de gestos. Para ello, se emplea el reconocimiento testeado del objetivo anterior y se aplica a un caso de uso con un brazo robot.

- Validación del sistema de control remoto 5G y comparación con Wi-Fi: Una vez completada la implementación del sistema, se lleva a cabo una validación y prueba del sistema, incluyendo pruebas de estabilidad, velocidad de respuesta y precisión del sistema en un entorno inalámbrico. Se realiza una comparación del rendimiento en diferentes condiciones de red para evaluar su eficacia tanto con 5G como con Wi-Fi.

III. Metodología

En la fase preparatoria de este estudio, se llevó a cabo una revisión en profundidad de investigaciones punteras y estudios de casos sobre los últimos avances tecnológicos en 5G, las aplicaciones de control remoto de brazos robóticos y los beneficios potenciales de 5G en la mejora del rendimiento de los sistemas de control remoto. A través de este proceso, se recopiló y comprendió información sobre el posible impacto de las características de la red 5G, como el importante ancho de banda y la baja latencia, en la precisión del control y la velocidad de respuesta del brazo robótico. También se prestó atención a las aplicaciones de los brazos robóticos teledirigidos en distintos ámbitos, como la fabricación, la atención sanitaria, y a los requisitos específicos de rendimiento de red de estas aplicaciones.

A continuación, la investigación adoptó una estrategia de desarrollo por fases, empezando por las conexiones físicas más básicas y pasando gradualmente a soluciones de comunicación 5G más avanzadas. Inicialmente, se construyó un sencillo modelo de brazo robótico controlado mediante una conexión por cable con el objetivo de verificar la eficacia de la lógica de control y la respuesta mecánica subyacente. Tras pruebas y evaluaciones iniciales, el sistema pasó a controlarse de forma inalámbrica mediante Wi-Fi. El principal objetivo de esta fase era probar el rendimiento de la comunicación inalámbrica en términos de tiempo real y fiabilidad, y prepararse para la eventual adopción de la comunicación 5G. Por último, para analizar y evaluar las características de la red 5G, se construyó una arquitectura de red 5G de baja latencia y alta fiabilidad para el control remoto del brazo robótico. Se optimizó la ruta de transmisión de datos, además de reducirse la latencia de la señal mediante la parametrización de su configuración radio.

En términos de sensórica, se utilizó el dispositivo Leap Motion 2 para la captura de gestos de alta precisión, aprovechando además las potentes capacidades de simulación del entorno Unity para el control preciso del brazo robótico y poder mantener una interacción fluida con una interfaz de usuario. El concepto central de esta investigación es profundizar gradualmente mediante estudios empíricos, partiendo de tareas de control sencillas y aumentando la complejidad del sistema, incluida la introducción de tecnologías de detección avanzadas, algoritmos de control complejos y, en última instancia, la realización del control remoto del brazo robótico basado en la comunicación 5G. Este enfoque evolutivo paso a paso nos permite evaluar meticulosamente los retos técnicos y las soluciones en cada etapa, garantizando que todo el sistema se diseñe y realice sobre una base plenamente validada.

Además, también se prueban y evaluan sistemas que utilizan Wi-Fi, comparando el rendimiento de las tecnologías de comunicación Wi-Fi y 5G en términos de respuesta en tiempo real, latencia de la comunicación y estabilidad. Esta comparación nos permitirá comprender mejor las ventajas de la

tecnología 5G frente a la tecnología Wi-Fi tradicional y evaluar su eficacia práctica en aplicaciones de control remoto de brazos robóticos.

Fig. 2: Puesto de Trabajo

IV. Estado del arte

IV.1. Descripción general de la tecnología de reconocimiento de gestos

La investigación en la tecnología 5G proporciona mejoras en aplicaciones centradas en brazos robóticos y control remoto, haciendo que la interacción entre robots y humanos sea más natural e intuitiva. Entre ellas, la tecnología de reconocimiento de gestos es uno de los medios importantes para realizar dicha interacción.

El desarrollo de la tecnología de reconocimiento de gestos se remonta a los años sesenta. En aquella época, la tecnología de reconocimiento de gestos se basaba principalmente en principios ópticos, utilizando cámaras y otros equipos para captar los movimientos de las manos, y se procesaba mediante algoritmos informáticos para reconocer los gestos de las manos. Sin embargo, este método tiene muchos problemas, como las condiciones de iluminación, la postura de la mano y otros factores que influyen en los resultados del reconocimiento, y la precisión de este no es alta. [2]

Con el desarrollo de la tecnología de visión por ordenador, la tecnología de reconocimiento de gestos está pasando gradualmente de los métodos ópticos a los basados en el procesamiento de imágenes. Este método requiere el preprocesamiento de la imagen, como el filtrado, la detección de bordes, etc., y luego utiliza algoritmos de aprendizaje automático para entrenar el modelo y lograr el reconocimiento de gestos. En comparación con el método óptico, este método tiene una mayor precisión de reconocimiento, pero todavía hay algunos problemas, como una velocidad de reconocimiento lenta y un rango de reconocimiento limitado. [3]

Sin embargo, con el desarrollo de la tecnología de reconocimiento de gestos, el reconocimiento de gestos tiene un papel crucial en el diseño de interfaces hombre-máquina inteligentes y eficientes, y actualmente el reconocimiento de gestos se ha aplicado a diversos campos como el reconocimiento del lenguaje de signos, la monitorización inteligente y la realidad virtual. Con el progreso continuo de la tecnología, el reconocimiento de gestos irá madurando cada vez más, y sus aplicaciones serán cada vez más amplias. En el futuro, la tecnología de reconocimiento de gestos se convertirá en uno de los medios importantes para lograr la interacción entre el brazo robótico e Internet háptico, de modo que la interacción entre robots y personas sea más natural e intuitiva. [4]

IV.2. Descripción general de la tecnología de los brazos robóticos

Como una de las ramas importantes en el campo de la inteligencia artificial, la tecnología de control de brazos robóticos se ha desarrollado y aplicado rápidamente en los últimos años. El desarrollo de la tecnología de control de brazos robóticos se remonta a la década de 1950, y ha pasado por varias fases de desarrollo.

- **Primera etapa. Control mecánico**: El control mecánico es la primera etapa de la tecnología de control de brazos robóticos, y se basa principalmente en la estructura mecánica y los algoritmos de control para controlar el movimiento del brazo. La tecnología de control mecánico adopta principalmente el control de bucle abierto, es decir, la señal de control actúa directamente sobre la estructura mecánica para alcanzar el objetivo de control a través del movimiento de la estructura mecánica. La tecnología de control mecánico tiene las ventajas de una estructura sencilla y un bajo coste, pero la precisión del control no es alta y no puede adaptarse a entornos complejos. [5]

- **Segunda etapa. Control en bucle cerrado**: Con el desarrollo de la teoría de control, el control de bucle cerrado se ha convertido gradualmente en la corriente principal de la tecnología de control de brazos robóticos. El control de bucle cerrado adopta un algoritmo de control de realimentación, es decir, mide el estado de movimiento del brazo robótico y añade la señal de realimentación a la señal de control para controlar el movimiento del brazo robótico. La tecnología de control en bucle cerrado, comparada con la de control en bucle abierto, tiene una mayor precisión y estabilidad de control, pero el coste es mayor y requiere algoritmos de control complejos.

- **Tercera etapa. Control inteligente**: El control inteligente es la última etapa de la tecnología de control del brazo robótico, que utiliza principalmente la tecnología de inteligencia artificial, como el aprendizaje automático, el aprendizaje profundo, etc., para lograr el control del brazo robótico. La tecnología de control inteligente tiene un mayor grado de adaptación e inteligencia, puede adaptarse a entornos y tareas complejas, pero requiere una gran cantidad de datos y recursos informáticos, y el algoritmo es más complejo.

- **Cuarta Etapa. Control háptico**: El control háptico es una importante dirección de desarrollo de la tecnología de control de brazos robóticos, que puede hacer realidad la percepción del brazo robótico sobre el objeto y la retroalimentación táctil. La tecnología de control háptico utiliza sensores y algoritmos de control para lograr la percepción y el control del brazo robótico sobre el objeto, lo que puede mejorar la estabilidad y la precisión del brazo robótico, y también mejorar el grado de inteligencia del brazo robótico.

El desarrollo de la tecnología de control del brazo robótico ha pasado por cuatro etapas: control mecánico, control en bucle cerrado, control inteligente y control háptico, y cada etapa tiene sus ventajas y limitaciones. Con el desarrollo de la tecnología, la futura tecnología de control del brazo robótico será más inteligente y autónoma, y podrá adaptarse mejor a entornos y tareas complejos.

Con el desarrollo de la tecnología 5G, la aplicación del brazo robótico y de Internet háptico está cada vez más extendida. En el campo de la tecnología de control, las técnicas de control de brazos

robóticos son fundamentales para conseguir movimientos precisos y eficaces. A continuación, se enumeran tecnologías habituales para controlar brazos robóticos. [5]

Respecto a la tecnología de sensores, esta es una de las bases del control del brazo robótico. Los sensores pueden detectar el estado de movimiento del brazo robótico y la información del entorno externo, y convertir esta información en señales que puedan ser reconocidas por el brazo robótico. Las tecnologías de sensores más comunes son:

Sensores láser: pueden detectar la trayectoria y la posición del brazo robótico, con gran precisión, estabilidad y fiabilidad, entre otras ventajas.

- **Sensores hápticos**: pueden detectar la respuesta táctil del brazo robótico, para lograr un control y un movimiento más finos.
- **Sensor de cámara**: puede detectar la información del entorno que rodea al brazo robótico, como el color, la forma y el tamaño, para conseguir una navegación autónoma y evitar obstáculos, entre otras funciones.

Por otro lado, el algoritmo de control es el núcleo del control inteligente del brazo robótico. El algoritmo de control puede calcular la trayectoria de movimiento y las instrucciones de control del brazo robótico en función de la información recogida por el sensor, y transferirlas al actuador del brazo robótico para conseguir el movimiento y el funcionamiento del brazo robótico. Los algoritmos de control más comunes son:

- **Control difuso**: adecuado para tratar sistemas no lineales, inciertos y variables en el tiempo, con buena robustez y adaptabilidad.
- **Control de redes neuronales**: simula el mecanismo de funcionamiento de las neuronas en el cerebro humano, con un buen ajuste no lineal y capacidad de aprendizaje adaptativo.
- **Control por algoritmo genético**: simula el proceso de evolución biológica en la naturaleza, con buena capacidad de búsqueda global y optimización.

IV.3. Descripción general de la tecnología de comunicaciones 5G

Con el progreso continuo de la ciencia y la tecnología, la tecnología de la comunicación también se actualiza constantemente. Entre ellas, la tecnología de comunicación 5G, como nueva generación de tecnología de comunicación, se ha convertido en el centro de atención de la gente. ¿Cuál es la historia del desarrollo de la tecnología de comunicación 5G? [5]

El desarrollo de la tecnología de comunicación 5G se remonta a la década de 1980. En aquella época, para satisfacer las necesidades de las comunicaciones móviles, se empezaron a investigar nuevas tecnologías de comunicación. Entre ellas, se propuso OFDM, una tecnología multiportadora que puede mejorar eficazmente la eficiencia de las comunicaciones y la velocidad de transmisión, y que poco a poco se convirtió en la corriente principal.

Con el desarrollo continuo de la tecnología de comunicaciones móviles, la gente exige cada vez más a la tecnología de comunicaciones. Por ejemplo, se necesita mayor velocidad de transmisión, menor retardo, mayor fiabilidad y cobertura. Con el fin de satisfacer estas demandas, se ha comenzado a investigar una nueva generación de tecnología de comunicación, a saber, la tecnología de comunicación 5G.

En 2016, la Unión Internacional de Telecomunicaciones (UIT) dio a conocer su visión y sus requisitos para la 5G. La visión de la 5G es proporcionar servicios de comunicación de velocidad ultrarrápida, baja latencia, alta fiabilidad y capacidad para soportar una variedad de escenarios de aplicación. Los requisitos de la 5G incluyen velocidades de transmisión más altas, menor latencia, mayor fiabilidad y una cobertura más amplia.[6]

En 2017, la UIT aprobó formalmente el estándar 5G, que consta de dos partes, es decir, la nueva interfaz aérea 5G (NR) y la red central 5G (NGC). La nueva interfaz aérea 5G es una tecnología de comunicación inalámbrica basada en la tecnología OFDM, que puede proporcionar una mayor velocidad de transmisión y una menor latencia, mientras que la red central 5G es un nuevo tipo de arquitectura de red, que puede soportar más escenarios de aplicación y un mayor nivel de inteligencia. Nivel inteligente.

En 2019, la primera red 5G comercial del mundo se lanzó oficialmente en Shanghái (China). Esta red fue construida por China Mobile y puede proporcionar una velocidad de descarga de hasta 20 Gbps y una velocidad de subida de 1 Gbps. Además, la red 5G también tiene menor latencia y mayor fiabilidad, lo que puede soportar más escenarios de aplicación.

El desarrollo de la tecnología de comunicación 5G es un proceso iterativo y evolutivo. Desde la tecnología OFDM hasta los estándares 5G, se ha estado investigando y explorando continuamente para satisfacer las necesidades de comunicación en constante cambio. [7]

V. Diseño

V.1. Diseño de la demostración de control remoto

Con el fin de verificar la fluidez y la comparación de rendimiento del dispositivo de control remoto para controlar a distancia el brazo robótico en diferentes condiciones de red, se construye un demostrador de control remoto de brazo robótico conectado 5G para realizar pruebas de rendimiento exhaustivas, y se utilizan varios escenarios de red construidos por el laboratorio de pruebas para la verificación y comparación.

Los requisitos de red del robot de control remoto son principalmente dos aspectos del sistema de transmisión de imágenes y la transmisión de comandos de control remoto, que implica el uso del dispositivo Leap2 para recibir información en tiempo real del movimiento de la mano humana, y luego la información del movimiento de la mano es recopilada y procesada por la unidad en el lado del PC. Posteriormente, la información de control resultante se transmite en tiempo real a través de la red 5G o Wi-Fi a la Raspberry Pi para su reconocimiento y análisis, y el brazo robótico realiza las acciones correspondientes de acuerdo con los comandos. A continuación, el brazo robótico ejecuta la acción correspondiente según la instrucción.

En este sistema, el flujo de datos comienza con el dispositivo Leap2, que se conecta al ordenador mediante un cable USB. En el lado del ordenador, la información sobre el movimiento de la mano transmitida desde el dispositivo Leap2 es recibida y procesada por la plataforma de interacción basada en Unity. En este proceso, el dispositivo Leap Motion se considera el flujo ascendente y el ordenador el flujo descendente.

A continuación, la información procesada sobre el movimiento de la mano se transmite a través de uno de los dos esquemas de transmisión siguientes: un esquema 5G o un esquema Wi-Fi.

Esquema 5G: la información procesada se transmite al dispositivo central conectado al USRP B210 a través de un módem 5G. el módem 5G se comunica con la red 5G y transmite los datos al dispositivo central, donde el USRP B210 actúa como un dispositivo de radio definido por software que se encarga de recibir y transmitir señales de radio como estación base.

Opción Wi-Fi: otra opción es transmitir la información procesada a un router conectado a la Raspberry Pi a través de una red Wi-Fi. El router recibe los datos y los transmite a la Raspberry Pi, que actúa como centro de procesamiento y control de datos, y se encarga de recibir, analizar y ejecutar los comandos de control.

En última instancia, el brazo robótico realiza acciones basadas en las órdenes recibidas por la Raspberry Pi.Este diseño del sistema garantiza un flujo de datos fluido y una ejecución precisa de los comandos, al tiempo que define claramente la relación entre cada componente y la dirección del flujo de datos.

La estructura es la siguiente:

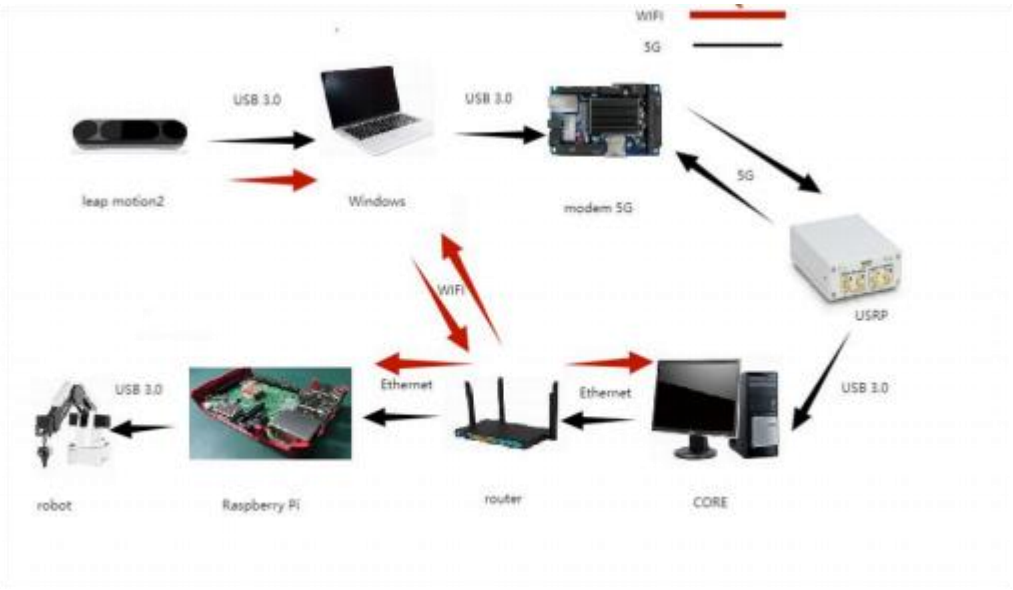


Fig. 3: Diagrama de flujo del diseño

V.2. Diseño de algoritmos de reconocimiento de gestos en Unity

Al diseñar el algoritmo de reconocimiento de gestos de Unity basado en Leap Motion, lo primero que hay que tener en cuenta es cómo capturar e interpretar eficazmente los movimientos gestuales del usuario. Leap Motion proporciona capacidades precisas de seguimiento de la mano, por lo que la idea central es utilizar los datos que proporciona para determinar el estado gestual del usuario. Utilizando las API proporcionadas por Leap Motion, se puede obtener la posición y postura de la mano en el espacio en tiempo real, que es la base para que el algoritmo reconozca el gesto. Utilizando la API proporcionada por Leap Motion, se puede obtener en tiempo real la posición y postura de la mano en el espacio, que es la base para que el algoritmo reconozca el gesto.

A continuación, el algoritmo debe analizar los datos de la mano para distinguir los distintos gestos. En mi diseño, el estado específico del gesto se determina calculando la distancia relativa y el cambio de dirección entre la mano y la posición anterior, combinados con las características del gesto (por ejemplo, la fuerza de agarre). Por ejemplo, comparando la distancia y la dirección del movimiento de la mano izquierda, el algoritmo puede determinar si el gesto se mueve hacia la izquierda, la derecha, arriba o abajo, lo que se basa en un conocimiento profundo de las características del movimiento de la mano y en un cálculo preciso.

Además, para que el reconocimiento de gestos sea más eficaz y preciso, se introduce en el diseño un mecanismo de gestión de estados. Este mecanismo consiste en supervisar en tiempo real los cambios de gesto y actualizar en consecuencia el estado del sistema. Este paso es clave para una interacción fluida y garantiza que el sistema sea capaz de responder rápidamente a los gestos del usuario y realizar las acciones correspondientes.

V.3. Diseño de algoritmos defuncionamiento del brazo robótico

Desarrollar un algoritmo eficiente de manipulación de brazos robóticos que permita controlar con precisión un brazo robótico para realizar tareas complejas. Para ello, se llevarán a cabo dos fases de investigación y desarrollo: una fase de pruebas preliminares con un modelo de brazo robótico sencillo para validar la funcionalidad básica y la eficiencia del algoritmo, seguida de una fase de investigación formal con un brazo robótico "Magician" de última generación para investigarlo en profundidad y aplicarlo.

Los algoritmos deben ser capaces de procesar y analizar los comandos de movimiento de la mano humana introducidos a través de la interfaz gráfica o los comandos del programa, y luego traducirlos en comandos de movimiento del brazo robótico. Al tiempo que garantizan un alto grado de precisión y capacidad de respuesta, los algoritmos deben tener en cuenta los distintos modelos y prestaciones del brazo robótico, y ser capaces de ajustar o afinar los parámetros para adaptarse a distintas configuraciones de hardware. Además, deben incluirse mecanismos de detección de fallos y respuesta de emergencia para garantizar la seguridad y estabilidad del proceso operativo.

- **Pruebas preliminares**:
  - **Selección y configuración de un brazo robótico simple**: seleccione un modelo de brazo robótico simple con funciones básicas y configure los sensores y actuadores necesarios.
  - **Desarrollo preliminar del algoritmo**: basándose en los parámetros de funcionamiento del brazo mecánico simple, desarrolle la versión inicial del algoritmo de funcionamiento. El algoritmo debe ser capaz de realizar las tareas básicas de agarre, desplazamiento y colocación.
  - **Pruebas funcionales y evaluación**: con la ayuda del software de control, pruebe la capacidad del algoritmo para controlar el brazo robótico y realizar las tareas predefinidas, y realice los ajustes y optimizaciones necesarios basándose en los resultados de las pruebas.
- **Construcción de equipos de investigación formal**:
  - **Selección del brazo robótico**: Adoptar el dobot magician de alto rendimiento, equipado con sensores de alta precisión y actuadores avanzados.
  - **Optimización y adaptación del algoritmo**: de acuerdo con las funciones avanzadas y los índices de rendimiento del brazo robótico "Magician", se optimiza y adapta en profundidad el algoritmo desarrollado inicialmente.
  - **Pruebas exhaustivas de rendimiento**: se comprueba el rendimiento de los algoritmos optimizados en escenarios de funcionamiento más complejos, incluidas tareas de funcionamiento en varios pasos, posicionamiento de precisión y funcionamiento estable a largo plazo.

V.4. Diseño de la red 5G

En lo referente a los requisitos de procesamiento para 5G-NR, estos son muy superiores a los de 4G, por lo que se requiere un PC o servidor de gama alta. Actualmente el ordenador dedicado a la ejecución del núcleo de red OAI dispone de:

- Intel Core i7 6900K (8 núcleos).
- 16GB DDR,
- 480GB SSD.

En cuanto al equipamiento dedicado al despliegue radio, se elige el USRP B210, que proporciona una plataforma totalmente integrada de Periférico Universal de Radio por Software (USRP™) de placa única con cobertura de frecuencia continua de 70 MHz a 6 GHz. Diseñada para la experimentación de bajo coste, combina el transceptor de conversión directa AD9361 RFIC para proporcionar hasta 56 MHz de ancho de banda en tiempo real, una FPGA Spartan6 abierta y reprogramable, y una rápida conectividad SuperSpeed USB 3.0, además de una cómoda alimentación por bus.

Fig. 4: USRP B210.

Respecto al dispositivo final, el RM500Q-AE de Quectel es un módulo M.2 diseñado para la banda 5G sub-6GHz, compatible con el estándar 3GPP Release 15 y optimizado para IoT industrial y comercial, así como para aplicaciones de enhanced Mobile BroadBand (eMBB). Estos módulos admiten los modos StandAlone (SA) y Non StandAlone (NSA) y proporcionan velocidades de enlace descendente de hasta 2,5 Gbps y de enlace ascendente de 900 Mbps.

Fig. 5: RM500Q-AE

VI. Desarrollo

VI.1. Desarrollo de la tecnología de reconocimiento de gestos

Fase preliminar: reconocimiento y localización inicial de gestos mediante Leap Motion 2 y su propio software Ultraleap Tracking. El objetivo de esta fase era verificar la precisión y fiabilidad del dispositivo Leap Motion en la captura de gestos complejos.

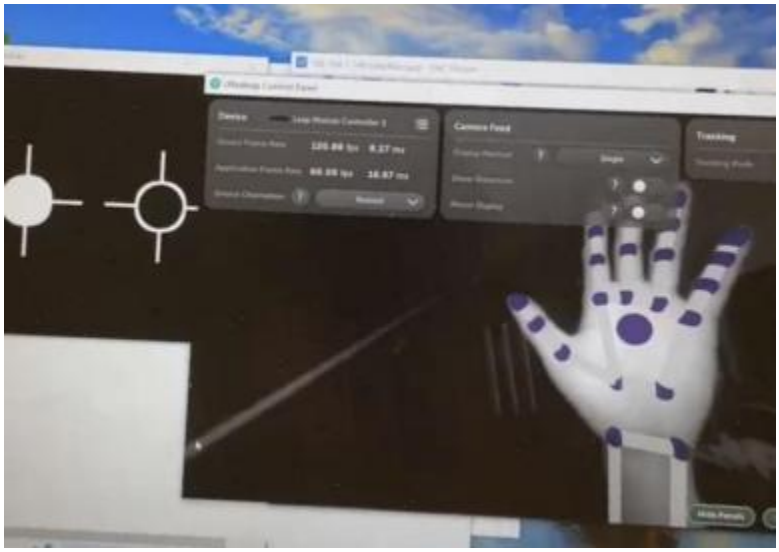


Fig. 6: Ultraleap Tracking

Módulo de reconocimiento de gestos: desarrollo de un módulo avanzado de reconocimiento de gestos para analizar los datos proporcionados por el software Ultraleap Tracking con el fin de reconocer movimientos gestuales más complejos, como mover las manos hacia arriba, abajo, izquierda y derecha. Utiliza las potentes capacidades de procesamiento gráfico de Unity y el SDK de Leap Motion para lograr una captura de alta precisión y una visualización 3D en tiempo real del reconocimiento de gestos.

Diseño de interfaz interactiva: Se diseñó una interfaz interactiva intuitiva de Unity que muestre información en tiempo real de la captura de gestos e información de estado del brazo robótico. Incluye las funciones básicas de funcionamiento del brazo robótico, como movimiento, rotación, agarre, etc.

Integración de comunicación de red: Se ha implementado un módulo de comunicación de red responsable de la transmisión en tiempo real de los comandos de control generados por la aplicación Unity al sistema de control del brazo robótico (por ejemplo, Raspberry Pi) a través de la red 5G o Wi-Fi. El diseño tiene en cuenta la latencia y la estabilidad de la red para garantizar señales de control precisas y en tiempo real.

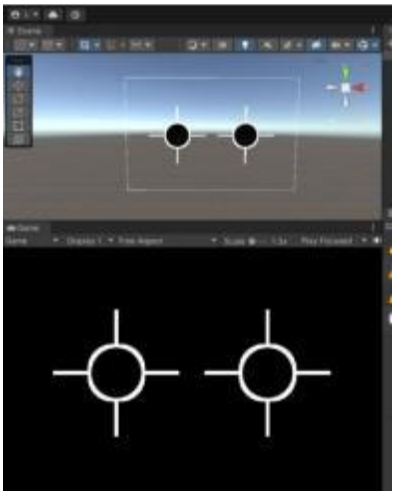


Fig. 7: Modelos de software Unity

El desarrollo específico del código respecto al control por gestos es el siguiente:

- **Definición de variables y procesamiento de valores de estado**: Definir palmLeft, palmRight, hL (mano izquierda), hR (mano derecha) para el seguimiento de la posición y el estado de la mano, utilizar LeapServiceProvider para obtener los datos de la mano. Comprobando el cambio de valor en el setter de stateValue y llamando a OnStateChange(), se implementa eficazmente el seguimiento del cambio de estado.

```
public class Manager : MonoBehaviour
{
    // Center points of both hands
    public Vector3 lastPosLeft;
    public Vector3 lastPosRight;

    public Transform palmLeft;
    public Transform palmRight;
    public Hand hL;
    public Hand hR;
    public LeapServiceProvider provider;
    public Text recieveText;

    public string _stateValue;
    public string stateValue
    {
        get { return _stateValue; }
        set
        {
            //Trigger countdown when the value changes
            if (_stateValue != value)
                OnStateChange();

            _stateValue = value;
        }
    }
    /// <summary>
    /// Whether it is in countdown mode
    /// </summary>
    bool isCounting;
    /// <summary>
    /// Whether sending is allowed
    /// </summary>
    bool canSend;
```

- **Establecer el estado de inicialización**: tempState = "0"; indica el estado inicial, suponiendo que no se detecta ninguna acción gestual. Obtener la mano izquierda y

derecha a través de provider.GetHand(Chirality.Left) y provider.GetHand(Chirality.Right).

```
void Update()
{

    //Default state
    tempState = "0";
    //Get information about the left and right hands
    hL = provider.GetHand(Chirality.Left);
    hR = provider.GetHand(Chirality.Right);
```

- **Manejo de gestos de zurdos y diestros**: Si se detecta una mano izquierda (hL! = null), entonces actualiza tempState de acuerdo a la distancia y dirección en que se movió la mano izquierda respecto a la posición anterior. Utiliza Mathf.Abs(x) y Mathf.Abs(y) para determinar si la dirección principal del movimiento de la mano es horizontal o vertical, y actualiza el estado según la dirección específica del movimiento. Si la mano izquierda cierra el puño (hL.GrabStrength == 1), el estado se actualiza a "j-1". Del mismo modo, se detecta la mano derecha y se actualiza el estado. Si la mano derecha cierra el puño, el estado se actualiza a "j-0".

```
// If left hand is detected
if (hL != null)
{
    //Left hand default state is "x"
    tempState = "x";
    //Calculate the distance from left hand to the center point
    Vector3 dis = palmLeft.position - lastPosLeft;
    float x = dis.x;
    float y = dis.z;
    float z = dis.y;

    //Switch state based on distance
    if (Mathf.Abs(x) > Mathf.Abs(y))
    {
        //L-right Left hand moves to the right
        if (x > 0.05f) tempState = "x-2";
        //L-left Left hand moves to the left
        else if (x < -0.05f) tempState = "x-1";
    }
    else
    {
        //L-up Left hand moves up
        if (y > 0.05f) tempState = "y-1";
        //L-down Left hand moves down
        else if (y < -0.05f) tempState = "y-2";
    }
    if (hL.GrabStrength == 1)
        tempState = "j-1";
}
```

- **Interacción UI**: Se utiliza una condición switch para cambiar el color y la posición de los dos objetos imagen imgLeft e imgRight basándose en diferentes valores de estado. Por ejemplo, cuando el stateValue es "x-1", la imagen izquierda (imgLeft) se mueve hacia la izquierda (la coordenada x es -100) y el color es negro, mientras que la imagen derecha (imgRight) permanece sin cambios.

```
void ShowUI()
{
    switch(stateValue)
    {
        case "0":
        case "j-0":
            imgLeft.color = Color.black;
            imgRight.color = Color.white;
            imgLeft.transform.localPosition = Vector3.zero;
            imgRight.transform.localPosition = Vector3.zero;
            break;
        case "x-1":
            imgLeft.color = Color.black;
            imgRight.color = Color.black;
            imgLeft.transform.localPosition = new Vector3(-100, 0, 0);
            imgRight.transform.localPosition = Vector3.zero;
            break;
        case "x-2":
            imgLeft.color = Color.black;
            imgRight.color = Color.black;
            imgLeft.transform.localPosition = new Vector3(100, 0, 0);
            imgRight.transform.localPosition = Vector3.zero;
            break;
```

- **Ajustes de inicialización**: En la fase Awake de Unity, primero se configuran los objetivos de la aplicación, como la velocidad de fotogramas y la resolución, lo que ayuda a optimizar el rendimiento y la apariencia del juego. Y se llama a tcp.ReadSetting() para cargar la información de configuración del servidor TCP.

```
public TcpClient tcp;
void Awake()
{
    Application.targetFrameRate = 30;
    Screen.SetResolution(1280, 960, false);
    tcp.ReadSetting();
}
Thread msgThread;
```

- **Lógica de Comunicación TCP**: El método SocketSend de la clase TcpClient es utilizado para enviar datos en el método SendMsg y el thread IESendMsg. Al manejar el envío de mensajes en un hilo separado o en una concatenación de hilos, se evita el bloqueo del hilo principal de Unity. El método SendMsg se ejecuta en un bucle infinito, comprobando la bandera canSend cada 500 milisegundos para determinar si se debe enviar el mensaje, y si canSend es verdadero, enviando el stateValue actual a través del cliente TCP. El hilo IESendMsg también se ejecuta en un bucle infinito, pero itera utilizando el enfoque de concatenación de Unity para minimizar el impacto en el hilo principal. Cada bucle espera un waitTime preestablecido (0,5 segundos), que equilibra la frecuencia de envío de mensajes con el rendimiento de la aplicación.

```
public void SendMsg()
{
    while (true)
    {
        if (canSend)
        {
            tcp.SocketSend("" + stateValue);
            Debug.Log(stateValue);
        }
        Thread.Sleep(500);
    }
}
WaitForSeconds waitTime = new WaitForSeconds(0.5f);
IEnumerator IESendMsg()
{
    while (true)
    {
        if (canSend)
        {
            tcp.SocketSend("" + stateValue);

            Debug.Log(stateValue);
        }
        yield return waitTime;
    }
}
```

- **Control del temporizador**: Utilice la variable temporizador para controlar la frecuencia con la que se envían los mensajes. Cada frame menos la diferencia de tiempo (Time.deltaTime), cuando el temporizador es menor que 0 y canSend es true, se envía el stateValue actual. Después de enviar el mensaje, reinicia el temporizador y comprueba si se ha recibido un mensaje del cliente TCP. Si lo hay, actualiza la casilla receiveText recieveText.

```
    timer -= Time.deltaTime;

    if (canSend && timer < 0)
    {
        timer = 0.5f;
        tcp.SocketSend("" + stateValue);
        if (!string.IsNullOrEmpty(tcp.recvStr))
        {
            recieveText.text = tcp.recvStr;
        }
        Debug.Log(stateValue);
    }
}
```

- **Respuesta a Cambio de Estado**: El método OnStateChange es llamado cuando el valor del estado cambia. Si el envío de mensajes no está permitido en ese momento (canSend es false), entonces todos los hilos se detienen y se inicia un nuevo hilo de cuenta atrás. El procedimiento IECountDown implementa una cuenta atrás de 1 segundo durante la cual el envío de mensajes está deshabilitado. Cuando finaliza la cuenta atrás, canSend se establece en true, permitiendo el envío de mensajes de nuevo.

```
public void OnStateChange()
{
    //If the state value changes during countdown, restart the countdown
    if (!canSend)
        StopAllCoroutines();

    Debug.Log("Value changed");
    StartCoroutine(IECountDown());
}
/// <summary>
/// Countdown function
/// </summary>
/// <returns></returns>
public IEnumerator IECountDown()
{
    //Do not allow sending until countdown is complete
    canSend = false;
    yield return new WaitForSeconds(1);//Countdown for 1 second

    canSend = true;
}
```

VI.2. Desarrollo de brazos robóticos:

Fase preliminar: Los experimentos iniciales utilizaron el brazo robótico Adeept 5-DOF, y llevaron a cabo la autoconstrucción general del brazo robótico y la configuración necesaria del sistema Raspberry Pi, en esta etapa, la atención se centra en la verificación de la estructura mecánica básica y la lógica de control.

Fig. 8: Robot Adeept 5-DOF

Partiendo del mando a distancia que viene con el brazo robótico, verificamos las funciones básicas y el rango de movimiento del brazo robótico, y realizamos inicialmente la función de control remoto a través del sistema Raspberry Pi y su propio sdk. Después, utilizamos el sistema de control remoto desarrollado por unity para hacer que el brazo robótico realice una serie de acciones con los comandos gestuales de la mano humana, y realizamos pruebas de control en entornos de red Wi-Fi y 5G sucesivamente para evaluar la influencia de la red en el rendimiento del control remoto. Como este brazo robótico utiliza motores paso a paso, el rango de rotación de las articulaciones está limitado a $180°$, lo que supone una limitación para probar determinados escenarios de funcionamiento muy flexibles.

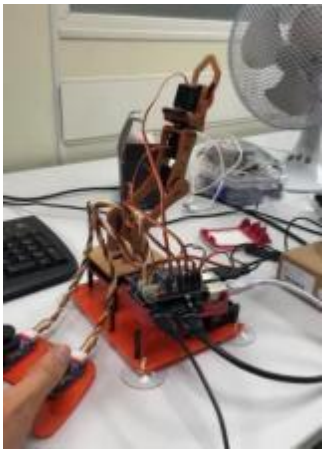


Fig. 9: Control del brazo robótico por control remoto

Fase de desarrollo: actualización al brazo robótico Magician con servomotores, cuyas articulaciones tienen un mayor rango de movimiento, lo que proporciona una capacidad de funcionamiento más flexible y una mayor precisión. Además, para facilitar la realización de pruebas detalladas y estandarizadas, se utilizó DobotStudio, el software de control que viene con el Brazo Mágico, para establecer un rango de prueba de movimiento específico para el Brazo Mágico con el fin de garantizar la coherencia y comparabilidad de las pruebas.

Fig. 10: Sistema DobotStudio

Además, el software de interacción desarrollado sobre la base de Unity se amplió y optimizó para las características del brazo robótico Magician, incluido el ajuste fino del control del servomotor y el reconocimiento de gestos más complejos. Se realizaron pruebas en profundidad del brazo robótico Magician en entornos de red Wi-Fi y 5G, con especial atención a la precisión y el tiempo de respuesta del control del servomotor, para evaluar el impacto de las diferentes condiciones de red en el rendimiento del sistema avanzado de control remoto.

Por último, se logró con éxito el objetivo de reconocer de forma remota y precisa los comandos gestuales de la mano humana y controlar el brazo robótico para realizar tareas específicas a través de redes Wi-Fi y 5G. El brazo robótico del mago realiza acciones precisas basadas en los comandos gestuales recibidos, como pellizcar un pequeño cuadrado y moverlo a una posición especificada demostrando un alto grado de precisión y capacidad de respuesta, y

verificando la estabilidad y fiabilidad en diferentes condiciones de red (especialmente el entorno de red 5G).

Fig. 11: Robot Dobot Magician

El desarrollo específico del código es el siguiente: Se empleó la API proporcionada por Dobot para interactuar con el brazo robótico mediante la carga de una biblioteca de vínculos dinámicos (DLL). Se seleccionó el puerto serie (COM3) y la velocidad de transmisión (115200) para la comunicación, y utilizó "CON_STR" para asignar el valor de enumeración del estado de conexión Dobot a una descripción de cadena más fácil de entender, y mostrar la información correspondiente de acuerdo con el código de estado devuelto (conexión correcta, dispositivo no encontrado, dispositivo ocupado).

```
import time
import socket
import DobotDllType as dType

CON_STR = {
    dType.DobotConnect.DobotConnect_NoError:  "Connection successful",
    dType.DobotConnect.DobotConnect_NotFound: "DobotConnect_NotFound",
    dType.DobotConnect.DobotConnect_Occupied: "DobotConnect_Occupied"
}

# Load Dll and get the CDLL object
api = dType.load()
# Connect Dobot
state = dType.ConnectDobot(api, "COM3", 115200)[0]
print("Device status:", CON_STR[state])
```

- **Configurar un servidor TCP/IP**: Utiliza la librería de sockets de Python para configurar un servidor TCP, enlazarlo a la dirección IP y puerto especificados, y comenzar a escuchar peticiones de conexión. Se implementa la función de comunicación de red, lo que permite al brazo robótico recibir comandos remotos.

```
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Bind IP address and port number
server_address = ("192.168.1.107", 1234)
server_socket.bind(server_address)

# Start the server and listen for connection requests
server_socket.listen(1)
print("Server is running, waiting for client connection...")

# Accept client connection
client_socket, client_address = server_socket.accept()
print("Client connected:", client_address)
```

- **Definir los parámetros de movimiento del brazo robótico**: Define el rango de movimiento y la posición inicial del brazo robótico en el código, establece el rango de movimiento mínimo y máximo para los ejes x, y, z, estos parámetros limitan el rango de movimiento del brazo robótico para evitar que sobrepase sus limitaciones físicas o entre en una zona insegura. También se definen una posición inicial (x_0, y_0, z_0) y una posición actual (x, y, z), además 'speed' establece la velocidad de movimiento del brazo robótico y 'lastIndex' se utiliza para rastrear el último comando enviado al brazo robótico. El 'lastIndex' se utiliza para realizar un seguimiento del índice del último comando enviado al brazo robótico, lo que resulta útil en la gestión de colas de comandos, especialmente cuando se requiere sincronización o monitorización del estado de ejecución del brazo robótico.

```
# Movement range
x_min = 197
x_max = 260
y_min = -38
y_max = 84
z_min = -29
z_max = 130
# Origin setting
x_0 = 217.0
y_0 = -0.71
z_0 = 100.3
x = 217.0
y = -0.71
z = 100.3
speed = 10.0
lastIndex = 0
```

- **Lógica de control del brazo robótico**: Se crea un bucle infinito utilizando while True, y cuando el cliente se conecta, el código entra en un bucle que espera y procesa los datos recibidos del cliente. Las coordenadas actuales del brazo robótico (X, Y, Z) se imprimen primero dentro del bucle, estas coordenadas se establecen al principio del bucle y se actualizan en las partes posteriores del bucle en función de los comandos recibidos. Los datos de hasta 1024 bytes se reciben del cliente utilizando 'client_socket.recv(1024).decode()' y se decodifican (por defecto en UTF-8). Finalmente, si los datos recibidos son nulos (no datos), se salta el bucle, lo que significa que se desconecta el cliente y se imprimen los datos recibidos, y luego el servidor envía un mensaje de reconocimiento al cliente.

```
# try:
while True:
    # pos = dType.GetPose(api)
    # pos =dType.GetPTPJointParams(api)
    # x = int(pos[4])
    # y = int(pos[5])
    # z = int(pos[6])
    # rHead = pos[7]
    print("x="+str(x)+"; y="+str(y)+"; z="+str(z))
    # Create socket object
    data = client_socket.recv(1024).decode()
    if not data:
        break
    print(data)
    response = "Server has received the message: " + data
    client_socket.sendall(response.encode())
```

- **Análisis y procesamiento de comandos**: Utilizando el modo de movimiento PTP (punto a punto), el código controla el movimiento del brazo robótico basándose en los comandos de datos recibidos. Con los datos recibidos, el código utiliza las funciones de la API de Dobot para controlar el movimiento del brazo robótico, por ejemplo, para mover la posición o controlar las ventosas y las mordazas de agarre. Para cada comando, el código añade los comandos de acción correspondientes a una cola y los ejecuta uno a uno en un bucle infinito, esperando y recibiendo los datos enviados por el cliente. Además, funciones como 'dType.SetPTPCmd' y 'dType.SetEndEffectorSuctionCup' se utilizan para actualizar el estado o la posición del brazo del robot, el La variable 'lastIndex' se utiliza para registrar el índice del último comando ejecutado.

En función de los datos recibidos, se ejecuta la acción correspondiente. Por ejemplo:

"j-0": abrir la ventosa.
"j-1": mover el brazo a la posición inicial.
"x-1"/"x-2": desplazamiento positivo/negativo a lo largo del eje x.
"y-1"/"y-2": desplazamiento en sentido positivo/negativo a lo largo del eje y.
"z-1"/"z-2": desplazarse en sentido positivo/negativo a lo largo del eje z.
"z-3"/"z-4": abre/cierra las mordazas.

```
# Async PTP Motion
# j1 -90~90, j2 0~90 j3 -15~60
if data == 'j-0':
    print('in j-0')
    dType.SetEndEffectorSuctionCup(api, 1, 1, isQueued=0)  # Open the pump
elif data == 'j-1':
    x = x_0
    y = y_0
    z = z_0

    lastIndex = dType.SetPTPCmd(api, 1, x, y, z, 0, isQueued=0)[0]  # Go to the origin
    continue
elif '0' in data:
    print('0')
    dType.SetEndEffectorSuctionCup(api, 1, 0, isQueued=0)  # Close the pump
elif 'x-1' in data:
    print('x xddd')
    if x < x_max:
        x += speed
        lastIndex = dType.SetPTPCmd(api, 2, x, y, z, 0, isQueued=0)[0]
        # print(x)
elif 'x-2' in data:
    if x > x_min:
        x -= speed
        lastIndex = dType.SetPTPCmd(api, 2, x, y, z, 0, isQueued=0)[0]
        # print(x)
```

- **Cola de comandos y gestión de excepciones**. ‘dType.SetQueuedCmdStartExec(api)': inicia la ejecución de comandos en la cola de comandos y utiliza un bucle while y 'dType.GetQueuedCmdCurrentIndex(api)' para comprobar el índice del comando que se está ejecutando actualmente y asegurarse de que se ejecutan todos los comandos. dType.SetQueuedCmdStopExec(api)': detiene la ejecución de la cola de comandos una vez que todos los comandos han finalizado. Mediante la cola de comandos, se pueden poner en cola múltiples comandos de acción para su ejecución. Este diseño tiene en cuenta la coherencia y suavidad de los movimientos del brazo robótico, así como los posibles conflictos de comandos, y puede controlar eficazmente los movimientos del brazo robótico para garantizar la precisión y seguridad de los movimientos. Al final del programa o cuando se produce un error, se cierran los sockets del cliente y del servidor y se desconecta la conexión con el brazo robótico.

```
    # Start to Execute Command Queue
    dType.SetQueuedCmdStartExec(api)

    # Wait for Executing Last Command
    #while lastIndex > dType.GetQueuedCmdCurrentIndex(api)[0]:
        # dType.dSleep(5)

    # Stop to Execute Command Queued
    dType.SetQueuedCmdStopExec(api)
    # print("

    # Stop executing commands
    # Stop to Execute Command Queued
    dType.SetQueuedCmdStopExec(api)
    # print("Device status: Execution has been stopped!")
# except:
#     print("Connection error")
client_socket.close()
server_socket.close()

# Disconnect
# Disconnect Dobot
dType.DisconnectDobot(api)
```

VI.3. Configuración de la red 5G.

- **Configurar CN 5G genérico**: configurar requisitos previos de OAI CN5G, archivo de configuración de OAI CN5G, extraer imagen docker de OAI CN5G, ejecutar OAI CN5G.
- **Tarjeta SIM**: Utilizar la aplicación del proyecto Open Cells uicc-v3.2 para programar la tarjeta SIM.
- **Configurar OAI gNB**: construir UHD desde el código fuente, construir OAI gNB, y ejecutar OAI gNB (elegir tipo USRP B210).
- **Ejecutar UE**: configurar RM500Q de Mobile Telecom, prueba de ping y realizar prueba iPerfde enlace descendente

VI.4. Evaluación del rendimiento y métodos de adquisición de datos de prueba

En esta prueba, realizamos una prueba de transmisión de datos 5G para un lado del brazo robótico, recibiendo paquetes de enlace ascendente de 10 Mb/s y paquetes de enlace

descendente de 30 Mb/s respectivamente para simular escenarios de alto tráfico o alta densidad de transmisión de datos, con el fin de evaluar la velocidad de respuesta y la estabilidad de la red en condiciones extremas. Por otra parte, para comparar el rendimiento en distintas condiciones de red, también realizamos pruebas de datos Wi-Fi, que incluían transmisiones de paquetes Wi-Fi de 50 Mb/s de enlace ascendente y descendente.

Durante la prueba, monitorizamos y registramos el tiempo de respuesta y la precisión del brazo robótico al realizar acciones basadas en comandos gestuales, y medimos y analizamos métricas como la cantidad de transmisión de datos recibida por segundo, el ancho de banda, el jitter y el número de paquetes perdidos/total de paquetes. Además, tuvimos en cuenta posibles interferencias inalámbricas y evaluamos la estabilidad y la tasa de pérdida de paquetes del sistema en un entorno con interferencias.

Para el retardo de paquetes ping de extremo a extremo, nuestro host CN5G estableció una longitud de paquete de 64 bytes y realizó una prueba ping en el TUE (equipo de usuario de prueba) de destino, y registró los valores máximos, mínimo y medio del retardo ping en 1 minuto. Teniendo en cuenta que el brazo robótico envía paquetes con la regularidad de decenas a cien bytes de datos cada 10 milisegundos, se puede utilizar el retardo del paquete ping (64 bytes) para estimar inicialmente el retardo del envío de la información de control.

En esta prueba, la información de la estación base (SA) 5G NR y del terminal 5G (equipo de usuario de prueba) se muestra en la Tabla 1.

Parámetros	Configuración
Tipo	5G NR TDD
Asignación de slots en Downlink, Uplink y Flexible	8D/4U/2FL
Periodo de muestreo	2.5 ms
Bandas de frecuencia	2.3GHz
Ancho de banda 5G	20MHz
Configuración de antenas de estación base	1Tx/1Rx

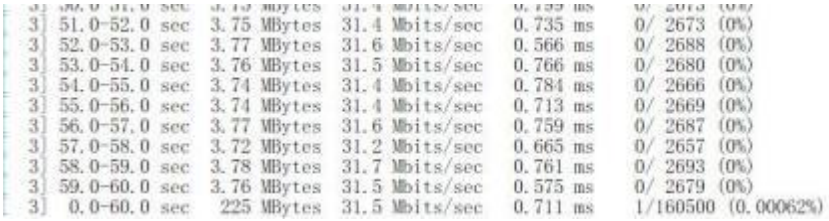
Tabla 1. Parámetros de los equipos de la red de pruebas 5G y del entorno de red

VI.5. Resultados y análisis de las pruebas

A través de las comparaciones de las pruebas, pueden hacerse las siguientes observaciones y conclusiones:

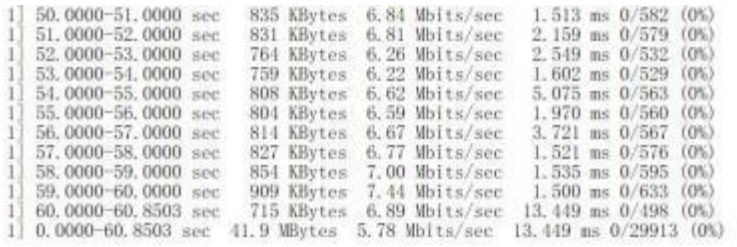
Descarga limitada a 30Mbps con 5G: Velocidad de transmisión de datos: la velocidad de transmisión fluctúa dentro de un rango, con una media de unos 31,5 Mbits/seg. Fluctuación: la fluctuación es baja, con una media de unos 0.711 ms, lo que indica un retardo relativamente

estable en la transmisión de datos. Pérdida de datos: la tasa de pérdida de datos es extremadamente baja, con sólo 1 paquete perdido, una tasa de pérdida del 0.00062%.



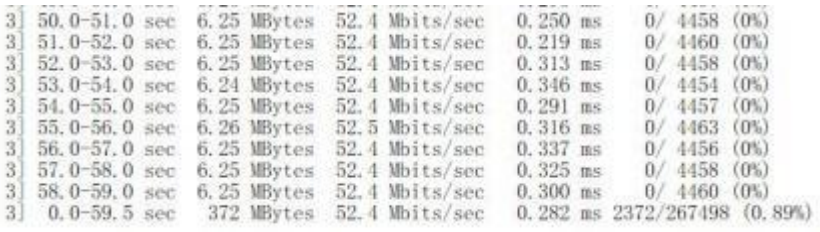
3]	51.0-52.0 sec	3.75 MBytes	31.4 Mbits/sec	0.735 ms	0/ 2673 (0%)
3]	52.0-53.0 sec	3.77 MBytes	31.6 Mbits/sec	0.566 ms	0/ 2688 (0%)
3]	53.0-54.0 sec	3.76 MBytes	31.5 Mbits/sec	0.766 ms	0/ 2680 (0%)
3]	54.0-55.0 sec	3.74 MBytes	31.4 Mbits/sec	0.784 ms	0/ 2666 (0%)
3]	55.0-56.0 sec	3.74 MBytes	31.4 Mbits/sec	0.713 ms	0/ 2669 (0%)
3]	56.0-57.0 sec	3.77 MBytes	31.6 Mbits/sec	0.759 ms	0/ 2687 (0%)
3]	57.0-58.0 sec	3.72 MBytes	31.2 Mbits/sec	0.665 ms	0/ 2657 (0%)
3]	58.0-59.0 sec	3.78 MBytes	31.7 Mbits/sec	0.761 ms	0/ 2693 (0%)
3]	59.0-60.0 sec	3.76 MBytes	31.5 Mbits/sec	0.575 ms	0/ 2679 (0%)
3]	0.0-60.0 sec	225 MBytes	31.5 Mbits/sec	0.711 ms	1/160500 (0.00062%)

Subida limitada a 10Mbps con 5G: Velocidad de transferencia de datos: la velocidad de transferencia fluctúa dentro de un rango, con una media de unos 5,78 Mbits/seg. Utilización del ancho de banda de la red: una vez más, al no facilitarse información específica sobre el ancho de banda, no es posible realizar un análisis preciso. Fluctuación de fase: la fluctuación de fase fue alto, con una media de 13,449 ms, lo que indica una latencia relativamente alta en la transmisión de datos. Pérdida de datos: no se produjo ninguna pérdida de datos, con una tasa de pérdida del 0%.



1]	50.0000-51.0000 sec	835 KBytes	6.84 Mbits/sec	1.513 ms	0/582 (0%)
1]	51.0000-52.0000 sec	831 KBytes	6.81 Mbits/sec	2.159 ms	0/579 (0%)
1]	52.0000-53.0000 sec	764 KBytes	6.26 Mbits/sec	2.549 ms	0/532 (0%)
1]	53.0000-54.0000 sec	759 KBytes	6.22 Mbits/sec	1.602 ms	0/529 (0%)
1]	54.0000-55.0000 sec	808 KBytes	6.62 Mbits/sec	5.075 ms	0/563 (0%)
1]	55.0000-56.0000 sec	804 KBytes	6.59 Mbits/sec	1.970 ms	0/560 (0%)
1]	56.0000-57.0000 sec	814 KBytes	6.67 Mbits/sec	3.721 ms	0/567 (0%)
1]	57.0000-58.0000 sec	827 KBytes	6.77 Mbits/sec	1.521 ms	0/576 (0%)
1]	58.0000-59.0000 sec	854 KBytes	7.00 Mbits/sec	1.535 ms	0/595 (0%)
1]	59.0000-60.0000 sec	909 KBytes	7.44 Mbits/sec	1.500 ms	0/633 (0%)
1]	60.0000-60.8503 sec	715 KBytes	6.89 Mbits/sec	13.449 ms	0/498 (0%)
1]	0.0000-60.8503 sec	41.9 MBytes	5.78 Mbits/sec	13.449 ms	0/29913 (0%)

Descarga limitada a 50Mbps con Wi-Fi: Velocidad de transmisión de datos: la velocidad de transmisión fluctúa dentro de un rango, con una media de unos 52.4 Mbits/seg. Jitter: el jitter es bajo, con una media de unos 0.3345 ms, lo que indica un retardo relativamente estable en la transmisión de datos. Pérdida de datos: se perdieron 2372 de 267498 datagramas, con una tasa de pérdida de alrededor del 0.89%.



3]	50.0-51.0 sec	6.25 MBytes	52.4 Mbits/sec	0.250 ms	0/ 4458 (0%)
3]	51.0-52.0 sec	6.25 MBytes	52.4 Mbits/sec	0.219 ms	0/ 4460 (0%)
3]	52.0-53.0 sec	6.25 MBytes	52.4 Mbits/sec	0.313 ms	0/ 4458 (0%)
3]	53.0-54.0 sec	6.24 MBytes	52.4 Mbits/sec	0.346 ms	0/ 4454 (0%)
3]	54.0-55.0 sec	6.25 MBytes	52.4 Mbits/sec	0.291 ms	0/ 4457 (0%)
3]	55.0-56.0 sec	6.26 MBytes	52.5 Mbits/sec	0.316 ms	0/ 4463 (0%)
3]	56.0-57.0 sec	6.25 MBytes	52.4 Mbits/sec	0.337 ms	0/ 4456 (0%)
3]	57.0-58.0 sec	6.25 MBytes	52.4 Mbits/sec	0.325 ms	0/ 4458 (0%)
3]	58.0-59.0 sec	6.25 MBytes	52.4 Mbits/sec	0.300 ms	0/ 4460 (0%)
3]	0.0-59.5 sec	372 MBytes	52.4 Mbits/sec	0.282 ms	2372/267498 (0.89%)

Subida limitada a 50Mbps con Wi-Fi: Velocidad de transmisión de datos: la velocidad de transmisión fluctúa dentro de un rango, con una media aproximada de 50.0 Mbits/seg. Jitter: la media es de unos 0.0315 ms. Pérdida de datos: menos pérdidas que en la prueba DL, con sólo 126 de 255094 datagramas perdidos, lo que supone aproximadamente un 0.049%.

1]	50.0000-51.0000 sec	6.10 MBytes	51.2 Mbits/sec	0.038 ms	0/4351 (0%)
1]	51.0000-52.0000 sec	5.96 MBytes	50.0 Mbits/sec	0.103 ms	85/4335 (2%)
1]	52.0000-53.0000 sec	5.25 MBytes	44.1 Mbits/sec	0.035 ms	41/3788 (1.1%)
1]	53.0000-54.0000 sec	6.65 MBytes	55.8 Mbits/sec	0.060 ms	0/4741 (0%)
1]	54.0000-55.0000 sec	5.97 MBytes	50.1 Mbits/sec	0.039 ms	0/4258 (0%)
1]	55.0000-56.0000 sec	5.97 MBytes	50.1 Mbits/sec	0.025 ms	0/4259 (0%)
1]	56.0000-57.0000 sec	5.95 MBytes	49.9 Mbits/sec	0.025 ms	0/4246 (0%)
1]	57.0000-58.0000 sec	5.95 MBytes	49.9 Mbits/sec	0.035 ms	0/4245 (0%)
1]	58.0000-59.0000 sec	5.94 MBytes	49.8 Mbits/sec	0.023 ms	0/4238 (0%)
1]	59.0000-60.0000 sec	6.00 MBytes	50.3 Mbits/sec	0.041 ms	0/4279 (0%)
1]	0.0000-60.0003 sec	357 MBytes	50.0 Mbits/sec	0.038 ms	126/255094 (0.049%)

Según los resultados de las pruebas, las conexiones Wi-Fi funcionan de manera más estable en términos de velocidades de transmisión. Además, durante la comunicación de enlace ascendente, la velocidad de transmisión de 5G puede ser limitada, lo que genera grandes fluctuaciones. Este fenómeno puede verse afectado por una variedad de factores, incluida la congestión de la red, la intensidad de la señal, el rendimiento del dispositivo, etc. En la comunicación de enlace descendente, 5G utiliza recursos de espectro más amplios y tecnología de modulación y demodulación más avanzada, por lo que puede lograr velocidades de transmisión más altas. Sin embargo, en la comunicación de enlace ascendente, esto puede deberse a factores como la capacidad limitada del canal o el diseño de la arquitectura de la red, lo que resulta en la pérdida de datos. la velocidad es menor. Además, es necesario considerar que la velocidad se ve limitada por el ancho de banda empleado, acotado a 20MHz en la banda n40.

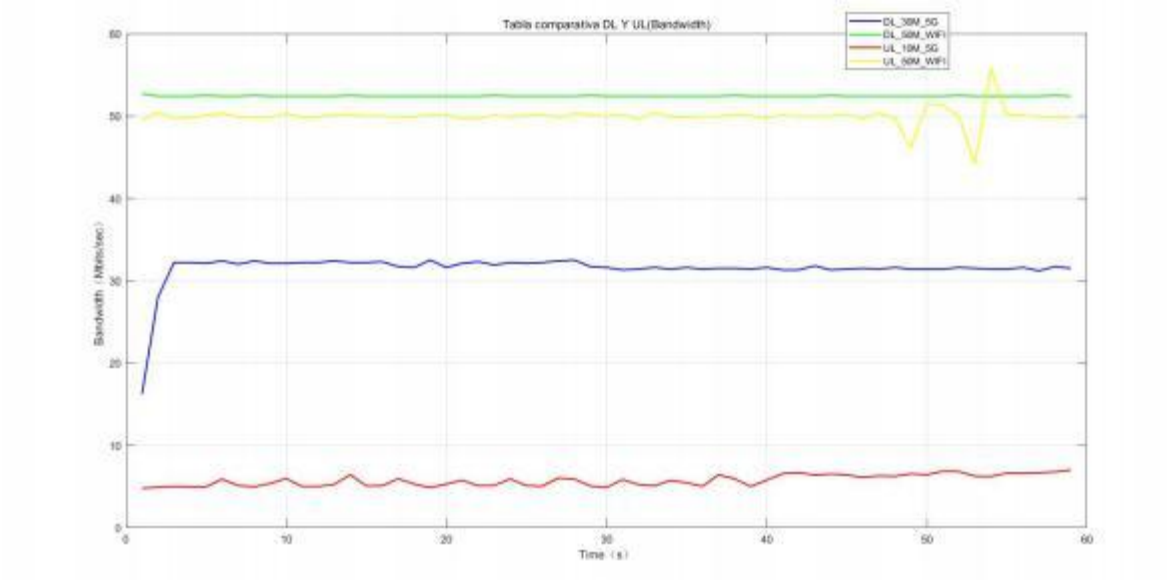


Fig. 12: Comparación del ancho de banda

En cuanto al jitter, la conexión Wi-Fi muestra un jitter bajo, lo que indica que el retardo en la transmisión de datos es relativamente estable. Sin embargo, cuando la transmisión de enlace ascendente 5G es de 10 Mbps, el jitter es mayor, lo que puede deberse a las interferencias del entorno (por ejemplo, otras señales inalámbricas, obstáculos físicos) que pueden provocar fluctuaciones en la calidad de la señal, causando así jitter. La estabilidad del jitter es uno de los indicadores más importantes de la calidad de la transmisión de datos y el rendimiento de la red. Un jitter variable significa menores fluctuaciones en el retardo de la transmisión de datos y una experiencia de usuario más fluida.

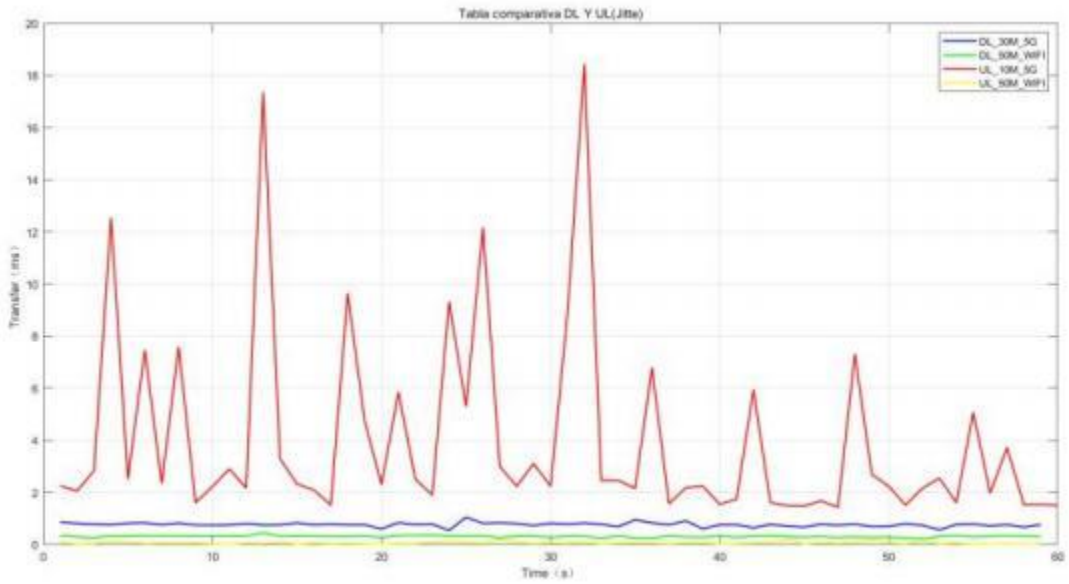


Fig. 13: Latencia comparativa (Transfer)

En términos de tasa de pérdida de paquetes, las conexiones 5G tienen tasas de pérdida de paquetes muy bajas, que son casi insignificantes. Esto puede atribuirse al concepto de diseño y la implementación técnica de la red 5G, incluyéndose un correcto despliegue de los componentes, asegurando calidad de la señal para transmitir y recibir. En comparación, la tasa de pérdida de paquetes de la conexión Wi-Fi es ligeramente mayor, pero aún dentro del rango aceptable, lo que puede estar relacionado con las características de la red Wi-Fi y la interferencia ambiental.

En resumen, si bien las conexiones Wi-Fi han mostrado tasas de transferencia y estabilidad más altas en las pruebas, las conexiones 5G estables, pero con velocidad limitada por el espectro disponible. Para el futuro, continuar optimizando la infraestructura y la tecnología de la red 5G para mejorar su velocidad de transmisión y la estabilidad en las comunicaciones de enlace ascendente será una dirección de desarrollo importante. La ventaja de la conectividad 5G radica en sus capacidades potenciales de descarga de alta velocidad, especialmente para aplicaciones de alta demanda, como descargas de archivos grandes y transmisión de video de alta definición, 5G podrá brindar una mejor experiencia y una transmisión de datos más eficiente.

V. Resumeny conclusiones

Esta investigación ha desarrollado un innovador sistema de control remoto basado en la tecnología 5G, utilizando Leap2 para la detección de gestos y el software Unity para el procesamiento eficiente de la información gestual.

La arquitectura del sistema integra módulos de comunicación 5G avanzados como RM500Q y USRP B210, optimizados para una baja latencia y una alta fiabilidad, lo que garantiza una transmisión casi instantánea de los comandos gestuales al brazo robótico. Además, la inclusión de código optimizado para la aplicación en la arquitectura del sistema desempeña un papel crucial en el procesamiento de datos, permitiendo cálculos y análisis en tiempo real cerca de la fuente de datos, acelerando la interacción entre el sensor y el robot.

Las pruebas han demostrado mejoras significativas en la precisión del control y la eficiencia de la comunicación, estableciendo un nuevo estándar para la interacción hombre-máquina en entornos críticos. Los resultados clave de este proyecto incluyen la latencia de comunicación 5G optimizada para el control remoto y la implementación exitosa del control de reconocimiento de gestos en tiempo real. Estas contribuciones suponen un avance en el campo del control remoto, abriendo nuevas posibilidades para su aplicación en los campos de la medicina, la fabricación y la domótica.

VI. Trabajo futuro

Con el envejecimiento de la población y otros factores que provocan un aumento continuo de los costes de mano de obra en todo el mundo y la escasez de mano de obra, la sustitución de la mano de obra tradicional por robots se ha convertido en una tendencia irrevocable. En la actualidad, los robots industriales se utilizan en gran número en la industria manufacturera, sobre todo en los talleres, donde pueden sustituir algunas tareas manuales repetitivas y ahorrar mano de obra.

Con el progreso de la tecnología, se han desarrollado diversos productos robóticos, lo que hace que los productos robóticos empiecen a expandirse a otras industrias e inspiren un enorme espacio de mercado. En algunas industrias tradicionales de riesgo, como el mantenimiento a distancia de tuberías subterráneas, el rescate a distancia en caso de corrimiento de tierras, el rescate en caso de incendio, el desminado y la protección contra explosiones, etc., el entorno operativo de estas industrias es duro y peligroso para los seres humanos, y en estos trabajos se pueden utilizar robots para realizar algunas tareas que no pueden realizar los seres humanos. Sin embargo, en algunas industrias de alto riesgo, cada uno se encontró con una situación compleja, no se puede reutilizar completamente el método de procesamiento original, no se puede utilizar un algoritmo inteligente unificado para controlar el robot para completar de forma independiente, la necesidad de control humano a distancia.

Si se utiliza la red celular inalámbrica Wi-Fi o 4G, el ancho de banda y la latencia de la red inalámbrica existente no pueden satisfacer de forma estable algunos escenarios de alta velocidad y baja latencia debido a la pobre anti-interferencia de la red Wi-Fi y la limitación de la velocidad de transmisión y la latencia de la red 4G. Estos cuellos de botella técnicos hacen que los robots colaborativos de control remoto encuentren dificultades en las aplicaciones industriales prácticas, y hasta ahora no han podido escalar el desarrollo y la implantación.

La red 5G, con gran ancho de banda y baja latencia, puede resolver estos cuellos de botella técnicos, la 5G está aportando nuevas oportunidades para el desarrollo industrial de los robots colaborativos teledirigidos, los robots teledirigidos pueden utilizarse en el futuro para la gestión de crisis y las industrias que realizan operaciones peligrosas, como amenazas de bomba, corrientes de alto voltaje, radiación, contaminación atmosférica, bacterias, campos magnéticos peligrosos, etc. La 5G proporciona redes de baja latencia y alta fiabilidad y las redes Wi-Fi pueden hacer que la latencia sea más estable en comparación con las redes Wi-Fi, y pueden hacer que la latencia sea más estable en comparación con las redes Wi-Fi. 5G proporciona redes de baja latencia y alta fiabilidad con un rendimiento de latencia mejorado en comparación con las redes Wi-Fi.

Sin embargo, la latencia de transmisión de la red es solo una parte de la latencia de extremo a extremo, junto con otros gastos generales de latencia de transmisión y procesamiento dentro del dispositivo. Sin equipos optimizados, las redes 5G solo pueden dar respuesta a una parte de las aplicaciones con sensibilidades de latencia en torno al segundo nivel, mientras que las sensibilidades de latencia por debajo del segundo nivel requieren una mayor optimización de los equipos y de los protocolos de transporte industrial. En el futuro, características de la 5G como la computación de borde y las redes segmentadas pueden utilizarse para optimizar aún más el rendimiento de la latencia de extremo a extremo de la red.

AGRADECIMIENTOS

He tenido la suerte de recibir mucha ayuda y apoyo valiosos durante la redacción de esta tesis de máster. Me gustaría expresar mi más sincero agradecimiento a todos aquellos que me han prestado su ayuda y aliento.

En primer lugar, me gustaría dar las gracias especialmente a mi supervisor, el profesor Víctor. El profesor Víctor Sempere no sólo me ha ayudado enormemente en el plano académico, sino que también me ha prestado mucha atención y orientación.

También me gustaría dar las gracias especialmente a mi supervisor, el investigador Andrés. A lo largo de todo el proceso de investigación, el profesor Andrés me guió pacientemente y me dio muchos consejos y ayuda valiosos, tanto en conocimientos teóricos como en funcionamiento práctico. Su cuidadosa orientación y su rica experiencia me han beneficiado enormemente en el proceso de investigación.

Me gustaría dar las gracias a todos los profesores, compañeros y amigos que me han prestado su apoyo y ayuda. Sin su ayuda, mi viaje académico no habría sido tan tranquilo.

Por último, me gustaría dar las gracias a mi familia por su comprensión y apoyo, es vuestro amor lo que me da fuerzas para seguir adelante.

BIBLIOGRAFÍA

[1] Sun, Yutong, (2019). Aplicaciones pioneras 5G: control remoto en tiempo real.

https://zhuanlan.zhihu.com/p/61356436

[2] Wu Xia, Zhang Qi, and Xu Yanxu. A review of the current development status of gesture recognition research. Electronic Science and Technology 6, 2013, 171-174.

[3] Rautaray, Siddharth S. Vision based hand gesture recognition for human computer interaction: a survey [J]. ARTIFICIAL INTELLIGENCE REVIEW, 2015, 43(1):1-54

[4] Yuan B, Cha Chen D. Current status and outlook of the development of gesture recognition technology. Science and Technology Innovation 32, 2018, 95-96.

[5] Guo, X., Xu, W., Tang, W.Q. and Wen, C. (2019) Research on Optimization of Static Gesture

Recognition Based on Convolution Neural Network. 2019 4th International Conference on Mechanical, Control and Computer Engineering(ICMCCE), Hohhot, 25-27 October 2019, 398-3982.

[6] CheriseBlues.(2021) [Beginner's Guide] The most complete Leap Motion technical summary on the web.https://blog.csdn.net/qq_41858463/article/details/118023776

[7] Developer documentation for Dobot Magician. Search for the Right Manual | Dobot Download Center (dobot-robots.com)

[8] D. I. Kosmopoulos, A. Doulamis and N. Doulamis, "Gesture-based video summarization", Image Processing 2005. ICIP 2005. IEEE International Conference on, vol. 3, pp. 111-1220-3, 2005.