Universidad Nacional Cuyo

Facultad de Ingeniería

Realidad Virtual

Proyecto Integrador:

Simulación de control de movimiento de Puma 560 mediante Realidad Aumentada

Fecha de entrega: 13/02/2023

Profesor:

Lic. Javier Rosenstein

Alumno:

Agustín Lezcano

Índice

Realidad Virtual

Simulación de control de movimiento de Puma 560 mediante Realidad Aumentada.

Agustín Lezcano

6 de marzo de 2023

Resumen

En este trabajo se desarrolló una aplicación de Realidad Aumentada utilizando el motor de videojuegos Unity, la plataforma de desarrollo Vuforia y el lenguaje de programación Python. La aplicación permitió generar un modelo virtual de un robot Puma de 6 grados de libertad mediante la interacción con el usuario a través de una interfaz cliente-servidor. La aplicación resultó ser una herramienta útil para visualizar y comprender el funcionamiento del robot Puma en un entorno de Realidad Aumentada.

1. Introducción

La Realidad Aumentada es una tecnología cada vez más popular y accesible, que permite añadir información digital a la realidad del usuario a través de dispositivos como smartphones o tablets. Esta tecnología ha encontrado aplicaciones en diversos campos, como el entretenimiento, la educación y la industria. En este trabajo se presenta una aplicación de Realidad Aumentada desarrollada con el motor de videojuegos Unity, la plataforma de desarrollo Vuforia y el lenguaje

de programación Python, que permite generar un modelo virtual de un robot Puma de 6 grados de libertad mediante la interacción con el usuario a través de una interfaz cliente-servidor.

2. Potenciales aplicaciones

Para dar una justificación a la creación de la aplicación, se pensó cuáles son los factores a tener en cuenta al instalar un robot en una planta industrial y cuáles son las soluciones que brindaría la misma a fines prácticos. Se pueden resumir los factores a tener en cuenta de la siguiente forma:

- Uno de los factores más importantes para la instalación de un robot en una planta industrial es asegurarse de que esté diseñado y programado adecuadamente para realizar las tareas que se le hayan asignado.
- La seguridad es otro factor importante a tener en cuenta al instalar un robot en una planta industrial, tanto para proteger al robot como a las personas que trabajan en la planta.
- También es importante asegurarse de que el robot esté integrado de manera eficiente en el flujo de trabajo de la planta y de que se realicen pruebas adecuadas para garantizar su buen funcionamiento.

Pensando en lo anteriormente mencionado, las posibles soluciones que se pueden generar son las siguientes:

- La realidad aumentada puede ser muy útil para la instalación de un robot en una planta industrial, ya que puede mostrar en tiempo real cómo se deben colocar y conectar los componentes del robot y proporcionar instrucciones en tiempo real sobre cómo programar y configurar el robot.
- La realidad aumentada puede ser muy útil en cuanto al espacio de trabajo del robot. Por ejemplo, una aplicación de realidad aumentada podría mostrar en tiempo real una representación

virtual del robot en su espacio de trabajo, lo que puede ayudar a visualizar cómo se desplazará el robot y cómo interactúa con su entorno. Esto puede ser especialmente útil para planificar el espacio de trabajo del robot de manera eficiente y evitar colisiones o interferencias con otros elementos de la planta. En general, la realidad aumentada puede ser una herramienta valiosa para optimizar el espacio de trabajo de un robot en una planta industrial.

3. Software utilizado

3.1. Unity

Unity es una plataforma de desarrollo de videojuegos y aplicaciones que permite a los desarrolladores crear proyectos en 2D y 3D utilizando diferentes lenguajes de programación. Incluye un editor de escenas y herramientas para facilitar el desarrollo, como sistemas de física y audio, gráficos en 3D y una biblioteca de assets y recursos. Unity se ha utilizado en una gran variedad de juegos y aplicaciones, tanto móviles como para consolas y PC. Algunos ejemplos de proyectos desarrollados con Unity incluyen Angry Birds 2, Temple Run, Pokemon Go, Ori and the Blind Forest, Slender: The Arrival y Firewatch. También se ha utilizado para desarrollar aplicaciones no relacionadas con los juegos, como Vuforia y Microsoft Flight Simulator. En resumen, Unity es un motor de videojuegos versátil y ampliamente utilizado para crear proyectos de alta calidad para diferentes plataformas.

Figura 1: Logo de Unity

3.1.1. Espacio de trabajo

Se utilizará la versión 2021.3.3f1, que es una de las últimas versiones estables del motor. El espacio de trabajo de Unity se compone de varias partes diferentes, cada una con un propósito

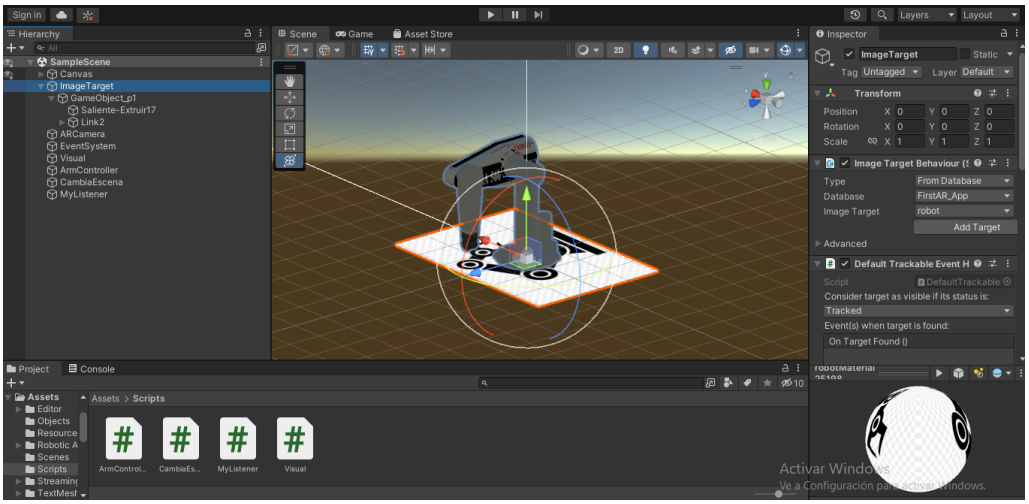


Figura 2: *Entorno y distintas ventanas en Unity.*

específico. Aquí hay una breve descripción de algunas de las partes más importantes del espacio de trabajo de Unity:

- La ventana del proyecto: esta ventana muestra todos los archivos y carpetas relacionados con el proyecto de Unity, y permite a los usuarios navegar y gestionar estos archivos y carpetas.
- La ventana del inspector: esta ventana muestra todas las propiedades y características de un elemento seleccionado en el juego, y permite a los usuarios modificar estas propiedades y características.
- La ventana de la escena: esta ventana muestra el mundo en el que se desarrolla el juego, y permite a los usuarios ver y manipular los elementos del juego.
- La ventana del game view: esta ventana muestra cómo se verá el juego cuando se juegue, y permite a los usuarios probar el juego mientras lo desarrollan.

- La ventana del menú: esta ventana contiene todas las opciones y comandos disponibles en Unity, y permite a los usuarios acceder a estas opciones y comandos para personalizar y controlar el desarrollo del juego.

3.2. Python

Python es un lenguaje de programación de propósito general y de alto nivel. Fue diseñado para ser fácil de leer y escribir, y cuenta con una amplia variedad de bibliotecas y frameworks que lo hacen adecuado para una amplia gama de usos, desde el desarrollo web hasta el análisis de datos y la ciencia de la computación.

Figura 3: Logo de Python

Una de las principales ventajas de Python es que su sintaxis es muy clara y concisa, lo que lo convierte en un lenguaje ideal para principiantes en la programación. Además, Python es un lenguaje de código abierto, lo que significa que es gratuito y puede ser utilizado y modificado por cualquier persona. Python es un lenguaje de programación de alto nivel y fácil de aprender. Se utiliza ampliamente en aplicaciones web, ciencia de datos, inteligencia artificial y más. Una de las características más útiles de Python es su capacidad para trabajar con sockets TCP/IP, lo que le permite enviar y recibir datos a través de una red de manera fácil y rápida. Esto lo convierte en una excelente opción para desarrollar aplicaciones de red y aprovechar al máximo la capacidad de comunicación de una red.

3.3. Vuforia

Vuforia es una plataforma de desarrollo de realidad aumentada creada por la empresa PTC. La plataforma ofrece herramientas y tecnologías para crear aplicaciones de realidad aumentada que pueden ser utilizadas en una variedad de dispositivos, como smartphones y tablets.

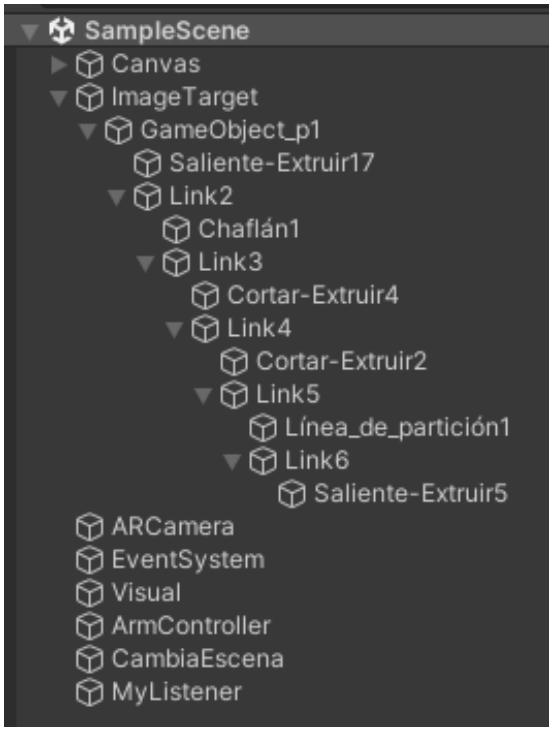
Vuforia utiliza tecnologías de visión por computadora para reconocer objetos y superficies en el entorno y sobreponer información virtual sobre ellos. La plataforma permite a los desarrolladores crear aplicaciones de realidad aumentada de alta calidad y ofrece una amplia gama de características y herramientas para crear contenido y experiencias de realidad aumentada personalizadas.La versión utilizada es la 9.8.13, que no es la última estable (se ha lanzado la versión 10.12 pero la misma no cuenta con soporte para versiones de Android menores a la versión 8-Oreo) por lo tanto, para darle un rango más grande de dispositivos soportados, se optó por la versión anterior.

Figura 4: Logo de Vuforia

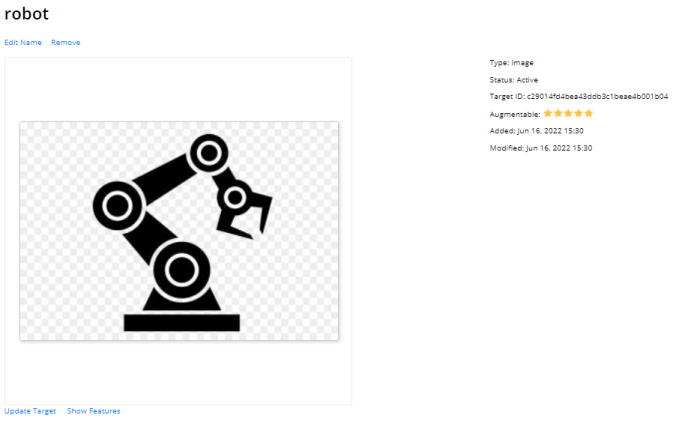
3.3.1. Integración con el entorno Unity

Los Target en Vuforia son imágenes o patrones que pueden ser reconocidos por una aplicación de realidad aumentada desarrollada con la plataforma de Vuforia. Para utilizar los Target, es necesario cargar las imágenes o patrones en la plataforma de Vuforia y luego utilizarlos en una aplicación de realidad aumentada. Cuando se ejecuta la aplicación y se visualiza un Target a través de la cámara de un dispositivo móvil, la aplicación detecta el Target y superpone contenido virtual en él para crear una experiencia de realidad aumentada. Los Target son detectados mediante el uso de algoritmos de reconocimiento de patrones. En la figura 5b se puede observar como se ve uno de los target desde la página web de Vuforia. Se debe tener licencia y credenciales y se puede tener más de un target a la vez. Se suelen agregar los mismos a bases de datos.

En Unity, se debe instalar la herramienta Vuforia como un complemento para poder utilizar la imagen Target. Esto se hace descargando un archivo ÜnityPackage"desde la página de Vuforia. Luego, en la jerarquía de Unity, se agrega un elemento .ARCamera z otro elemento llamado ÏmageTarget". Como hijo de este último elemento se incorpora el modelo del robot (ver figura 5a). Cabe destacar que cada articulación es hija de la anterior. De esta forma, cuando la aplicación detecta la imagen Target, se desplegará el modelo 3D correspondiente.



(a) Jerarquía del robot



(b) Target en el entorno de Vuforia

3.4. Arduino

Arduino es una plataforma de hardware y software libre que se utiliza para desarrollar proyectos electrónicos basados en microcontroladores. Los distintos modelos de Arduino difieren en cuanto a su tamaño, el tipo de microcontrolador que incluyen y la cantidad de entradas y salidas que tienen disponibles. El lenguaje de programación de Arduino es una variante de C++ que se ha simplificado para que sea fácil de aprender y utilizar para personas que no tienen experiencia previa en programación. La plataforma Arduino se utiliza ampliamente en proyectos de electrónica y robótica, como la creación de prototipos de dispositivos electrónicos, robots autónomos y sistemas de control de procesos industriales. También se utiliza en proyectos educativos para enseñar a los estudiantes sobre electrónica y programación. Un

Figura 6: Logo de Arduino

control remoto basado en Arduino tiene varias ventajas. Primero, permite a los usuarios construir un control remoto personalizado que se adapte a sus necesidades y preferencias específicas. En segundo lugar, los controladores Arduino son generalmente más asequibles que los controladores comerciales, lo que los hace adecuados para proyectos de bajo presupuesto. Además, los controladores Arduino son fáciles de programar y modificar, lo que permite a los usuarios agregar nuevas funcionalidades y mejorar su control remoto a medida que avanzan en sus habilidades de programación.

3.5. Visual Studio Code

Visual Studio Code es un editor de código fuente y un entorno de desarrollo integrado gratuito y de código abierto que permite escribir y editar código en varios lenguajes de programación y proporciona herramientas y características para facilitar el desarrollo de software. Es compatible con diferentes lenguajes y plataformas, cuenta con una gran cantidad de extensiones y plugins disponibles y es ligero y fácil de usar. Además, permite realizar depuración en tiempo real, tiene integración con controles de versiones y facilita el trabajo en equipo en un mismo proyecto. También es gratuito y de código abierto.

Figura 7: Logo de Arduino

4. Hardware Utilizado

4.1. Control Remoto

Para hacer uso del entorno virtual desarrollado es necesaria una IHM (Interfaz Hombre Máquina) que permita la interacción entre el sistema y el usuario. Para este proyecto la IHM utilizada es un *joystick* basado en Arduino.

Para el control remoto se propone como solución utilizar un canal de radiofrecuencia bidireccional, utilizando como transmisor un control remoto que usa como “cerebro” un Arduino Pro

Mini y como receptor, un Arduino Uno que se conecta de manera Serial con la computadora. Se toma la decisión de utilizar esta configuración para la comunicación debido a que se puede reutilizar el control remoto de otros proyectos, abaratando costos. Cabe destacar que existen otras opciones, tales como realizar la comunicación entre el control y el dispositivo móvil mediante Bluetooth, lo cual haría prescindible la computadora.

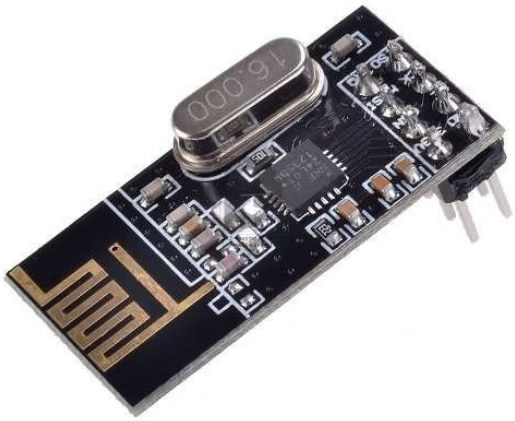
La comunicación por la cual se transmiten los datos necesarios para el funcionamiento se lleva a cabo por radio-frecuencias gracias al uso del modulo transceptor NRF24L01. El Arduino es el encargado de procesar la información de los controles y comunicarse con el módulo NRF24L01. Luego éste los envía por radio-frecuencia. La función del módulo de carga es cargar de forma segura la batería de litio que contiene. El módulo step-up cumple la función de elevar el voltaje de 3.7V (baterías) a 5V, debido a que el último mencionado es el que utiliza Arduino.

Componente	Cantidad
Arduino Pro-mini	1
Analógicos	2
Botones	4
Potenciometros	2
Switch	2
Módulo de carga	1
Módulo Step-up	1

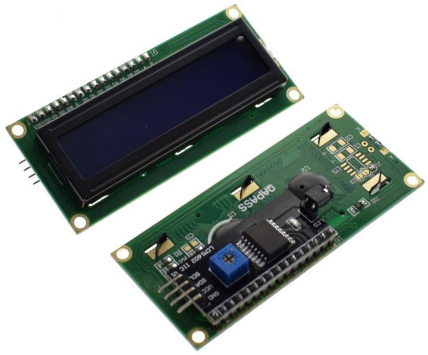
4.1.1. NRF24L01

El módulo NRF24L01 (figura 8a) es un componente de comunicación inalámbrico que se puede usar con Arduino para permitir la comunicación entre dispositivos. Se conecta a los pines RX y TX de Arduino y se controla con una librería especial en el código de Arduino. Una vez configurado, puede transmitir y recibir datos inalámbricamente en diferentes bandas de frecuencia y modos de operación. La banda de 2,4 GHz es una de las bandas industriales, científicas y médicas

(ISM) reservadas internacionalmente para dispositivos de baja potencia sin licencia, como teléfonos inalámbricos, dispositivos Bluetooth, NFC y redes WiFi.



(a) Módulo NRF24L01



(b) Módulo LCD1602

4.1.2. Pantalla LCD

Para tener una mejor realimentación de la posición actual del robot (en el modo Remoto) se adicionó una pantalla LCD (figura 8b) que sirve para ver la posición actual de cada grado de libertad y la posición consigna.

Para la comunicación de la pantalla con el receptor se usa I2C, que es un protocolo de comunicación de datos que permite la conexión entre dispositivos electrónicos a través de dos hilos, SDA y SCL, que permiten enviar y recibir datos.

La selección de este tipo de comunicación es que solo necesito 2 hilos para realizar la comunicación; uno de estos hilos se utiliza para enviar datos (SDA) y el otro se utiliza para recibir señales de reloj (SCL) que sincronizan las operaciones de comunicación. En este caso particular se puede realizar la comunicación tanto por SPI como por I2C, ya que el microcontrolador utilizado en el control lo permite, pero por cuestiones de modularidad del sistema se decide conectar la pantalla

LCD al receptor.

Movimiento sincrónico, no punto a punto

5. Interfaz: Operaciones de movimiento

Control cinemático directo

Se desarrolla a continuación el modo de operación del robot, posteriormente detallando cada módulo. El brazo robótico se puede controlar de dos maneras: en "Modo Local", donde se utilizan sliders en la aplicación para mover las articulaciones de manera independiente, respetando sus límites articulares. En "Modo Remoto", se utiliza un control remoto que envía una cadena de comandos (trama) a través de un receptor Arduino y puerto serial a una computadora. Un algoritmo en Python recibe la trama y la envía a la aplicación a través de sockets TCP/IP. La aplicación procesa el mensaje y da la orden de movimiento correspondiente. En el Modo Remoto, se muestran los valores comandados para cada articulación en forma de texto. La trama debe comenzar y terminar con dos puntos y un punto final, respectivamente, y contener el ángulo de cada grado de libertad con tres cifras, separadas por una barra. La aplicación actúa como servidor, atendiendo las consultas que se envían desde el conjunto control-computadora, que actúa como cliente. El flujo de trabajo se puede ver en la figura 9.

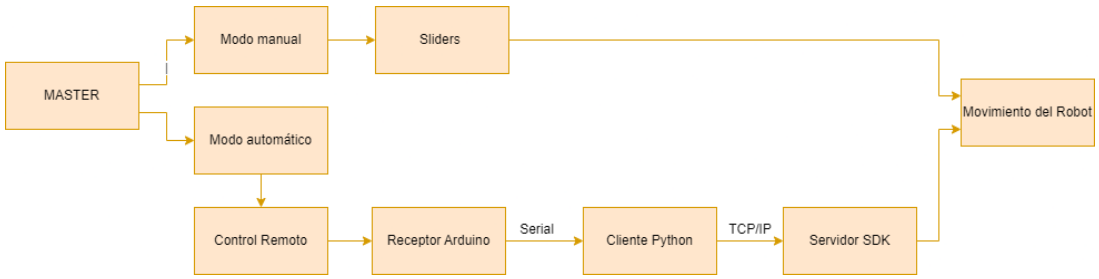


Figura 9: Flujo de trabajo

5.1. Menú Principal

Hay una clase llamada Cambia Escena, que cuenta con un atributo llamado SceneNumber, en el cual al presionar los distintos botones que componen la GUI (modo local, modo remoto, salir) se le asigna un valor a este atributo y este define el comportamiento del programa, que objetos ocultar

y cuáles mostrar. El cambio de valor del atributo se logra mediante un método propio de la clase Botón de Unity ('OnClick()'). La clase que domina el comportamiento de la interfaz es Visual, en la cual en la función Awake() se definen qué objetos estarán activos y cuales no. Una vez que se presionó un botón (en este caso el boton manual) se desactivan los objetos que no servirán para este modo, esto se hace evaluando el valor actual del atributo SceneNumber de CambiaEscena en la función Update(). El menú principal se puede ver en la figura 10.

Figura 10: GUI (interfaz de Usuario)

5.1.1. Modo Manual

En este modo, al presionar sobre el botón "Local Mode", se desactivan los elementos correspondientes al modo remoto tales como elementos de Texto y otros botones y se activan los sliders.

En la clase ArmController simplemente lo que se hace es definir límites articulares (que también son los límites de cada slider) y se dan dos funciones, CheckInput() que captura el valor actual del Slider, y ProcessMovement(), en la cual se interpola el valor capturado y se asigna el mismo al Transform de rotación de cada articulación. En esta última función, lo que se hace es tomar el valor

de cada slider, y rotarlos con Quaternion.Euler(x,y,z), función en la cual se ingresan los ángulos como parámetros como si fuesen angulos de Euler y se convierte la rotación a un cuaternión, más eficiente. Luego se realiza una interpolación esférica suave entre el ángulo anterior y la consigna y se asigna el resultado al valor de Transform de las articulaciones (ver figura 11). Se actualiza el valor del array startRotation para que al interpolar el valor deseado se haga desde el valor actual.

```
//Interpolación esférica suave
for(int i=0;i<6;i++){
    smoothRotation[i] = Quaternion.Slerp(startRotation[i], endRotation[i], 0.1f);
}
// Asigna la rotación suave al objeto
for(int i=0;i<6;i++){
    Link[i].rotation = smoothRotation[i];
}
```

Figura 11: Método de interpolación

5.1.2. Modo Remoto

En este modo, se comanda el movimiento del brazo desde el control remoto, el cual funciona con radiofrecuencia. En primera instancia, se ejecuta el programa en Python encargado de realizar la comunicación vía TCP/IP entre el cliente y el servidor. Se detecta el puerto en el que está el receptor, se pide una confirmación del número de puerto (esto sirve si hay más de un dispositivo conectado a otro puerto serial) y luego se pide que se escriba la IP del celular donde está corriendo la aplicación. Una vez que la conexión fue realizada con éxito, se procede a pasar los comandos desde el control remoto. A través de un potenciómetro se define la posición angular que debe ocupar el robot, indicando los ángulos desde la primera articulación (hombro) hasta la última (muñeca). la forma de operación es la siguiente:

- Se enciende el control remoto.
- Se activa la comunicación RF con un switch (se controla en el receptor).
- Se gira el potenciómetro hacia la posición deseada (hay un feedback de la posición en la pantalla

LCD).

- Se presiona el botón de OK para los ángulos.
- Se presiona el botón de enviar.

Cabe destacar que los ángulos de posición de cada articulación están dentro de los límites articulares del robot, los mismos se definen en el receptor RF. Una vez enviado el mensaje, se codifica en el receptor, iniciando la trama con ":", separando cada ángulo con "/" y finalizando con un punto final. Una vez que se codificó la trama, se envía el mensaje por medio de TCP/IP hasta el socket definido del otro dispositivo (se eligió el puerto 25001 para realizar la comunicación, aunque se puede modificar a criterio). La aplicación, a través del TcpListener, oficiará de servidor en modo de espera, y cuándo llegue un mensaje se decodificará el mensaje y se asignarán las posiciones articulares correspondientes. La clase TcpListener proporciona métodos que aceptan solicitudes, como se requiere que no sean bloqueantes, se usa el método Pending(). Se crea un TcpListener utilizando un IPEndPoint, una dirección IP local y un número de puerto. Con el método Start() se comienzan a escuchar las peticiones.

El mensaje, que es de tipo string, se divide en 6 valores (para eso están los delimitadores "/") por lo tanto se convierte luego cada uno de esos valores en un valor entero y éste será el ángulo de posición de cada grado de libertad. La clase MyListener tiene como atributos el puerto, el texto de la trama y las posiciones. En el primer método se inicia el Listener y se dan las posiciones iniciales. Luego en Update (se ejecuta cada frame) si el Listener está disponible se convierte el string en un array de strings y se asignan los valores a las posiciones. Se usa una corutina para hacer rotar las articulaciones de manera suave y gradual. Una corutina es una función especial en Unity que se ejecuta de manera asíncrona y puede ser suspendida y reanudada en diferentes momentos. En este caso, sirve para realizar el giro en un período de tiempo definido, y no hacer que los objetos giren inmediatamente (ver figura12).

Como la aplicación cuenta con entradas del mundo real que se combinan con elementos del

```
IEnumerator RotateObject(float duration)
{
    float elapsedTime = 0f;

    while (elapsedTime < duration)
    {
        for(int i=0;i<6;i++){
            smoothRotation[i] = Quaternion.Slerp(startRotation[i], endRotation[i], elapsedTime / duration);
        }
        // Asigna la rotación suave al objeto
        for(int i=0;i<6;i++){
            Link[i].rotation = smoothRotation[i];
        }
        elapsedTime += Time.deltaTime;
        yield return null;
    }

    //Hace al valor actual ser el inicial para el siguiente frame
    for(int i=0;i<6;i++){
            startRotation[i]=Link[i].rotation;
        }
}
```

Figura 12: Corutina utilizada

mundo virtual. Esta aplicación, al ser realizada en Realidad Aumentada, forma parte de la Realidad Mixta (figura 13). Se obtiene una buena representación de como se comportaría el robot de manera real y que tiene interacción en tiempo real.

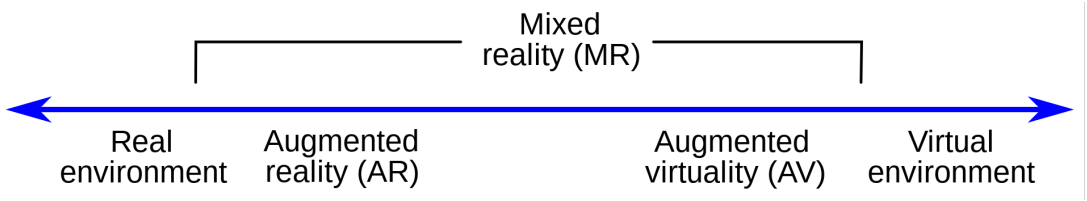


Figura 13: continuo de la virtualidad

6. Resultados

Se logró de manera satisfactoria crear una aplicación en la que se simule el movimiento de las articulaciones de un robot Puma 560 de seis grados de libertad. A su vez, también se integró con un control de tipo *joystick* permitiendo un mayor nivel de inmersión con el medio y emulando los controladores industriales.

Se logró un movimiento fluído del robot al seguir las consignas y que el mismo sea acorde tanto a las posiciones de los sliders en el modo local como las que se muestran en la pantalla del control remoto en el modo remoto.

Se lograron tiempos aceptables en el seguimiento de consignas y en la comunicación entre el cliente (computadora y control) y el servidor (aplicación en el celular).

7. Conclusión

Durante el desarrollo del proyecto de Realidad virtual pude implementar los conocimientos adquiridos durante el cursado de la materia, además de poder relacionar conceptos vistos en otras materias y cómo se pueden integrar entre sí.

En cuanto a las herramientas, me sirvió para poder recordar conocimientos de algunas con las que se experimentó, por ejemplo MATLAB cuando se estaba evaluando cómo hacer el proyecto, o de Python. A su vez, aprendí conceptos de programación muy interesantes, tal como la conexión en sockets en Python, la programación en C y el entorno Unity.

8. Futuras implementaciones

Hay algunas áreas en las que se podría mejorar la aplicación, como por ejemplo:

- Utilizar la librería RTB de Peter Corke (ver figura 14a) para mejorar la precisión y el rendimiento del brazo robótico en la aplicación. La misma puede ser en el mismo cliente de Python que ya se tiene o en MATLAB.
- Agregar más modelos en la aplicación, como por ejemplo distintos brazos robóticos o componentes adicionales, para hacerla más versátil y adaptable a diferentes situaciones. Cabe destacar que para facilitar los cálculos cinemáticos y/o dinámicos, el Toolbox tiene cargados algunos modelos de robots.

- Poder grabar una serie de movimientos para que el robot realice la trayectoria.
- Aprovechar la potencia de la computadora para realizar cálculos más complejos y precisos en tiempo real, lo que podría mejorar el rendimiento y la precisión del brazo robótico en la aplicación.
- Conectar el Display LCD directamente al microcontrolador del Control Remoto
- Cambiar la interfaz de conexión, como por ejemplo a Bluetooth o utilizando un módulo WiFi de bajo coste tal como el ESP8266 con pila TCP/IP completa y capacidad de MCU (Micro Controller Unit) producida por el fabricante chino Espressif Systems.
- Diseñar un control remoto más parecido a los controles industriales (ver figura 14b).

(a) Toolbox RGB para Python

(b) Teach Pendant Industrial

Referencias

[1] Arduino (2022). Página de referencia de arduino. https://www.arduino.cc/reference/en/.

[2] freeCodeCamp.org (2022). Augmented reality for everyone - full course. https://www.youtube.com/watch?v=WzfDo2Wpxks&t=6711s.

[3] Microsoft (2022). Referencia de c#. https://learn.microsoft.com.

[4] Unity (2022a). Página educativa de unity. https://www.youtube.com/@unity.

[5] Unity (2022b). Referencia de unity. https://docs.unity3d.com/.

[6] Vuforia (2023). Documentación oficial. https://developer.vuforia.com/.

[7] Wikipedia (2023a). Augmented reality. https://en.wikipedia.org/wiki/Augmented_reality.

[8] Wikipedia (2023b). Continuo de la virtualidad. https://es.wikipedia.org/wiki/Continuo_de_la_virtualidad.