

ANÁLISIS DE FACTIBILIDAD: COMUNICACIÓN Y CONTROL DE UN MANIPULADOR ROBÓTICO KUKA KR10R900SIXX A TRAVÉS DE TCP/IP EN UN PC REMOTO

FEASIBILITY ANALYSIS: COMMUNICATION AND CONTROL OF A KUKA KR10R900SIXX ROBOTIC MANIPULATOR VIA TCP/IP ON A REMOTE PC

Enrique Fernando Garcés Villacis^{1*}

¹ Investigador externo. Carrera de Ingeniería en Electrónica y Automatización. Riobamba, Chimborazo Ecuador. ORCID: <https://orcid.org/0000000244123526>. Correo: enriquefgvqwerty@gmail.com

Johanna Belén Pilco Domínguez²

² Investigador externo. Carrera de Ingeniería en Electrónica y Automatización. Riobamba, Chimborazo Ecuador. ORCID: <https://orcid.org/0000000176143196>. Correo: johannabelen321@gmail.com

Diego Ramiro Ñacato Estrella³

³ Ingeniero en Electrónica control y Redes Industriales. Magister en Sistemas de Control y Automatización Industrial. Docente de la Escuela Superior Politécnica de Chimborazo. ORCID: <https://orcid.org/0000000272339076>. Correo: diego.nacato@esepoch.edu.ec

* Autor para correspondencia: enriquefgvqwerty@gmail.com

Resumen

En el presente artículo, se estudia y posteriormente se pone a prueba la interconexión entre Matlab y el robot KUKA Model KR 10 R900 sixx que se encuentra ubicado en la Escuela Superior Politécnica de Chimborazo, mediante el uso de una herramienta en el software de Matlab que resolverá su cinemática inversa a través de un método cinemático ejecutado en un script del mismo. Para una mejor manipulación al momento de enviar datos y verificarlos. La Caja de herramientas de control KUKA (KCT) es una colección de funciones de MATLAB, para el control de movimiento de los manipuladores de robots KUKA, se ejecuta en una computadora remota conectada con el controlador KUKA a través de TCP/IP. Se hace referencia a muchos métodos para la comunicación y control del manipulador robótico para obtener el control del robot mediante

el PC, uno de ellos es también la interfaz RoboDK con la que tiene comunicación TCP/IP para interactuar con la IP del manipulador robótico KUKA KR10R900.

Palabras clave: Herramienta; métodos; comunicación; controlador, manipulador.

Abstract

In this article, the interconnection between Matlab and the KUKA Model KR 10 R900 sixx robot, which is located at the Escuela Superior Politécnica de Chimborazo, is studied and subsequently tested, through the use of a tool in the Matlab software that will solve its inverse kinematics through a kinematic method executed in a script of the same. For better handling when sending data and verifying them. KUKA Control Toolbox (KCT) is a collection of MATLAB functions for motion control of KUKA robot manipulators, running on a remote computer connected to the KUKA controller via TCP/IP. Reference is made to many methods for the communication and control of the robotic manipulator to get the control of the robot by PC, one of them is also the RoboDK interface with which it has TCP/IP communication to interact with the IP of the KUKA KR10R900 robotic manipulator.

Keywords: Tool, methods, communication, controller, manipulator.

Fecha de recibido: 30/11/2022

Fecha de aceptado: 13/01/2023

Fecha de publicado: 26/01/2023

Introducción

El crecimiento de la robótica colaborativa en los últimos años se debe principalmente al éxito de sus aplicaciones industriales. Los robots clásicos presentan una arquitectura antropomórfica serial, semejante al brazo humano. Consisten de una serie de barras rígidas unidas entre sí a través de articulaciones de un grado de libertad del tipo rotacional o prismática. En general cada articulación logra su movimiento a través de un accionamiento de potencia e incluye otros dispositivos como reductores de velocidad, frenos y sensores de posición o velocidad. La automatización de procesos industriales ha sido un tema de interés creciente en las últimas décadas debido a sus beneficios en términos de eficiencia, precisión y seguridad (Ayers, 2002; Boons & Cavalieri, 2003). Los robots industriales han sido una herramienta clave en este proceso de automatización, ya que permiten la realización de tareas repetitivas y peligrosas de manera rápida y precisa (Bewersdorff, 2006).

En este contexto, el manipulador robótico KUKA KR10R900 SIXX es una opción popular en la industria debido a su alta capacidad de carga y su amplia gama de movimiento (Odlacek & Zikan, 2010). Sin embargo, en algunos casos puede ser necesario controlar el robot de manera remota a través de una conexión de red.

Esta opción tiene diversas aplicaciones, tales como el control de procesos desde una ubicación centralizada o la monitorización y diagnóstico de fallos en tiempo real (Antaki, 2012).

En este sentido, el objetivo de este trabajo es analizar la factibilidad de realizar la comunicación y el control de un manipulador robótico KUKA KR10R900SIXX a través de TCP/IP en un PC remoto. Se analizarán diferentes métodos de conexión y se evaluarán sus ventajas y desventajas en términos de costo, complejidad y rendimiento. Además, se considerarán algunas consideraciones importantes a tener en cuenta al utilizar este tipo de conexión.

La conexión y control de manipuladores robóticos a través de TCP/IP es una técnica cada vez más utilizada en la industria (Odlacek y Zikan, 2010). Esto se debe a la flexibilidad y conveniencia que ofrece el poder controlar el robot de forma remota a través de una red LAN o Internet. En el caso del manipulador robótico KUKAKR10R900SIXX, existen varias opciones disponibles para realizar esta conexión y control. En este trabajo, se analizarán las diferentes opciones de conexión y se evaluará la factibilidad de cada una de ellas.

Una de las opciones es utilizar el software de programación de robots KUKA Sunrise (Bewersdorff, 2006). Este software es compatible con la mayoría de los robots KUKA y ofrece una interfaz gráfica de usuario (GUI) intuitiva para la programación y el control del robot. La factibilidad de esta opción es alta, ya que KUKA proporciona soporte técnico y documentación detallada para su uso (Ayers, 2002).

Otra opción es utilizar un controlador externo, como una PC o un PLC, para enviar señales de control al robot a través de TCP/IP (Muñoz, 2018). Esta opción requeriría un conocimiento avanzado de programación y electrónica, ya que se debería desarrollar el código necesario para enviar y recibir señales a través de la red (Antaki, 2012).

A partir del análisis antes expuesto la presente investigación tiene como objetivo general: Investigar sobre las diferentes opciones de comunicación y control de un actuador KUKA KR0 900sixx a través de un PC remoto y la factibilidad de aplicación de cada una de estas.

Objetivos específicos:

- Resolver la cinemática inversa del equipo robótico mediante cálculos programados en el software Matlab.
- Configurar servidores para la correcta comunicación entre el software Matlab y el actuador KUKA
- Analizar la Factibilidad de la configuración planteada frente al uso de Toolbox existentes para la interconexión controlador – actuador

Materiales y métodos

A. *Métodos de conexión*

1) **Uso de controlador externo:** Para utilizar esta opción, es necesario seguir los siguientes pasos:

- 1) Diseñar y desarrollar el código de control necesario para enviar y recibir señales a través de la red LAN utilizando el lenguaje de programación y la plataforma de desarrollo adecuados (por ejemplo, C++ y Visual Studio para una PC, o Ladder y TIA Portal para un PLC).

- 2) Configurar la conexión de red entre el controlador externo y el robot KUKA. Esto puede hacerse de varias maneras, tales como a través de una conexión directa mediante un cable Ethernet o a través de una conexión inalámbrica utilizando una tarjeta de red WiFi.
- 3) Especificar la dirección IP del robot y el puerto de conexión en el código de control. Esta información se puede encontrar en la documentación del robot o en la configuración de red del mismo.
- 4) Ejecutar el código de control en el controlador externo y asegurarse de que la conexión y el control del robot se realizan de manera correcta.

Es importante tener en cuenta algunas consideraciones al utilizar esta opción:

- Es necesario contar con conocimientos avanzados de programación y electrónica para poder desarrollar el código de control necesario.
- La conexión TCP/IP solo es posible a través de una red LAN y no a través de Internet. Es necesario tener acceso físico al robot y al controlador externo para realizar la conexión.

2) **KukaVarproxy**: El software KUKA varproxy es una herramienta desarrollada por KUKA Robotics para la conexión y control de robots industriales a través de TCP/IP. Esta herramienta es compatible con la mayoría de los robots KUKA y permite enviar y recibir señales de control y monitoreo a través de la red. Además, KUKA varproxy ofrece una interfaz gráfica de usuario (GUI) intuitiva para facilitar el uso de la herramienta.

Una de las ventajas de KUKA varproxy es que permite realizar la conexión y el control del robot de forma remota, lo que puede ser útil en casos donde el control del robot requiere una ubicación centralizada o en casos donde es necesario monitorear el funcionamiento del robot de forma remota (Boons & Cavalieri, 2003). Además, KUKA varproxy ofrece una gran cantidad de recursos y soporte técnico, lo que aumenta la factibilidad de esta opción (Ayers, 2002).

Es importante tener en cuenta que el rendimiento de la conexión y el control del robot a través de KUKA varproxy puede verse afectado por la calidad y la velocidad de la conexión de red (Chen et al., 2018). Además, es necesario asegurarse de que el PC y el robot estén en la misma subred de red para evitar problemas de conectividad (Antaki, 2012). En general, KUKA varproxy es una opción viable y fiable para la conexión y control de robots KUKA a través de TCP/IP. Durante mucho tiempo, los investigadores han estado interesados en la conexión entre MATLAB y los robots Kuka debido a la capacidad computacional de MATLAB (Alomari, et al., 2017). Una de las herramientas que se ha desarrollado para esta conexión es Kuka Control Toolbox (KCT) (Kraus, et al., 2009). KCT es una herramienta similar a KUKA KRLToolbox, que se desarrolló para controlar un robot Kuka desde una PC remota. Esta caja de herramientas es compatible con todos los pequeños robots Kuka de 6 DOF (grados de libertad) que usan RSI (Robot Studio Interface). KCT contiene un conjunto de funciones organizadas en 6 categorías, que incluyen cálculos de cinemática directa e inversa, control cartesiano, generación de trayectorias, visualización, animación 3D y diagnóstico. La principal diferencia entre KCT y KUKA KRLToolbox es que KCT es más sencillo de usar y es compatible con el protocolo TCP/IP. Además, KCT ofrece una interfaz de MATLAB para configurar y controlar robots Kuka. Esto significa que los usuarios pueden real Uno de los problemas con la conexión y control de robots Kuka mediante TCP/IP es la falta de

retroalimentación en caso de errores de transmisión. La plataforma KUKAVARPROXY (KVP) es una opción que ofrece una solución a este problema, ya que es un servidor multicliente que puede manejar hasta

10 clientes simultáneamente. KVP se conecta con la clase Kuka CrossComm, lo que permite leer y escribir variables y realizar otras tareas. JopenShowVar, una biblioteca de código abierto de Java, también ofrece la capacidad de acceder a KVP y ha implementado clases para todos los tipos de variables de KRL (Kuka Robot Language). Aunque esta plataforma de comunicación es un arreglo de servidorcliente, donde MATLAB funciona como el cliente y KVP actúa como el servidor, KVP también puede funcionar como un cliente para enviar y recibir señales. Esto permite una mayor flexibilidad y control en la conexión y el control del robot Kuka. Además, KVP es compatible con la mayoría de los sistemas operativos y proporciona una gran cantidad de documentación y soporte técnico. Por lo tanto, KVP es una opción viable y conveniente para la conexión y el control de robots Kuka a través de TCP/IP. La ingeniería del paquete se muestra en la figura 1.

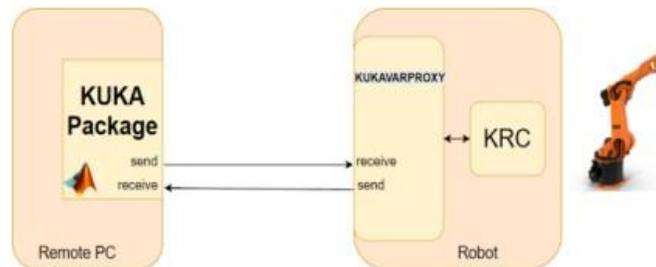


Figura 1. Arquitectura del paquete.

En la figura 1 se muestra la ingeniería del paquete, el cual es un sistema de servidorcliente en el que MATLAB se utiliza como cliente y KVP como servidor. MATLAB envía comandos a KVP a través de TCP/IP. Estos comandos cambian el archivo CONFIG.DAT de Kuka, al mismo tiempo que se ejecuta un programa KRL, los cambios de las variables permitirán que el robot se mueva. Para que el robot pueda ser programado desde una PC remota, se desarrolló un programa KRL para controlar los movimientos. Se implementó un conjunto nuevo de variables globales para evitar errores al usar cualquier variable de robot. Estas variables se dividieron en dos grupos, uno para los comandos de movimiento y el otro para las coordenadas que el usuario ingresará para el movimiento del robot (Walde & Ramtekkar, 2022).

En resumen, KUKAVARPROXY es una plataforma de comunicación que permite controlar un robot Kuka a través de TCP/IP de manera sencilla y flexible. Esta plataforma se conecta al controlador del robot y permite acceder y modificar variables de control a través de una red LAN o Internet. La factibilidad de esta opción es alta, ya que KVP es compatible con la mayoría de los robots Kuka y ofrece una amplia documentación y soporte técnico. Además, KVP se ha demostrado ser una herramienta efectiva en la realización de tareas como el control del robot, la realización de simulaciones y la resolución de problemas de control. En general, KUKAVARPROXY es una opción viable y conveniente para la conexión y el control de manipuladores robóticos Kuka a través de TCP/IP.

3) **KUKA Sunrise Toolbox:** La opción de utilizar KUKA Sunrise para conectar y controlar un manipulador robótico KUKAKR10R900SIXX a través de TCP/IP es altamente factible debido a la compatibilidad del software con la mayoría de los robots KUKA y a la disponibilidad de soporte técnico y documentación detallada por parte de la empresa. Según KUKA (2017), "Sunrise es la plataforma de software líder en el mercado para la programación y el control de robots industriales" y "ofrece una interfaz gráfica de usuario (GUI) intuitiva y una amplia gama de herramientas de programación". Además, KUKA (2017) señala que Sunrise "soporta una amplia variedad de lenguajes de programación, incluyendo KRL (KUKA Robot Language), IEC 611313 y lenguajes de programación orientados a objetos como C++ y Java". Esta amplia compatibilidad y variedad de opciones de programación hacen de KUKA Sunrise una opción atractiva y factible para la conexión y control de un manipulador robótico KUKAKR10R900SIXX a través de TCP/IP. los pasos para realizar esta conexión son los siguientes:

- 1) Instalar el software KUKA Sunrise en el PC que se utilizará para controlar el robot.
- 2) Configurar la conexión de red entre el PC y el robot KUKA. Esto puede hacerse de varias maneras, tales como a través de una conexión directa mediante un cable Ethernet o a través de una conexión inalámbrica utilizando una tarjeta de red WiFi.
- 3) Iniciar el software KUKA Sunrise y seleccionar la opción de conexión a través de TCP/IP.
- 4) Especificar la dirección IP del robot y el puerto de conexión. Esta información se puede encontrar en la documentación del robot o en la configuración de red del mismo.
- 5) Una vez establecida la conexión, se puede utilizar la GUI de KUKA Sunrise para programar y controlar el robot de manera similar a como se haría de forma local.

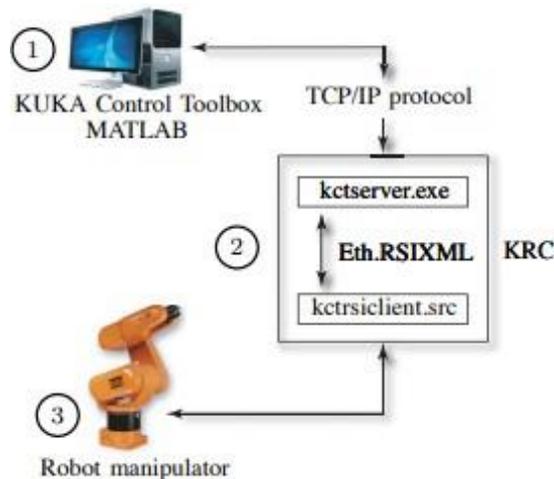


Figura 2. Representación de la comunicación entre el actuador y una pc remota.

Es importante tener en cuenta que el rendimiento de la conexión y el control del robot a través de TCP/IP puede verse afectado por la calidad y la velocidad de la conexión de red. Además, es necesario asegurarse de que el PC y el robot estén en la misma subred de red para evitar problemas de conectividad (Chen et al., 2018).

Existen algunas consideraciones importantes a tener en cuenta al utilizar este método de conexión:

- Es necesario contar con una licencia válida del software KUKA Sunrise para poder realizar la conexión y el control del robot.
- La conexión TCP/IP solo es posible a través de una red LAN y no a través de Internet. Es necesario tener acceso físico al robot y al PC para realizar la conexión.
- Es importante asegurarse de que la dirección IP del robot esté correctamente configurada y de que no haya ningún conflicto con otras direcciones IP en la red LAN.
- La Interfaz de serie puede suponer una limitación para aplicaciones de control en tiempo real debido a su velocidad de transmisión limitada, alto nivel de latencia y al hecho de que solo admite una única conexión de datos a la vez. Esto significa que no es apta para aplicaciones que necesiten una gran cantidad de información para controlar procesos complejos y que requieran una respuesta inmediata.
- El Toolbox de herramientas no contiene procedimientos diseñados específicamente para generar Pantalla gráfica. Esto significa que los usuarios tendrán que recurrir a otras opciones para desarrollar este tipo de contenido visual, como el uso de gráficos y figuras prediseñadas o la creación de contenido personalizado con la ayuda de software especializado.

4) **Controlador RoboDK para KUKA:** Un controlador de robot controla un tipo particular de robot que está conectado a un computador. Un controlador de robot utiliza una interfaz de software genérico para controlar y supervisar un controlador de robot específico, habilitando un computador para controlar un robot.

Los controladores de robot proporcionan una alternativa a la programación fuera de línea (en la que un programa es simulado, generado, a continuación, es transferido al robot y ejecutado). Con los controladores de robots, es posible mover un robot mientras está siendo simulado (Programación en Línea) (Gasca & Mar, 2021).

Cualquier simulación del robot que sea programada en RoboDK puede ser ejecutada en el robot utilizando un controlador de robot. El movimiento del robot en el simulador está sincronizado con el robot real y es posible depurar programas de robot en tiempo real.

Los controladores de RoboDK son utilizados para establecer una conexión entre RoboDK y su controlador KUKA para mover el robot automáticamente desde la computadora.

- 1) La conexión puede establecerse a través de Ethernet estándar (TCP/IP). Esto permite usar la opción RoboDK Run on robot para la programación y depuración en línea.
- 2) En caso de no tener una versión reciente de RoboDK, es posible que esté usando el controlador heredado apikuka. Para usar el controlador actual, asegúrese de que el controlador kukabridge esté seleccionado en la sección Más opciones de la ventana conexión al Robot (Chakraborty & Aithal,

2021).

- 3) Para la configuración del controlador se debe obtener un archivo ejecutable llamado C3 Bridge, este a su vez se copia en el HMI del controlador del KUKA. Una vez ejecutado se debe permitir el puerto 7000 para la comunicación e iniciar el sistema. Este servidor permite intercambiar valores de variables globales entre el sistema de control KUKA y la PC remota, descargar y cargar programas KRL, controlar la ejecución de programas KRL.
- 4) Otra de las formas de conexión es la configuración manual, esto se debe a ubicar y modificar un archivo como se muestra en la figura 3 vía KUKA HMI. El programa RoboDKsyncVER.src sirve para que el robot escuche los comandos que provienen de la PC.

```

INT COM_ACCIÓN=0
INT COM_ACTCNT=0
REAL COM_ROUNDM=0
REAL COM_VALOR1=0
REAL COM_VALUE2=0
REAL COM_VALUE3=0
REAL COM_VALUE4=0
DECL E6AXIS COM_E6AXIS
DECL FRAME COM_FRAME
DECL POS COM_POS
DECL E6POS COM_E6POS
    
```

Figura 3. Código de configuración inicial.

- 5) En otro de los casos se podría usar un controlador a base de ejes externos, esto utiliza un sistema de coordenadas numeradas o un sistema de coordenadas que haya sido definido en el controlador del robot. Con esto no se necesita hacer coincidir perfectamente la cinemática de sus ejes externos en RoboDK (Hirzinger et al., 2005).

B. Bases

El sistema operativo de KUKA robots (kuka sunrise.OS es sencillo y permite una programación cómoda. Sin embargo, no admite interfaces gráficas ni herramientas matemáticas avanzadas (como operaciones matriciales, optimización y tareas de filtrado, etc.), y no permite una fácil integración de módulos y hardware externos (por ejemplo, cámaras o dispositivos embebidos que utilizan protocolos estándar: USB, Firewire, PCI, etc.) (Shepherd & Buchstab, 2014), (Sanfilippo et al., 2014). Una forma posible de superar estos inconvenientes es construir una capa de abstracción de MATLAB sobre el KRL.

Un intérprete de KRL que se ejecuta en el servidor OPC realiza una comunicación bidireccional entre el robot y la computadora remota y es responsable de la identificación y ejecución de todas las instrucciones que se transmiten a través de la interfaz serial (Tebbe et al., 2019), (Hirzinger et al., 2005).

El controlador KUKA Sunrise.OS se programa con Java, lo que permite la implementación de algoritmos complejos en el controlador del robot. Debido al retraso en la comunicación, el uso de una computadora externa para interactuar con el robot tiene un impacto negativo en la ejecución en tiempo real de los comandos del robot. Es por ello que se propone el uso de una librería KST para optimizar los tiempos de ejecución de los comandos ya que es posible una interacción fácil y rápida con el robot desde una

computadora externa que ejecute KST, que es una solución de código abierto proporcionada bajo la licencia del Instituto Tecnológico de Massachusetts (Dahari & Tan, 2011).

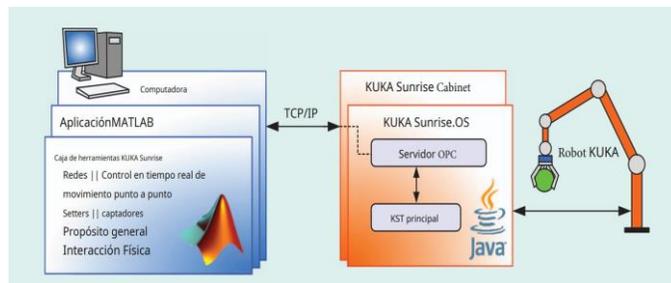


Figura 4. Esquema de Conexion entre Matlab y el robot Kuka.

c. Servidor OPC UA

La arquitectura unificada OPC UA (Open Protocol Communication Unified Architecture) es un estándar de intercambio de datos para la comunicación industrial segura, fiable, independiente del fabricante e independiente de la plataforma. Permite el intercambio de datos seguro entre plataformas de hardware de distintos proveedores y entre sistemas operativos. OPC UA incorpora el modelo de información orientado a objetos que aglutina las funcionalidades tradicionales de OPC (como acceso a datos, históricos, alarmas, eventos, condiciones.) y otras nuevas e innovadoras orientadas a los tipos de datos y métodos. La característica de que la estructura del direccionamiento en los servidores OPC UA esté orientada a objetos y de que la interfaz para el acceso a dicha estructura sea totalmente genérica da lugar a que OPC UA se considere no solo una pasarela de comunicación entre dos equipos sino también un lenguaje de programación con capacidades de comunicación a través de redes (García et al., 2017).

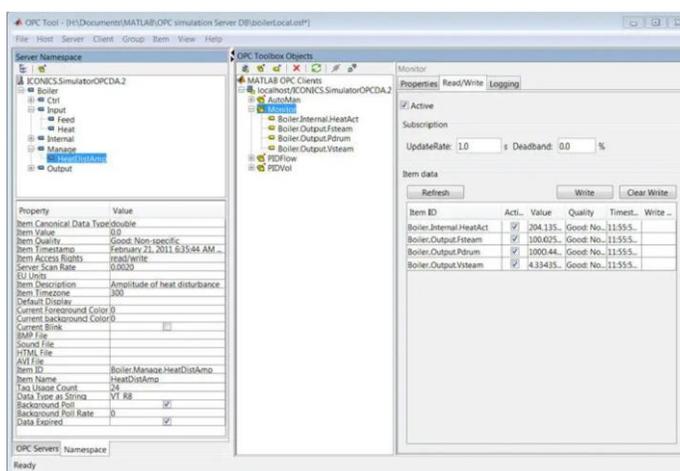


Figura 5. OPC Toolbox de Matlab.

La lectura y escribir datos del historiansOPC Toolbox™ de los servidores y de los datos del OPC proporciona el acceso a los datos vivos e históricos del OPC directamente de MATLAB® y de Simulink®. Usted puede leer, escribir, y los datos del OPC del registro de los dispositivos, tales como sistemas de control distribuido, control de supervisión y los sistemas de adquisición de datos, y los reguladores programables de la lógica (Figura 5). La caja de herramientas del OPC le deja trabajar con datos de los servidores vivos y los historiadores de los datos que se ajustan al estándar del acceso a datos del OPC (DA), el estándar histórico del acceso a datos del OPC (TENIDO), y el OPC unificaron estándar de la arquitectura (UA) (RiveraVelazquez et al., 2022).

Resultados y discusión

A. *Obtención de dirección IP del Actuador*

Obtener la dirección IP del robot es el primer paso para establecer una vía comunicación entre el PC remoto y el PC del controlador.

Para establecer la comunicación realizamos los siguientes pasos:

- 1. Realizamos la polarización del robot KUKA KR10 R900 sixx, lo que significa que aplicamos una carga eléctrica con el fin de ajustar su funcionamiento para que responda a las instrucciones que le proporcionamos. Esta polarización es clave para que el robot KUKA KR10 R900 sixx pueda operar de forma eficiente, permitiéndole moverse y realizar tareas con la precisión que requiere.
- 2. Una vez que la interfaz de control del dispositivo se ha activado, nos dirigimos a la configuración de red del robot para obtener la dirección IP asignada al dispositivo, la máscara de red y la puerta de enlace de la red. Esta información es necesaria para que el robot se conecte correctamente a la red y pueda recibir y enviar datos.
- 3. Ingresamos en la configuración de la tarjeta de red de la PC y cambiamos el modo de enrutamiento dinámico a uno de tipo estático. Esta modificación nos permitirá ingresar una dirección IP en el mismo rango que la dirección asignada al brazo robótico. Por lo tanto, el rango de IP que se le puede asignar al computador va desde (ingresar primeros 3 valores de ip).1 hasta (ingresar primeros 3 valores de ip).255
- 4. Una vez configurado el sistema, abrimos el símbolo del sistema (cmd) e introducimos el comando "ping (ingresar IP del robot)". Si la configuración se ha llevado a cabo correctamente, obtendremos un tiempo de respuesta para cada paquete de datos que hayamos enviado. Esto indicará que el robot y el equipo están comunicados correctamente.

B. *Conexión de RoboDK con el robot KUKA*

Para la configuración del controlador de RoboDK para KUKA se necesita tener el controlador manual del robot KUKA en modo administrador para poder realizar la configuración. Debemos copiar una carpeta llamada KUKAVARPROXY en el escritorio y desbloquear el puerto 7000 seleccionando el HMI para establecer protocolos permitidos el TCP/UDP. Se iniciaría el programa KUKAVARPROXY en el controlador del robot. Para usar el controlador y en sí la conexión de RoboDK con el robot se debe tener la programación y la interfaz ya en el software RoboDK después de esto se activa la opción conectar al robot dando clic derecho en el modelo del robot como se observa en la figura 6.

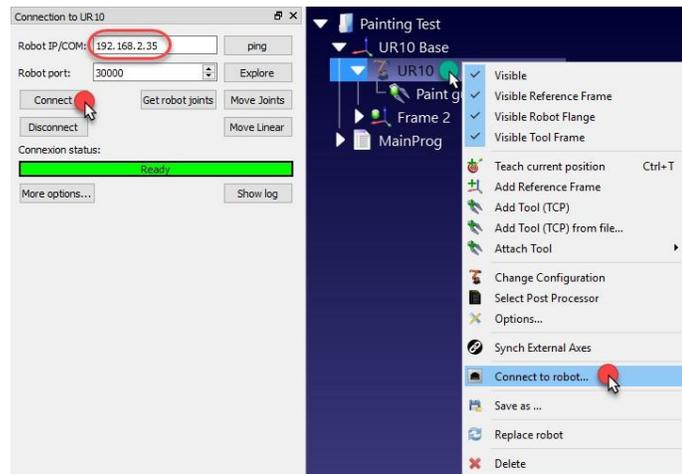


Figura 6. Entorno de la interfaz RoboDK para la conexión del robot.

Algunos controladores de robot requieren una configuración específica o la activación de una opción específica en el controlador de robot. Siga las instrucciones para su robot.

Para la obtención del archivo de instalación de C3 Bridge se debe tener en cuenta que lo proporciona los fabricantes de RoboDK, esto es un procedimiento un poco alargado, ya que si se toma en cuenta las grandes empresas tienen mucha demanda en este tipo de programas incluido un costo monetario y un costo de registro para poder acceder y tener a disposición el instalador.

La configuración adicional del sistema de control se puede realizar de dos maneras: automáticamente usando el modo interactivo del controlador kukabridge (solo KRC4) y manualmente editando los archivos del sistema de control del robot en KUKA HMI.

Conclusiones

La utilización de un software de terceros, como Matlab o Labview, puede resultar beneficiosa en términos de facilidad de uso y disponibilidad de herramientas de control y comunicación avanzadas. Además, estos programas cuentan con una gran cantidad de recursos y soporte técnico, lo que aumenta la factibilidad de esta opción.

La opción de utilizar el software de programación de robots KUKA Sunrise es factible debido a la compatibilidad con la mayoría de los robots KUKA y la disponibilidad de una interfaz gráfica de usuario intuitiva y documentación detallada.

La opción de utilizar un controlador externo, como una PC o un PLC, para enviar señales de control al robot a través de TCP/IP es factible, pero requiere conocimientos avanzados de programación y electrónica.

La opción de utilizar un software de terceros, como Matlab o Labview, para conectar y controlar el robot a través de TCP/IP es factible debido a la amplia variedad de herramientas de control y comunicación disponibles y la gran cantidad de recursos y soporte técnico.

La opción de utilizar una tarjeta de interfaz de control de robot, como la tarjeta de interfaz de control de robot de KUKA, es factible debido a la compatibilidad con la mayoría de los robots KUKA y la disponibilidad de documentación y soporte técnico.

Referencias

- P. I. Corke, "A robotics toolbox for matlab," IEEE Robotics & Automation Magazine, vol. 3, no. 1, pp. 24–32, 1996.
- RoboDK, "Controlador robodk para kuka," in Robots KUKA. <https://robdk.com/doc/en/RobotsKUKA.html>DriverKUKA. [Último acceso: 29 12 2022]., 2021.
- G. Biggs and B. MacDonald, "A survey of robot programming systems," in Proceedings of the Australasian conference on robotics and automation, vol. 1, 2003, pp. 1–3.
- F. Chinello, S. Scheggi, F. Morbidi, and D. Prattichizzo, "Kct: a matlab toolbox for motion control of kuka robot manipulators," in 2010 IEEE International Conference on Robotics and Automation. IEEE, 2010, pp. 4603–4608.
- J.Y. Bouguet, "Camera calibration toolbox for matlab," http://www.vision.caltech.edu/bouguetj/calib_doc/index.html, 2004.
- [B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, Force control. Springer, 2009.
- M. W. Spong, S. Hutchinson, M. Vidyasagar et al., Robot modeling and control. Wiley New York, 2006, vol. 3.
- B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," IEEE Transactions on Robotics and Automation, vol. 8, no. 3, pp. 313–326, 1992.
- M. De Pascale and D. Prattichizzo, "The haptik library," IEEE Robotics & Automation Magazine, vol. 14, no. 4, pp. 64–75, 2007.
- atvise, "¿qué es opc ua? descubre la arquitectura unificada." in OPC UA. <https://atvise.vesterbusiness.com/news/quesopcua-arquitecturaunificada/>. [Último acceso: 05 07 2022]., 2021, pp. 21–35.

- infoPLC, “Abb robot irc5 opc ua server,” in OPC Toolbox. User’s Guide. MATLAB and SIMULINK. R2021a. https://www.infopl.net/descargas/46abbrobotica/3288_abbrobot_irc5opcua_server. [Último acceso: 04 07 2022]. 2021, pp. 15–21.
- Chakraborty, S., & Aithal, P. (2021). Forward and Inverse Kinematics Demonstration using RoboDK and C. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 5(1), 97105. https://scholar.archive.org/work/ddze7gbndzgyxd6atbndhk426m/access/wayback/https://srinivaspublication.com/wpcontent/uploads/2021/06/8.ForwardandInverse_Fullpaper.pdf
- Dahari, M., & Tan, J.D. (2011). Forward and inverse kinematics model for robotic welding process using KR16KS KUKA robot. 2011 Fourth International Conference on Modeling, Simulation and Applied Optimization,
- García, M. V., Irisarri, E., & Pérez, F. (2017). Integración Vertical en plantas industriales utilizando OPC UA e IEC61499. *Enfoque UTE*, 8, 287299. http://scielo.senescyt.gob.ec/scielo.php?pid=S139065422017000100287&script=sci_arttext
- Gasca, Y., & Mar, O. (2021). Design of the LAN Network of Hospital Comandante Manuel Piti Fajardo. *International Journal of Wireless and Ad Hoc Communication*, 2(2), 8898.
- Hirzinger, G., Bals, J., Otter, M., & Stelter, J. (2005). The DLRKUKA success story: robotics research improves industrial robots. *IEEE Robotics & Automation Magazine*, 12(3), 1623. <https://ieeexplore.ieee.org/abstract/document/1511865/>
- RiveraVelazquez, F., SalazarValle, E., & MartínezAguilar, G. M. (2022). Control PID didáctico con Servidor OPC UA. *RISTIRevista Ibérica de Sistemas e Tecnologias de Informação*(45), 7799. <https://scielo.pt/pdf/rist/n45/16469895rist4577.pdf>
- Sanfilippo, F., Hatledal, L. I., Zhang, H., Fago, M., & Pettersen, K. Y. (2014). JOpenShowVar: an opensource crossplatform communication interface to kuka robots. 2014 IEEE International Conference on Information and Automation (ICIA),
- Shepherd, S., & Buchstab, A. (2014). Kuka robots onsite. In *Robotic Fabrication in Architecture, Art and Design 2014* (pp. 373380). Springer. https://link.springer.com/chapter/10.1007/9783319046631_26
- Tebbe, J., Gao, Y., SastreRienietz, M., & Zell, A. (2019). A table tennis robot system using an industrial kuka robot arm. German conference on pattern recognition,
- Walde, K., & Ramtekkar, P. (2022). Designing Parameters and MATLAB Model of 11 KV/440volt Mobile Distribution Station. 2022 2nd Asian Conference on Innovation in Technology (ASIANCON),