

MODELADO NO PARAMÉTRICO DE ROBOT SCARA 3-GDL UTILIZANDO REDES NEURONALES RECURRENTE

NON-PARAMETRIC MODELING OF SCARA 3-GDL ROBOT USING RECURRENT NEURAL NETWORKS

Julio Hernández Díaz¹, Mauricio Iparraguirre Monzón², Carlos Montenegro Castro³, Jasmell Alva Alcántara⁴, Edgar Manzano Ramos⁵

¹²³⁴⁵Universidad Nacional de Trujillo, Facultad de Ingeniería, Escuela Académica Profesional de Ingeniería Mecatrónica, Av. Juan Pablo II s.n., Trujillo, La libertad, Perú, jhernandezd@unitru.edu.pe miparraguirre@unitru.edu.pe cmontenegroc@unitru.edu.pe josalva@unitru.edu.pe emanzano@unitru.edu.pe

RESUMEN

Los robots SCARA (Selective Compliance Assembly Robot Arm) se utilizan para trabajos de paletizado ya que son adecuados para el montaje y el posicionamiento en plano. En este artículo, nuestro objetivo es obtener un modelo dinámico de un robot SCARA de 3GDL haciendo uso de redes neuronales recurrentes (RNN). A partir de los antecedentes, se optó por utilizar un modelamiento no paramétrico, para evitar las pérdidas de algunas características del modelo obtenido tales como la fricción y el desgaste en los actuadores con tal de mejorar la precisión del robot. El flujo de datos se obtendrá a partir del simulador robótico CoppeliaSim. Luego, se deben tener los datos de entrada y salida para la red neuronal, efectuar un preprocesamiento y normalización de los datos para el entrenamiento. Se hará uso de las librerías existentes en el lenguaje de programación de alto nivel Python para desarrollar un algoritmo de identificación por redes neuronales recurrentes basado en el tipo de aprendizaje por retropropagación. Además, se realizará la validación y prueba correspondiente. Como producto final se espera tener un modelo que pueda representar el sistema físico con gran precisión, de tal forma que se puedan diseñar estrategias de control con buen rendimiento.

Palabras clave: RNN, SCARA, CoppeliaSim, modelado no paramétrico, Python.

ABSTRACT

SCARA (Selective Compliance Assembly Robot Arm) robots are used for palletizing work, which are suitable for mounting and flat positioning. In this paper, our objective is to obtain the dynamic model of 3-DOF SCARA robot using recurrent neural networks (RNN). Based on the background, it was decided to use a non-parametric modeling, to avoid the losses of some characteristics of the obtained model such as friction and wear in the actuators, thus improve the precision of the robot. Data Flow will be obtained by robotic simulator CoppeliaSim. Then, the input and output data for the neural network are obtained, preprocessing and normalization of the data are carried out for training. High-level programming language Python libraries will be used to develop an identification algorithm using RNN based on backpropagation learning. In addition, the corresponding validation and test will be carried out. As a final product, it is expected to have a model that can represent the physical system with great precision, in such a way that control strategies can be designed with good performance.

Keywords: RNN, SCARA, CoppeliaSim, Non-parametric modeling, Python.

*Autor para correspondencia: jhernandezd@unitru.edu.pe





INTRODUCCIÓN

Con el paso del tiempo, el uso de robots industriales está aumentando en áreas como alimentos, bienes de consumo, madera, plásticos y electrónica (Guatam 2017). Dentro de ellos, Los robots pick and place han tomado relevancia ya que sustituyen las tareas manuales de selección y son populares por su velocidad de procesamiento, control, precisión y exactitud; uno de los más utilizados para este trabajo es el robot SCARA (Selective Compliance Assembly Robot Arm) (Noshashi 2020).

Todo robot manipulador, como el robot SCARA, es un sistema dinámico que requiere de un controlador para gobernar su movimiento. Antes de diseñar un controlador, es necesario tener el modelo del robot que se desea controlar. Al modelar un sistema robótico de forma paramétrica, usualmente surge la contrariedad de no lograr introducir todas las variables importantes del sistema de forma íntegra, como los centros de masa de los eslabones y la matriz de inercia.

Asimismo, el propio entorno de trabajo del robot manipulador es un factor importante para considerar ya que puede ser capaz de afectar el comportamiento del sistema dinámico del robot en general. Frente a este inconveniente, el modelamiento no-paramétrico mediante redes neuronales tiene la capacidad de aprender comportamientos no lineales por medio de patrones de entrenamiento entrada/salida que

evaden el problema antes mencionado. Entre los tipos de redes neuronales más conocidas, se encuentran las redes neuronales FeedForward (FFNN), las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN).

Namigtle Jiménez propone un método para la reducción del número de parámetros de la estructura H-W a usar en el modelado de un brazo robótico mediante redes neuronales. Este método se basa en realizar una suposición de diseño (elegir funciones de activación lineales) y otra de entrenamiento (inicialización de valores iguales bloque por bloque), y de esa manera priorizar un bajo costo computacional y una mayor precisión (Namigtle 2017).

Grzeidak desarrolla un esquema de identificación empleando un análisis similar a la teoría de estabilidad de Lyapunov, utilizando el Lema de Barbalat y un modelo dinámico de red neuronal de capa única oculta (SHLNN), asegurando la convergencia de los errores residuales de estimación de estado a cero, la acotación de todos los errores de aproximación asociados y, por lo tanto, una mejor eficiencia computacional que los SHLNN convencionales (Grzeidak 2016).

Dyda, A. Oskin, y V. Artemiev demostraron que los modelos de redes neuronales recurrentes (RNN) se pueden entrenar con éxito para el comportamiento no lineal de un robot submarino (UR), siendo muy eficaces para la presentación de la dinámica del mismo. Además, redujeron la complejidad computacional de la identificación

de RNN representándolo como un conjunto de subredes neuronales entrenadas por separado (descomposición) (Dyda 2015).

Velázquez Velázquez, Mariaca Gaspar, Galván Guerra, y Ortiz Rodríguez resuelven el problema de identificación de un robot manipulador cilíndrico mediante el uso de las RNR con un algoritmo de aprendizaje backpropagation dinámico, obteniendo un modelo que permite inferir cómo el robot manipulador cilíndrico responderá a entradas desconocidas, y permitiendo el desarrollo de controladores para gobernar el comportamiento del robot sin necesidad de contar con algún conocimiento sobre el comportamiento dinámico del sistema (Velázquez 2011).

Rafiei, Aali Hosseini, y Akbarzadeh Tootoonchi usaron el método de red neuronal NARX con regresión bayesiana para identificar la dinámica del robot FUM SCARA con alta precisión, considerando la fricción y la dinámica del motor (Rafiei 2018).

CoppeliaSim es un software que permite la simulación de robots que se basa en una arquitectura de control distribuido. Además, es

multiplataforma y permite la creación de contenido portátil, escalable y fácil de mantener, un solo archivo portátil puede contener un modelo (o escena) completamente funcional, incluido el código de control. La API remota de CoppeliaSim permite la comunicación entre el simulador y una aplicación externa (es decir, una aplicación que se ejecuta en un proceso diferente, o en una máquina diferente), y admite transmisión de datos bidireccional. Actualmente es compatible con los siguientes lenguajes: C ++, Java, Python, Matlab y Lua (Robotics 2021).

Resultados de diferentes investigaciones prueban que las redes neuronales recurrentes (RNN) cuentan con la capacidad de realizar modelamiento y control de robots manipuladores. En el presente trabajo, se busca utilizar los datos de CoppeliaSim para realizar una identificación por redes neuronales recurrentes (RNN) que permita obtener un modelamiento no paramétrico de robot SCARA de 3GDL.

MÉTODOS

Para la identificación se sigue una serie de pasos preestablecidos, tal como se detalla en figura 1:



Figura 1. Metodología de obtención de modelo no paramétrico.

Se empieza con la selección del modelo del robot y algunos de los parámetros de entrenamiento de la red neuronal. Luego se procede a obtener los datos de entrada y de salida por medio de CoppeliaSim para el entrenamiento de las RNN. Una vez finalizado el entrenamiento, se hace una comparación entre el modelo no paramétrico obtenido y los datos originales de CoppeliaSim. Si el modelo no paramétrico converge, se puede validar la obtención de un modelo no paramétrico de robot SCARA. De lo contrario, se volverá a obtener los datos de entrada y salida para la red hasta que el modelo obtenido presente una buena convergencia al modelo deseado.

Selección de modelo de robot SCARA

El simulador de robots CoppeliaSim nos permite controlar y visualizar distintos parámetros de robots predeterminados y, asimismo, importar modelos externos de archivos URDF. Para esta investigación, se optó por la segunda opción, diseñando un modelo 3D en el Software de SolidWorks de robot SCARA de 3GLD (ver Fig. 2), seguidamente se utilizó la extensión `solidworks_urdf_exporter` para definir las 3 juntas del robot SCARA, dos revolutas y una prismática, respectivamente.

Se debe tener en cuenta que, al solo contar con 3GDL, no es posible controlar naturalmente la orientación del efector final. Por lo tanto, no será necesario recoger datos de orientación del efector final.

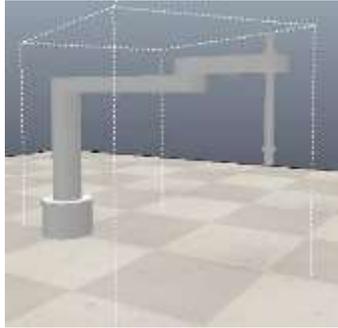


Figura 2. Robot SCARA 3GDL en CoppeliaSim.

Selección de parámetros de entrenamiento

El software de CoppeliaSim permite simular los torques o fuerzas generados por los actuadores propios del software, sin embargo, los autores comprobaron tanto empíricamente como teóricamente que su comportamiento no representa al de un motor de forma realista. Es debido a ello, que se optó por elegir sólo la posición del efector final tanto para datos de entrada como de salida.

Para nuestro robot SCARA de 3 GDL, se pretende que la red neuronal aprenda a predecir una salida $y(t)$ de posiciones futuras del robot SCARA, conociendo una entrada $x(t)$ de posición actual y la posición deseada o setpoint. Debido a esto, se concluye que la red debe reconocer 6 variables diferentes y, por consiguiente, la red neuronal contará con 6 neuronas de entrenamiento. Los otros parámetros, como el número de unidades de la capa escondida, la cantidad de batch y el número de épocas serán escogidos de forma

empírica durante el entrenamiento, de tal manera que el modelo obtenido sea el más adecuado.

El entrenamiento será realizado en el lenguaje de programación de alto nivel Python mediante el simulador de Jupyter Notebook con la finalidad de aprovechar la librería de Keras para RNN.

Obtención de datos de CoppeliaSim

CoppeliaSim es versátil para controlar de forma remota el simulador realizando una comunicación con un software externo. Para la obtención de datos (ver Fig. 4), será necesario controlar los actuadores con tal de obtener diferentes parámetros en el robot SCARA como lo son la velocidad angular (ver Fig. 5) y la posición del actuador final (ver Fig. 6).

Tal y como se muestra en Fig. 3, es posible obtener una gran variedad de datos de CoppeliaSim tales como tiempo, torques, fuerza, posiciones articulares, velocidades articulares, y posiciones del efector final.

t : tiempo; v : [s], $T1$: $v_{11}(t)$ [N.m], $T2$: $v_{12}(t)$ [N.m], F_{11} : $v_{11}(t)$ [N], P_1 : $v_{11}(t)$ [m],
 P_2 : $v_{12}(t)$ [m], P_3 : $v_{13}(t)$ [m], q_1 : $\theta_{11}(t)$ [rad], q_2 : $\theta_{12}(t)$ [rad],
 q_3 : $\theta_{13}(t)$ [rad], qp_1 : $\dot{\theta}_{11}(t)$ [rad], qp_2 : $\dot{\theta}_{12}(t)$ [rad], qp_3 : $\dot{\theta}_{13}(t)$ [rad]

Figura 3. Datos obtenibles desde CoppeliaSim.

Finalmente, se realiza el almacenamiento de 511, 937 líneas de datos de la simulación en un archivo

de texto mediante código en Lua, lenguaje propio de CoppeliaSim.

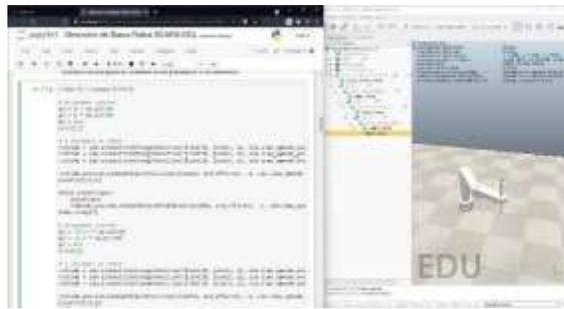


Figura 4. Obtención de datos CoppeliaSim-Python.

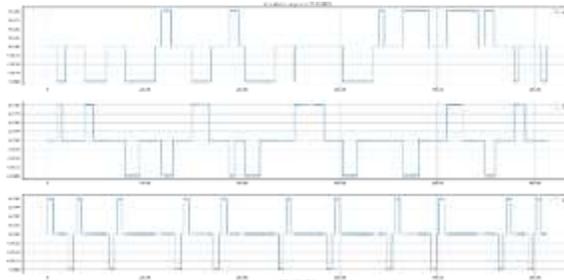


Figura 5. Velocidades angulares vs Tiempo

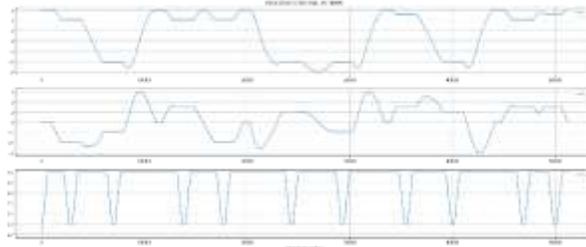


Figura 6. Posición del efector final vs Tiempo.

Entrenamiento de la red neuronal

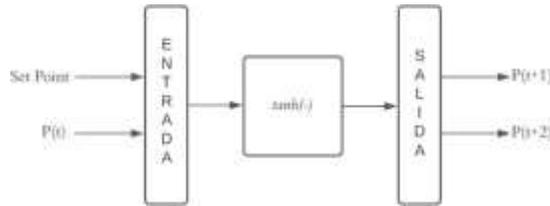


Figura 7. RNN simple a entrenar

Con la finalidad de obtener un modelo del robot SCARA, se utilizará el entrenamiento de una red

neuronal recurrente simple de función tal como la mostrada en Figura 7.

RESULTADOS Y DISCUSIÓN

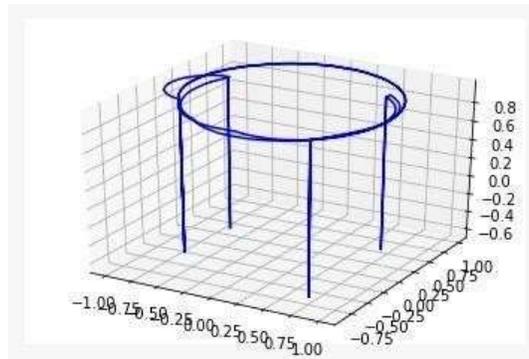


Figura 8. Trayectorias de entrenamiento.

Se obtuvieron 5 trayectorias similares como recolección de datos. Primero, se realizó un pre-procesamiento de los datos, incluyendo una reducción de datos duplicados consecutivos y una normalización de la trayectoria realizada por el efector final del robot, esto debido a que el rango de la función de activación a utilizar, $\tanh(\cdot)$, se encuentra en el intervalo $[-1,1]$. Las 3 primeras

colecciones de posición corresponden a 255096 datos y fueron utilizadas para el entrenamiento (ver Fig. 8 y Fig. 9); una colección de 82816 datos fue utilizada para validación (ver Figura 10 y Figura 11) y la última colección de 48451 datos fue utilizada para realizar una predicción a modo de comprobación.

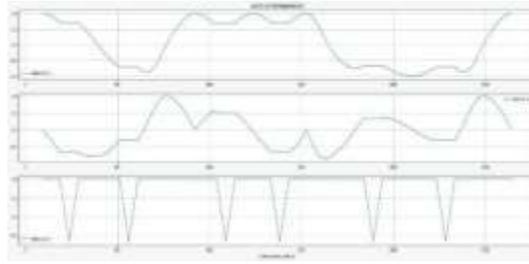


Figura 9. Datos de entrenamiento.

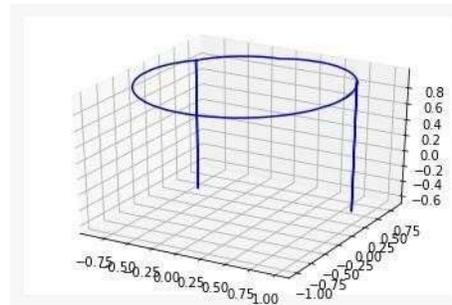


Figura 10. Trayectoria de validación.

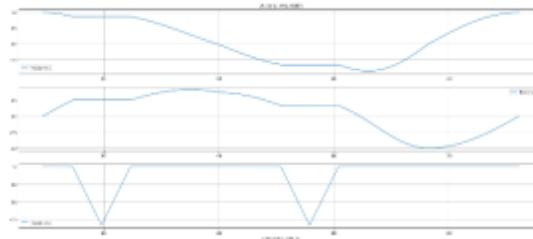


Figura 11. Datos de validación.

Namigtle Jiménez propone un método para la reducción del número de parámetros de la estructura H-W a usar en el modelado de un brazo robótico mediante redes neuronales. Este método se basa en realizar una suposición de diseño (elegir funciones de activación lineales) y otra de entrenamiento (inicialización de valores iguales bloque por bloque), y de esa manera priorizar un bajo costo computacional y una mayor precisión (Namigtle 2017). Además, Grzeidak desarrolla un esquema de identificación empleando un análisis

similar a la teoría de estabilidad de Lyapunov, utilizando el Lema de Barbalat y un modelo dinámico de red neuronal de capa única oculta (SHLNN), asegurando la convergencia de los errores residuales de estimación de estado a cero, la acotación de todos los errores de aproximación asociados y, por lo tanto, una mejor eficiencia computacional que los SHLNN convencionales (Grzeidak 2016).

A través del entrenamiento del modelo de RNN ya definido, se encontraron valores óptimos de entrenamiento con 13 unidades en la capa oculta, 180 épocas y un número de batch de 13000, obteniéndose así un 65.8% de índice de precisión en la validación, además de las gráficas de error

cuadrática medio y pérdidas (ver Fig. 12 y 13). Finalmente, se realizó la predicción de trayectoria de la última colección, obteniéndose una trayectoria suave y aceptable como se muestra en la Fig. 14 y 15.

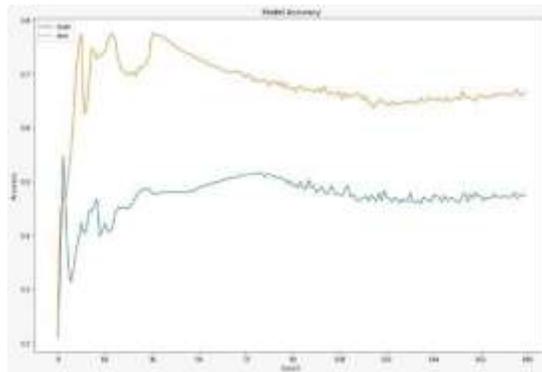


Figura 12. Precisión del modelo vs número de épocas

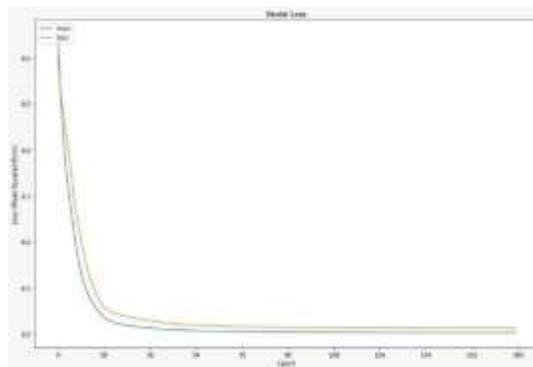


Figura 13. EMC obtenido utilizando la RNR propuesta.

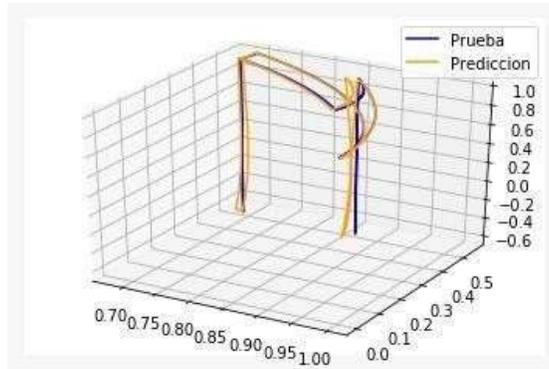


Figura 14. Prueba y predicción de trayectoria.

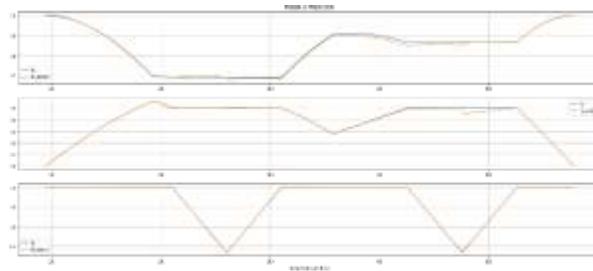


Figura 15. Prueba y predicción de datos.

Dyda, A. Oskin, y V. Artemiev demostraron que los modelos de redes neuronales recurrentes (RNN) se pueden entrenar con éxito para el comportamiento no lineal de un robot submarino (UR), siendo muy eficaces para la presentación de la dinámica del mismo. Además, redujeron la complejidad computacional de la identificación de RNN representándolo como un conjunto de subredes neuronales entrenadas por separado (descomposición) (Dyda 2015). Además, Velázquez Velázquez, Mariaca Gaspar, Galván Guerra, y Ortiz Rodríguez resuelven el problema de identificación de un robot manipulador cilíndrico mediante el uso de las RNR con un algoritmo de aprendizaje backpropagation dinámico, obteniendo un modelo que permite

inferir cómo el robot manipulador cilíndrico responderá a entradas desconocidas, y permitiendo el desarrollo de controladores para gobernar el comportamiento del robot sin necesidad de contar con algún conocimiento sobre el comportamiento dinámico del sistema (Velázquez 2011).

CONCLUSIONES

En el presente trabajo, se concluye que el Software de CoppeliaSim es una fuente vasta de datos de entrenamiento para obtener modelados no paramétricos robots como un SCARA. La red neuronal recurrente entrenada logra una validación de modelo del robot SCARA de 3 GDL con una precisión del 65,8 %.

TRABAJO A FUTURO

Queda abierta la posibilidad a utilizar otros tipos de RNN más eficientes y con mayores números de capas ocultas, como por ejemplo la LSTM. Asimismo, desarrollar modelos de cinemática directa e inversa son factibles mediante la obtención de datos desde CoppeliaSim.

AGRADECIMIENTO

Los autores agradecen el apoyo brindado a sus docentes y asesores de la Universidad Nacional de Trujillo del departamento de Ingeniería Mecatrónica.

CONFLICTO DE INTERÉS

El autor no tiene conflicto de interés de ninguna índole.

REFERENCIAS

- A. A. Dyda, D. A. Oskin y A. V. Artemiev, «Robot Dynamics Identification via Neural Network,» IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, vol. 8°, pp. 918-923, 2015.
https://www.researchgate.net/publication/347159994_Identification_of_Robot_Forward_Dynamics_via_Neural_Network
- C. Robotics, «Coppelia Robotics,» [En línea - 2021]. Available: <https://www.coppeliarobotics.com/>
- E. Grzeidak, «Identification of Nonlinear Systems Based on Extreme Learning Machine and Multiplayer Neural Networks (Master's thesis),» Universidad de Brasíla, Brasíla, 2016.
https://repositorio.unb.br/bitstream/10482/21603/1/2016_EmersonGrzeidak.pdf
- H. Rafiei, A. Aali Hosseini y A. Akbarzadeh Tootoonchi, «Modeling the Dynamic of SCARA Robot Using Nonlinear Autoregressive Exogenous Input Neural Network Model,» Iranian Conference on Electrical Engineering (ICEE), vol. 26th, pp. 994-999, 2018.
<https://www.semanticscholar.org/paper/Modeling-the-Dynamic-of-SCARA-Robot-Using-Nonlinear-Rafiei-Hosseini/2796e7f6d7fe9a03d90e4ab96527a3a8980a14a2>
- J. E. Velázquez Velázquez, C. R. Mariaca Gaspar, R. Galván Guerra y F. Ortiz Rodríguez, «Identificación de un Robot Cilíndrico U,» Congreso Nacional de Mecatrónica, vol. 10°, pp. 97-102, 2011.
<http://www.mecamex.net/anterior/articulos.htm>

- J. Namigtle Jiménez, «Identificación de Sistemas No Lineales Mediante Redes Neuronales, Equilibrio Entre Costo Computacional y Precisión (Tesis de Maestría),» TECNOLÓGICO NACIONAL DE MÉXICO Centro Nacional de Investigación y Desarrollo Tecnológico, Morelos, 2017.
https://www.cenidet.edu.mx/archivos/electronica/tesis/2017/371MC_jnj.pdf
- R. Guatam, A. Gedam, A. Zade y A. Mahawadiwar, «Review on Development of Industrial Robotic Arm,» International Research Journal of Engineering and Technology (IRJET), 2017.
https://www.researchgate.net/publication/322926656_Review_on_Development_of_Industrial_Robotic_Arm
- S. F. Noshahi, A. Farooq, M. Irfan, T. Ansar y N. Chumuang, «Desing and Fabrication of an Affordable SCARA 4-DOF Robotic Manipulator for Pick and Place Objects.,» International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), 2020.
<https://www.semanticscholar.org/paper/Design-and-Fabrication-of-an-Affordable-SCARA-4-DOF-Noshahi-Farooq/0cfe1feeabe3e6403cbd4e59723e615c23477936>